



Advanced Hybrid Approaches Based on Graph Theory Decomposition,
Modified Evolutionary Algorithms and Deterministic Optimisation
Techniques for the Design of Water Distribution Systems

by
Feifei Zheng

Thesis submitted to School of Civil, Environmental & Mining Engineering
of the University of Adelaide
in fulfillment of the requirements for
the degree of
Doctor of Philosophy

Copyright© Feifei Zheng, February, 2013.

Advanced Hybrid Approaches Based on Graph Theory Decomposition, Modified
Evolutionary Algorithms and Deterministic Optimisation Techniques for the
Design of Water Distribution Systems

By:

Feifei Zheng

Supervised by:

Angus R. Simpson, *Ph.D., Master of Science, B.E. (Civil) (Hons).*
Professor, School of Civil, Environmental & Mining Engineering,
The University of Adelaide

Aaron C. Zecchin, *Ph.D., B.E. (Civil) (Hons), B.Sc. (Math & Comp. Sci.)*
Lecturer, School of Civil, Environmental & Mining Engineering,
The University of Adelaide

Thesis submitted in fulfillment of the requirements for the degree of
Doctor of Philosophy

School of Civil, Environmental & Mining Engineering
Faculty of Engineering, Computer and Mathematical Sciences
The University of Adelaide
North Terrace, Adelaide, SA 5005, Australia
Telephone: +61 8303 4323
Facsimile: +61 8303 4359
Web: www.ecms.adelaide.edu.au/civeng
Email: feifei.zheng@adelaide.edu.au

Copyright[©] Feifei Zheng, February, 2013.

Contents

Contents	i
List of Figures and Tables (excluding journal articles)	iv
Abstract	v
Statement of Originality	vii
Acknowledgements	ix
List of Publications	x
Chapter 1. Introduction	1
1.1 Objectives of research	3
1.2 Outline of the thesis	3
1.3 Main contributions of research	6
Chapter 2. Literature review	9
2.1 Optimization model for water distribution system design	9
2.2 Deterministic optimization methods	10
2.2.1 Linear programming (LP)	10
2.2.2 Nonlinear programming (NLP)	14
2.2.3 Binary linear programming (BLP)	15
2.2.4 Analysis of deterministic optimization techniques	17
2.3 Evolutionary algorithms	18
2.3.1 Genetic algorithms	19
2.3.2 Differential evolution	21
2.3.3 Analysis of evolutionary algorithms	27
2.4 Hybrid optimization techniques	30
2.4.1 Combining EAs with deterministic optimization methods	30
2.4.2 Providing EAs with good estimates	33
2.4.3 Analysis of hybrid optimization techniques	33
2.5 Graph theory applications in water network design	34
2.6 Research gaps	35

Chapter 3. Journal Paper 1-Dynamically Expanding Choice Table GA	37
3.1 Synopsis	37
3.2 Journal Paper 1: A dynamically expanding choice table approach to genetic algorithm optimization of water distribution systems (Published in the <i>Journal of Water Resources Planning and Management</i>)	40
Chapter 4. Journal Paper 2-A Dither Creeping Mutation GA for WDS Optimization	51
4.1 Synopsis	51
4.2 Journal Paper 2: A non-crossover dither creeping mutation genetic algorithm for pipe network optimization (Published in the <i>Journal of Water Resources Planning and Management</i>)	54
Chapter 5. Journal Paper 3-A Self-adaptive DE for WDS Optimization	80
5.1 Synopsis	80
5.2 Journal Paper 3: A self-adaptive differential evolution algorithm applied to water distribution system optimization (Published in the <i>Journal of Computing in Civil Engineering</i>)	86
Chapter 6. Journal Paper 4-Combined NLP-DE Method for WDS Optimization	116
6.1 Synopsis	116
6.2 Journal Paper 4: A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems (Published in <i>Water Resources Research</i>)	120
Chapter 7. Journal Paper 5-Coupled BLP-DE Method for WDS Optimization	159
7.1 Synopsis	159
7.2 Journal Paper 5: A coupled binary linear programming and differential evolution approach for water distribution system optimization (Submitted to the <i>Journal of Water Resources Planning and Management</i>)	162
Chapter 8. Journal Paper 6-Decomposition Optimization of WDS with Multiple Sources	193
8.1 Synopsis	193

8.2	Journal Paper 6: A decomposition and multi-stage optimization approach applied to the optimization of water distribution systems with multiple supply sources (Published in <i>Water Resources Research</i>)	196
Chapter 9.	Journal Paper 7-Graph Decomposition Optimization of WDS	244
9.1	Synopsis	244
9.2	Journal Paper 7: A graph decomposition based approach for water distribution network optimization (Submitted to <i>Water Resources Research</i>)	248
Chapter 10.	Conclusions and Future Work	292
10.1	Thesis outcomes	292
10.2	Recommendations of optimization algorithms	294
10.3	Scope for future work	296
References		297

List of Figures and Tables (excluding journal articles)

Figures

Figure 2.1 The algorithm of heuristic linear programming (Morgan and Gouter 1985).....	14
Figure 5.1 A convergence comparison between DE and GA applied to the NYTP case study.....	81
Figure 5.2 A convergence comparison between DE and GA applied to the HP case study.....	81
Figure 5.3 Solutions of the DE and SGA applied to the HP case study.....	82

Tables

Table 2.1 Types of previously used EAs applied to WDS optimization.....	19
Table 2.2 Parameter statistics for various EAs (Tolson et al. 2009).....	28
Table 2.3 Results of GA runs applied to a gradually increasing number of decision variables (Zheng et al. 2011d).....	29
Table 5.1 Parameter values for each algorithm applied to two case studies.....	80

Abstract

The cost of water distribution system (WDS) design or rehabilitation is normally expensive. Over the past 40 years, a number of optimization¹ techniques have therefore been developed to find optimal designs for WDSs in order to save costs, while satisfying the specified design criteria. Often there are a large number of decision variables involved. The majority of currently available optimization techniques exhibit limitations when dealing with large WDSs. Two limitations include (i) finding only local optimal solutions and/or (ii) exhibiting computational inefficiency. The research undertaken in this dissertation has focused on developing advanced optimization techniques that are able to find good quality solutions for real-world sized or large WDS design or rehabilitation strategies with great efficiency. There were three objectives for the research: (i) the modification and improvement of currently available optimization techniques; (ii) the development of advanced hybrid optimization techniques (evolutionary algorithms combined with traditional deterministic optimization techniques) and (iii) the proposal of novel optimization methods with the incorporation of graph decomposition techniques.

The most novel feature of this research is that graph decomposition techniques have been successfully incorporated to facilitate the optimization for WDS design. A number of decomposition techniques have been developed to decompose WDSs by the use of graph theory in this research. Real-world sized or large WDSs are used to demonstrate the effectiveness of the proposed advanced optimization techniques described in this thesis. Results show that these advanced methods are capable of obtaining sound optimal solutions with significantly improved efficiency compared to currently available optimization techniques. The main contribution of this thesis is

¹ American spelling has been used in this thesis as all the publications included in this thesis have been submitted to or published in American journals.

the provision of effective and efficient optimization techniques for real-world sized or large WDS designs or rehabilitation problems.

Statement of Originality

I **Feifei Zheng** hereby declare that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution. To the best of my knowledge and belief in contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying, subject to the provision of the Copyright Act 1968.

I acknowledge that copyright of published works contained within this thesis (as list below) resides with the copyright holder (s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue, and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signature

Date

Journal Papers Published as Part of This Thesis Research:

Zheng, F., A. R. Simpson, and A. C. Zecchin (2011a). "A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems." *Water Resources Research*, 47(8), W08531.

Zheng, F., A. R. Simpson, and A. C. Zecchin (2011b). "Dynamically expanding choice-table approach to genetic algorithm optimization of water distribution systems." *Journal of Water Resources Planning and Management*, 137(6), 547-551.

Zheng, F., A. C. Zecchin and A. R. Simpson (2012a). "A self adaptive differential evolution algorithm applied to water distribution system optimization." *Journal of Computing in Civil Engineering*, doi:10.1061/(ASCE)CP.1943-5487.0000208.

Zheng, F., A. R. Simpson, and A. C. Zecchin (2012d). "A decomposition and multi-stage optimization approach applied to the optimization of water distribution systems with multiple supply sources." *Water Resources Research*, doi:10.1029/2012WR013160.

Zheng, F., A. R. Simpson, A. C. Zecchin, M. F. Lambert (2013). “A non-crossover dither creeping mutation genetic algorithm for pipe network optimization.” *Journal of Water Resources Planning and Management*, doi: 10.1061/(ASCE)WR.1943-5452.0000351.

Acknowledgements

I would take this opportunity to acknowledge the support and guidance of my academic supervisors Prof. Angus R. Simpson and Dr. Aaron C. Zecchin. I appreciate the research freedom I have enjoyed under their supervision. I thank them deeply for always being available to provide constructive suggestions and discuss any issues with great patience.

I would express my appreciation to School Editor Barbara Brougham for her assistance. I would also like to acknowledge the school staff and postgraduate students for their help and company over the past three years.

I would like to acknowledge the support of my family. To my wife Weiwei Bi for her love, patience and continuous support. This dissertation is as much as indirect product of her as it is a direct product of me. To my bother Bingbing Zheng for his help and encouragement. To my parents and my parents-in-law for their love and encouragement.

I acknowledge the research scholarship provided by Australian government and the University of Adelaide.

List of Publications

The following publications were produced from the research associated with the work presented within this thesis.

Journal papers:

- [1] **Zheng, F.**, A. R. Simpson, and A. C. Zecchin (2011a). “A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems.” *Water Resources Research*, 47(8), W08531.
- [2] **Zheng, F.**, A. R. Simpson, and A. C. Zecchin (2011b). “Dynamically expanding choice-table approach to genetic algorithm optimization of water distribution systems.” *Journal of Water Resources Planning and Management*, 137(6), 547-551.
- [3] **Zheng, F.**, A. C. Zecchin and A. R. Simpson (2012a). “A self adaptive differential evolution algorithm applied to water distribution system optimization.” *Journal of Computing in Civil Engineering*, doi:10.1061/(ASCE)CP.1943-5487.0000208.
- [4] **Zheng, F.**, A. R. Simpson, A. C. Zecchin and J. Deuerlein (2012b). “A graph decomposition based approach for water distribution network optimization.” *Water Resources Research*, submitted June 2012.
- [5] **Zheng, F.**, A. R. Simpson, and A. C. Zecchin (2012c). “A combined binary linear programming and differential evolution algorithm approach for water distribution system optimization.” *Journal of Water Resources Planning and Management*, submitted June 2012.
- [6] **Zheng, F.**, A. R. Simpson, and A. C. Zecchin (2012d). “A decomposition and multi-stage optimization approach applied to optimization of water distribution systems with multiple sources.” *Water Resources Research*, doi:10.1029/2012WR013160.
- [7] **Zheng, F.**, A. R. Simpson, A. C. Zecchin, M. F. Lambert (2013). “A non-crossover dither creeping mutation genetic algorithm for pipe network optimization.” *Journal of Water Resources Planning and Management*, doi: 10.1061/(ASCE)WR.1943-5452.0000351.

Conference papers:

- [1] **Zheng, F.**, A. R. Simpson, and A. C. Zecchin (2010). “A method for assessing the performance of genetic algorithm optimization for water distribution design.” *Proc., Water Distribution System Analysis 2010 (WDSA2010)*, ASCE, Tucson, AZ, USA.
- [2] **Zheng, F.**, A. R. Simpson, and A. Zecchin (2011c). “Performance study of differential evolution with various mutation strategies applied to water distribution system optimization.” *Proc., World Environmental and Water Resources Congress 2011*, ASCE, Palm Springs, California.
- [3] **Zheng, F.**, A. R. Simpson, and A. C. Zecchin (2011d). “Optimal rehabilitation for large water distribution systems using genetic algorithms.” *Proc., Australia Water Association (AWA) OzWater 2011*, Adelaide, SA, Australia.
- [4] **Zheng, F.**, A. R. Simpson, and A. C. Zecchin (2011e). “Parametric analysis of differential evolution algorithm applied to water distribution system optimization.” *Proc., 11th International Conference on Computing and Control for the Water Industry (CCWI 2011)*, Exeter, UK.
- [5] **Zheng, F.**, A. R. Simpson, and A. C. Zecchin (2012e). “A performance comparison of differential evolution and genetic algorithm variants applied to water distribution system optimization.” *Proc., World Environmental & Water Resources Congress (EWRI 2012)*, Albuquerque, New Mexico.
- [6] **Zheng, F.**, A. R. Simpson, and A. C. Zecchin (2012f). “A new hybrid optimization model for water distribution system optimization.” *Proc., Water Distribution System Analysis 2012 (WDSA 2012)*, ASCE, Adelaide, SA, Australia.
- [7] **Zheng, F.**, A. R. Simpson, and A. C. Zecchin (2012g). “A performance comparison between differential evolution variants and genetic algorithms for water network optimization.” *Proc., Water Distribution System Analysis 2012 (WDSA 2012)*, ASCE, Adelaide, SA, Australia.

Chapter 1. Introduction

A typical water distribution system (WDS) consists of pipes, reservoirs, pumps, valves and other hydraulic elements. WDSs are used to supply water to users within specific pressure levels under various demand conditions. For a completely new WDS, extensive planning is required to ensure that satisfactory delivery of water is provided to customers in the most reliable and economical way. For an existing WDS, as the water distribution system ages, an optimal rehabilitation strategy is normally needed to improve its service quality.

In most cases, the design, construction and rehabilitation costs for WDSs can be very large; often on the order of millions of dollars. Thus, the optimization of WDSs has historically been investigated by many researchers in order to potentially save significant costs. Optimization of a WDS design normally involves the determination of the optimal network layout, the pipe diameter sizes and the sizes of other system components, such as valves and pumps, thereby providing the minimum total cost life cycle while satisfying all the design constraints. The nonlinear relationship between pipe head loss and discharge, plus the discrete nature of the availability of pipe sizes that can be used, result in many complexities when optimally designing WDSs. For looped WDSs, in which pipe flows and nodal heads are unknown quantities, optimization offers particular challenges.

Traditionally, water engineers have designed WDSs using trial and error approach, and the final design is the result of a combination of engineering experience and judgment. However, the trial and error approach is time consuming and normally only an extremely limited number of WDS designs can be developed and assessed, indicating that a satisfactory solution is the outcome rather than an optimum solution.

Linear programming (LP) and nonlinear programming (NLP) have since been introduced by researchers in order to optimize WDS designs. These methods, unfortunately, are only suitable for 'tree' networks design (no loops involved) and only provide local optimal solutions for looped WDSs. In addition, LP and NLP cannot deal directly with a discrete search space. The optimal solutions provided by LP contain split pipe solutions (usually

two adjacent pipe diameters are assigned for a single link) and the final solutions generated by NLP include continuous pipe diameters, which are both impractical from an engineering perspective.

Subsequently, evolutionary algorithms (EAs) have been employed to optimize the design for WDSs. A number of EAs have been developed for water network optimization. The techniques have been successfully applied to a number of WDS optimization problems, and have proven to be more effective in finding optimal solutions compared with traditional deterministic optimization techniques (LP and NLP). However, EAs tend to locally converge when dealing with complex and large scale optimization problems, and can be inefficient for application in real-world sized WDSs, which normally involve large numbers of pipes and other hydraulic components.

Recently, there has been interest in combining the EAs with traditional optimization techniques such as LP and NLP for WDS optimization. These hybrid optimization techniques are motivated by the fact that EAs are effective in exploring a broad search space while LP and NLP are efficient in exploiting small regions within the whole search space. However, although a few hybrid optimization models have been developed for designing WDSs, they are largely limited to the research domain.

With growing populations and more complex social organisation, WDSs are becoming larger while the standards for the water supply are becoming stricter. More loops and other hydraulic facilities are now involved in WDSs. The increasing complexity and scale of water distribution systems have resulted in enormous challenges for current optimization techniques, and as a result, improved techniques are required. The development of novel methods of optimization to accommodate this demand was the purpose of the research described in this thesis.

1.1 Objectives of research

The four main objectives of this research are:

1. *To improve the performance of evolutionary algorithms (EAs) in terms of optimizing the design of WDSs.* The EAs investigated in this research include the genetic algorithm, which is the most frequently used EA, and the differential evolution algorithm, which is a relatively new EA that has recently received attention in terms of WDS optimization.
2. *To develop hybrid optimization methods that combine EAs with deterministic optimization techniques (such as linear programming (LP) or nonlinear programming (NLP)) for large water network optimization.* These hybrid techniques are used to deal with real-world sized water networks in the current research.
3. *To extend elements of graph theory to enable water network decomposition and develop novel decomposition concepts for use with water networks.* These decomposition techniques are used to partition the water networks in order to facilitate the design optimization. This is motivated by the fact that it is more effective and efficient for optimization techniques to find optimal solutions for relatively small-scale problems (sub-networks after the partitioning process has been carried out) compared to the original full problem (original entire water network).
4. *To develop advanced optimization techniques that incorporate elements of graph decomposition within the whole optimization process for WDSs.* These advanced optimization techniques aim to achieve optimal designs for real-world sized water networks.

1.2 Outline of the thesis

This thesis is a collection of published, accepted or submitted papers from internationally recognised Journals, as shown in the section of List of Publications within the thesis.

Chapter 2 reviews the formulation of the optimization problems for WDS design and the previously published algorithms for WDS optimization. The normal formulation of the optimization model for a WDS design is analyzed in Section 2.1. Deterministic

algorithms including LP, NLP and binary linear programming (BLP) that have been previously employed to optimize WDSs are presented in Section 2.2. Evolutionary algorithms (EAs) that have been applied to WDS optimization are reviewed in Section 2.3. In Section 2.4, the hybrid optimization techniques that have been reported in the literature are presented and analyzed. Section 2.5 describes the application of graph theory in WDS. Water network decomposition is one of the focuses of this thesis. In Chapter 2, the limits of these currently available optimization techniques are elaborated upon.

From Chapters 3 to 9, the titles of the chapters reflect the titles of the journal papers. At the beginning of each chapter, a synopsis of the research motivation and the novelties of the paper are described. The paper that has been submitted, accepted or published is then provided, followed by the short synopsis.

Chapters 3 to 5 focus on the methods that have been developed to improve the effectiveness of the evolutionary algorithms (genetic algorithms and the differential evolution algorithm), which is the first objective of this research (see Section 1.1).

Specifically, **Chapters 3 and 4** introduce a dynamically expanding choice table genetic algorithm and a non-crossover dither creeping mutation genetic algorithm for WDS optimization. The details of the two new GA variants developed in this research are presented in Chapters 3 and 4. In **Chapter 5**, a self-adaptive differential evolution algorithm (SADE) is proposed to reduce the effort required to tune the control parameter values of the DE. In addition, a new convergence criterion is developed and used in the proposed SADE to eliminate the need for the pre-specification of the computational budget.

Chapters 6 and 7 present two novel hybrid optimization techniques that combine EAs with deterministic optimization techniques, which is the second objective of this research (see Section 1.1). In **Chapter 6**, a combined NLP-DE method developed in this research is described, in which NLP is combined with DE to optimize the design of WDSs. The proposed NLP-DE is able to overcome the disadvantages of the currently available

hybrid optimization methods in terms of WDS optimization. A concept of the shortest-distance tree is introduced to enable network decomposition and an algorithm is developed to efficiently identify the shortest-distance tree for a water network.

In **Chapter 7**, a combined BLP-DE approach is presented. In the proposed BLP-DE method, a graph decomposition algorithm is first employed to identify trees and the core for the WDS that is being optimized. Then BLP is used to optimize the design for the trees while a DE algorithm is used to deal with the core optimization design. The proposed method takes advantage of both BLP and DE algorithms: BLP is capable of providing global optimal solution for the trees (no loops involved) with great efficiency, while DE is able to efficiently generate good quality solutions for the core (loops involved) with a reduced search space compared to the original full network. The algorithm details and the results of the BLP-DE applied to the WDS case studies are shown in Chapter 7.

Chapters 8 and 9 outline two advanced optimization techniques that have been developed in this research, which are the third and fourth objectives given in Section 1.1. A decomposition and multi-stage optimization approach for a WDS with multiple water supply sources is presented in **Chapter 8**. In this method, a novel decomposition concept-optimal source partitioning cut-set is proposed and outlined, which is used to partition the water network based on the water supply sources. In addition, an algorithm for efficiently identifying the optimal source partitioning cut-set is developed and presented. The multi-stage optimization method is first developed for water network optimization in this research. The concept of multi-stage optimization is based on the decomposition of large-scale and complex systems into independent subsystems. Each subsystem is optimized independently, and the optimal solutions for each subsystem are then combined to derive the optimal solution for the system as a whole.

Another advanced optimization method based on graph decomposition is proposed in **Chapter 9** of this thesis. A definition of a sub-network based on the connectivity properties of the whole network is given in Chapter 9. In this proposed advanced

algorithm, graph theory is employed to identify the sub-networks for a water network that is being optimized. Rather than optimizing the original network as a whole, the sub-networks are sequentially optimized by the DE algorithm, which is the most novel feature of the advanced optimization technique. The algorithm details are presented in Chapter 9. This is regarded as the most important outcome of this research work presented in this thesis.

In **Chapter 10**, the conclusions of the research are presented in Section 10.1 and possible future extensions based on this research are discussed in Section 10.2.

1.3 Main contributions of research

The five main contributions and innovations delivered by the current research are:

1. *Improving the performance of the evolutionary algorithms (Chapters 3, 4 and 5):*

Two new genetic algorithm variants are introduced for optimizing the design for WDSs. These are dynamically expanding choice table genetic algorithm (Chapter 3) and the non-crossover dither creeping mutation genetic algorithm (Chapter 4). These two GA variants have been demonstrated to be effective for WDS optimization in this research. The non-crossover dither creeping mutation genetic algorithm shows clearly that the performance of a GA can be achieved without crossover and that mutation, used in the right way, is just as effective. This is the first known work to develop a non-crossover and mutation only based genetic algorithm for WDS design.

A self-adaptive DE algorithm (SADE) for optimizing the design of WDSs is proposed in Chapter 5 in order to avoid the need to tune the control parameter values. In addition, a convergence criterion has been proposed in the SADE algorithm in order to avoid pre-specifying convergence conditions (such as the maximum number of allowable evaluations or maximum number of generations) for different optimization problems. Consequently, SADE significantly reduces the effort required for the trial-and-error process normally used to determine the effective parameters for use in the DE algorithm. The proposed SADE provides a robust tool for the optimization of the design of WDSs and rehabilitation of an existing WDS. This is

because (a) the proposed SADE algorithm does not require as much fine-tuning of parameter values nor does it require the pre-specification of a computational budget; and (b) the proposed SADE algorithm is able to find optimal solutions with good quality and great efficiency.

2. *Development of novel hybrid optimization algorithms that combine EAs with deterministic optimization techniques for WDS optimization (Chapters 6 and 7)*

A novel hybrid optimization technique that combines NLP and DE is proposed (Chapter 6). In this context, it is worth noting the comments given by the editorial panel of the journal *Water Resources Research*, which highlights the contribution of this work, saying:

This paper provided a scope for improving the several already attempted algorithms for water distribution system optimization and for searching the new algorithm. It is possible to develop exclusive software for optimal design of water distribution system once the research in this field advances. The authors of the manuscript have put forward some new ideas, which may result in the development of various other developments in the optimal design of large scale water distribution system in future.

Another new hybrid optimization approach that combines the DE with the BLP is introduced (Chapter 7). This proposed BLP-DE is able to find the current best known solutions for two benchmark WDS case studies with the best known efficiency and yield better quality solutions for a real-world case study compared to other EAs with greater efficiency. In the BLP-DE method, BLP is only used to optimize the design of the trees while a DE algorithm is utilized to deal with the optimization of the core portion of the network. The trees and core of the water network are identified using graph theory based on the connectivity properties of the original full water network. The research presented here is the first known work to employ different optimization techniques to optimize different parts of the water network while producing optimal solutions for the whole network.

3. *Development of decomposition techniques for water networks (Chapters 6, 8 and 9).*

A shortest-distance tree is proposed in this research as outlined Chapter 6, with which a looped water network is decomposed into a tree. The shortest-distance tree in the looped network is identified using the Dijkstra graph theory algorithm, for which an

extension in this research is proposed to find the shortest-distance tree for multi-source WDSs. The concept of the shortest-distance tree from water network decomposition is the first time that this idea has been proposed.

A novel concept of an optimal source partitioning cut-set is proposed in this research to decompose a complex water network with multiple supply sources (Chapter 8). The source partitioning cut-set concept is used in this research and an algorithm is developed in this research to efficiently identify the optimal source partitioning cut-set. This is developed by this research for the first time.

Identification of sub-networks of a complex water network using graph theory (Chapter 9) based on their connectivity properties. It is the first known work to use this sub-network identification method to enable water network design.

4. *Development of advanced optimization techniques for designing water networks (Chapters 8 and 9)*

A decomposition and multi-stage optimization technique is introduced to optimize the design of WDSs (Chapter 8). In the decomposition and multi-stage optimization method, graph decomposition technique is used to partition a complex water network into sub-networks. Then each sub-network is optimized independently, and the optimal solutions for each sub-network are then combined to derive the optimal solution for the whole original water network. This method has been demonstrated to be extremely effective for optimizing WDS with multiple supply sources. This research is the first known work to develop a decomposition and multi-stage optimization algorithm for WDS optimization.

A completely novel optimization method based on graph decomposition is developed in this research (Chapter 9). In this proposed method, sub-networks for a water network that is being optimized are identified using graph theory. Rather than optimizing the original network as a whole, the sub-networks are sequentially optimized by the evolutionary algorithm. This approach has been demonstrated to be effective, especially when dealing with large water networks. This part of the research represents the most significant element of the research presented in this thesis.

Chapter 2. Literature review

This Chapter provides a review of the relevant background regarding the optimization of water distribution system (WDS) design. The formulation of the optimization model for WDS design is first reviewed in Section 2.1. Then deterministic optimization methods that have previously been used to tackle WDS optimization problems are reviewed (Section 2.2), followed by a detailed review of evolutionary algorithms that have been applied to WDS optimization design (Section 2.3). Subsequently, the hybrid optimization techniques that have been proposed for optimizing WDS design are reviewed in Section 2.4. Finally, the graph theory applications in WDS optimization design are reviewed (Section 2.5). In addition, this Chapter gives an assessment of each type of optimization algorithm in terms of its capacity to deal with WDS optimization problems.

2.1 Optimization model for water distribution system design

The optimal design for a water distribution system (WDS) normally involves determination of pipe diameters, location and the capacity of tanks, and location and sizes of other hydraulic elements. The objective of WDS optimization is the minimization of life cycle system costs (pipes, tanks and other components) while satisfying a set of constraints at each node. Typically, an optimization model for a WDS design is given by:

$$\text{Minimize} \quad F = a \sum_{i=1}^{np} D_i^b L_i \quad (2.1)$$

Subject to:

$$H_{\min} \leq H \leq H_{\max} \quad (2.2)$$

$$G(H, D)=0 \quad (2.3)$$

$$D_i \in \{A\} \quad (2.4)$$

where F =network cost (to be minimized); D_i =diameter of the pipe i (usually selected from a discrete set of commercially available choices); L_i =length of the pipe i ; a ,

b =specified coefficients in the cost function; np =total number of pipes in the network; $G(H, D)$ =nodal mass balance and loop (path) energy balance equations for the whole water network; H =head at the nodes; H_{min} and H_{max} are the minimum and maximum allowable heads at the nodes; A = a set of commercially available pipe diameters.

Two features contribute to the nonsmoothness properties of WDS optimization problems. These include: (1) the pipe diameter choices being composed of discrete sizes rather than being continuous decision variables (Equation 2.4); and (2) the nonlinear term involving the discharge or velocity within the head loss equations (Equation 2.3). The nonsmooth nature of the landscape constituted by a WDS design problem results in many local optimal solutions, which poses a challenge when seeking good quality or global optimal solutions.

Due to the complexity of the WDS optimization problem, a large body of research has been undertaken to develop techniques for WDS optimization design in the past 40 years. Generally, these optimization techniques can be divided into three types, which are *deterministic optimization approaches*, *evolutionary optimization techniques* and the *hybrid optimization methods*. Each type is reviewed below.

2.2 Deterministic optimization methods

2.2.1 Linear programming (LP)

Alperovits and Shamir (1977) presented a linear programming (LP) gradient method to find the least-cost design for WDSs. In their proposed LP for WDS design, each pipe was assumed to be composed of segments of different pipe diameters and the formulation of this proposed LP model is given by:

$$\text{Minimize} \quad F = \sum_{i=1}^{np} \sum_{j=1}^m C(D_j) L_{ij} \quad (2.5)$$

Subject to:

$$H_k^{\min} \leq H_k \leq H_k^{\max} \quad (2.6)$$

$$H_k = H_s - \sum_{l=1}^{P(k)} h_l \quad (2.7)$$

$$h_l = \sum_{j=1}^m \omega \frac{L_{lj}}{C_j^\alpha D_j^\beta} Q_l^\alpha \quad (2.8)$$

$$\sum_{Loop} h_{nl} = 0 \quad (2.9)$$

$$\sum_{j=1}^m L_{ij} = L_i \quad (2.10)$$

$$D \in \{A\} \quad (2.11)$$

where $C(D_j)$ = cost per unit length (\$/m) for pipe diameter j ; L_{ij} = the segment length of pipe diameter j in link i ; m = total number of available discrete pipe diameters; np = total number of links of the WDS to be optimized; n = total number of nodes of the WDS to be optimized; H_k = head at node $k=1, \dots, n$; H_k^{\min} = minimum allowable head requirement at node k ; H_k^{\max} = maximum allowable head requirement at node k ; H_s = head at supply source node (reservoir or pump); $P(k)$ = water supply path from source node s to node k ; h_l = head loss in pipe l ; ω = numerical conversion constant which depends on the units; α, β = coefficients corresponding to the Hazen-Williams head loss equation; C_j = Hazen-Williams coefficient for pipe diameter j ; Q_l = pipe flow rates in pipe l (m^3/s); L_{lj} = the segment length of pipe diameter j in link l ; h_{nl} = head loss in pipe nl ; $\sum_{Loop} h_{nl}$ = the sum of

the head loss for each primary loop; L_i = the total length of link i ; A = the set of commercially available pipe diameters. Note that the formulation given by Alperovits and Shamir (1977) is based on the Hazen-Williams head loss equation. An LP could also be formulated in terms of Darcy-Weisbach head loss equation.

As can be seen from the LP model given above, the unknown lengths of pipe segments were the decision variables rather than the discrete pipe diameter sizes. The original nonlinear WDS optimization problem was therefore converted to a linear optimization problem because the cost of the network is linearly proportional to the length of each pipe segment. Constraints (2.6) to (2.8) ensure the pressures at all nodes are within the specified range. Equation (2.9) is the energy balance at each primary loop (path) of the water network. Constraint (2.10) ensures that the total length of each segment equals to the original total length of each link i . It should be highlighted that this LP model allows split pipe diameter solutions, in which various pipe diameters are selected for a single link and each pipe diameter is associated with a particular segment.

In the LP method proposed by Alperovits and Shamir (1977), the WDS optimization problem is decomposed into two stages, namely the inner and the outer stage. Initially, a set of pipe flows is selected for the water network that is being optimized to satisfy the continuity at each node in the outer stage. Then the LP model presented by Equations (2.5) to (2.11) is formulated and solved in the inner stage to find the combination of pipes that offers the least cost to the network based on the known flow distribution obtained in the outer stage. In addition, a vector of gradients of the cost against flow for each loop is obtained during the LP optimization. These gradients are, in turn, used to determine the magnitude and the direction of the loop-flow steps, thereby producing an updated flow distribution of the water network in the outer stage. The LP is rerun to find the least-cost design of the pipe network based on the updated flow distribution in the inner stage. This process is iteratively performed until no further improvement is achieved within the minimum step size allowed or the maximum number of iterations is exceeded.

In the LP method reported by Alperovits and Shamir (1977), a gradient is obtained for each loop flow change using the values of the dual variables at each iteration. Then, to reduce the cost of the network, a fixed step length for the loop flow variation is taken along the direction specified by the gradient. This approach was criticized by **Fujiwara et al. (1987)** for its inefficiency when dealing with relatively large WDSs, since the

search direction was a negative gradient and the dimension of the gradient vector was actually the total loops in the network.

The negative gradient direction is referred to as the direction of “steepest decent” in nonlinear programming terminology (Fujiwara et al. 1987). It is well known that the method of “steepest decent” normally requires a large number of iterations while making very slow progress toward a solution since the convergence rate of this algorithm is only linear (Fujiwara et al. 1987). Therefore, the LP method proposed by Alperovits and Shamir (1977) is subject to large computational overheads, resulting in inefficiency.

In order to overcome this drawback, Fujiwara et al. (1987) proposed a quasi-Newton technique and backtracking line-search method to determine the directions and magnitudes of the loop flows, which improved both convergence rate and speed when compared to the LP method originally used by Alperovits and Shamir (1977).

Quindry et al. (1981) presented another method to tackle the least-cost design for WDSs. In their method, a set of initial nodal pressures, rather than the flows as used by Alperovits and Shamir (1977), was assumed for the water network being optimized. Based on the known nodal heads, pipe diameters were then selected using LP to be the least-cost whilst satisfying the continuity equations at all nodes. A set of gradients of the cost against heads at all nodes was employed to ensure the iteration progressively moved towards the least-cost design.

Calhoun (1981) applied an LP to optimize tree networks, in which a pump was included. **Stephenson (1984)** also developed an LP to deal with the optimization of the trunk main pipes (the tree network), in which, the simplex method was employed to solve the LP. In addition, the LP has been extended by Stephenson (1984) to optimize the looped network. An assumption was made in his work that the least-cost network was in fact invariably a tree-like network. Hence, the looped network was reduced to a tree-like network first and then the LP was formulated for the tree network.

Morgan and Goulter (1985) described a heuristic LP approach linked with a network solver to find the least-cost design for WDSs. Figure 2.1 illustrates this algorithm. As shown in Figure 2.1, for a given combination of pipes for the water network initially, a hydraulic solver is carried out to maintain the continuity at all nodes and energy conservation in all simple loops. In addition, the flow distribution of the water network is obtained. Once this process is complete, a new pipe combination for the known flows in the pipes is generated using LP while satisfying minimum and maximum head requirements at each node. If the new pipe combination is the same as the one evaluated by the network solver, the least-cost design has been obtained. Otherwise, the new pipe combination is evaluated by the network solver again to produce an updated flow distribution. The process is iterative, and stops if there is no difference between the resulting pipe combination and the previous combination, or the maximum number of iterations has been exceeded.

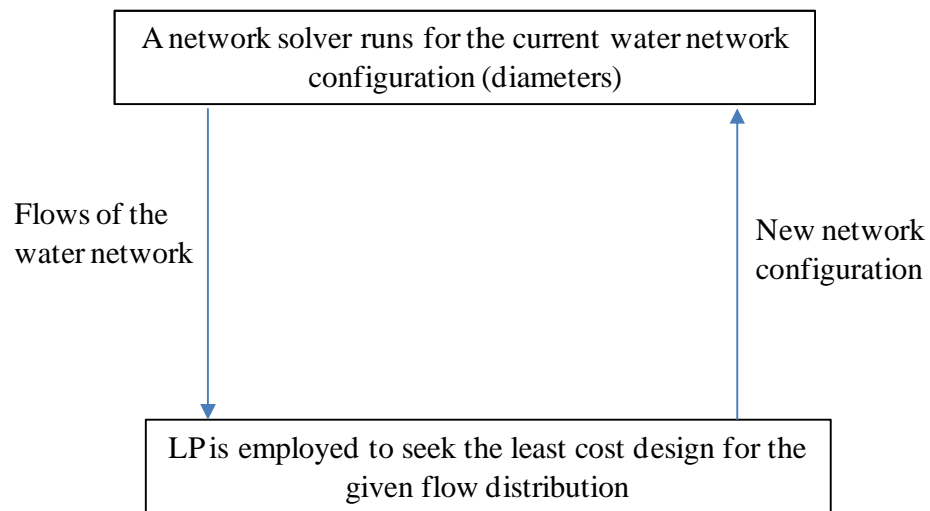


Figure 2.1 The algorithm of heuristic linear programming (Morgan and Goulter 1985)

2.2.2 Nonlinear programming (NLP)

The objective function of the least-cost design of WDSs with a set of constraints is mathematically nonlinear. Nonlinear programming (NLP) can handle the nonlinear problem directly and was introduced by researchers to optimize the design of WDSs.

Lansey and Mays (1989) proposed a generalised reduced gradient (GRG) NLP technique for dealing with WDS optimization problems. In their work, the original whole optimization problem was decomposed into the simulation and optimization steps. In their proposed simulation-optimization model, a hydraulic simulator is used to solve for the pipe flows and nodal pressures that satisfy the constraints for any given pipe diameters, while GRG NLP is employed to iteratively update the pipe diameters. The advantage of this method is that the size of the optimization problem is reduced as the hydraulic constraints are handled by the simulator.

Subsequently, **Fujiwara and Khang (1990)** proposed a two-stage decomposition NLP optimization technique for WDS design. In the first stage of the two-stage optimization method, a NLP gradient method was introduced to extend the LP gradient method proposed by Alperovits and Shamir (1977). A set of flows that satisfy the continuity at each node is first assumed, and then the NLP gradient method is employed to find a local optimal solution for the water network. In the second stage, the link head losses of the obtained local optimum in the first stage are fixed and the resulting concave problem is solved using NLP to obtain a new flow distribution. The resultant new flow distribution is used to restart the first phase, and the two stages are continued until no better local optimum can be found. The main advantage of the two-stage NLP optimization method proposed by Fujiwara and Khang (1990) is that it is able to generate a sequence of improving local optimal solutions. However, this method cannot guarantee the global optimal solution is found although it allows a move from one local optimal solution to another, better one.

2.2.3 Binary linear programming (BLP)

Samani and Mottaghi (2006) proposed a binary linear programming (BLP) approach for WDS design optimization, in which the objective function and constraints are linearized using zero-one variables. The formulation of a BLP model proposed by Samani and Mottaghi (2006) in terms of Hazen-Williams head loss equation is given by:

$$\text{Minimize} \quad F = \sum_{i=1}^N \sum_{j=1}^P X_{ij} L_i C(D_j) \quad (2.12)$$

Subject to:

$$\sum_m^{W_{k-R}} h_{fm} \leq H_R - H_k^{\min} \quad m \in W_{k-R} \quad (2.13)$$

$$h_{fi} = \sum_j^P \omega \frac{L_i X_{ij}}{C_j^\alpha D_j^\beta} q_i^\alpha \quad i \in T \quad (2.14)$$

$$\sum_{j=1}^P X_{ij} = 1 \quad (2.15)$$

$$D_i \in \{A\} \quad (2.16)$$

where N is the total number of pipes needs to be optimized; P is the total number of commercially discrete pipe diameters that can be used; L_i is the length of pipe i ; $C(D_j)$ is unit length cost of the pipe diameter D_j and X_{ij} is zero-one variable; H_R is the available head provided by the source node; H_k^{\min} is the minimum allowable head requirement for node k ; W_{k-R} is the water supply path from source node R to node k ; $\sum_m^{W_{k-R}} h_{fm}$ is total head loss involved in water supply path W_{k-R} h_{fi} is the head loss for pipe i ; q_i = flows in pipe i ; ω = numerical conversion constant which depends on the units of flows and diameters; α , β = coefficients and C_j = Hazen-Williams coefficient of pipe diameter j .

In Equation (2.12), $X_{ij}=1$ indicates that the diameter D_j is selected for pipe i while $X_{ij}=0$ represents the diameter D_j is not selected for pipe i . It is noted that no nonlinear terms are involved in the objective function F . In Equation (2.14) $X_{ij}=1$ implies that diameter D_j is used for pipe i and then the h_{fi} based on the selected diameter D_j is obtained. While $X_{ij}=0$ means that diameter D_j is not selected for pipe i and no head loss is involved for diameter D_j . As can be seen from Equation (2.14), by utilising zero-one variables, the nonlinear Hazen-Williams formula is converted to a linear formula if flows are known for each pipe.

Four steps are involved in the BLP method proposed by Samani and Mottaghi (2006):

Step 1: Each pipe in the water network to be optimized is initially assigned a commercially available pipe diameter.

Step 2: A hydraulic solver is performed for the known network configuration to obtain water flows for each pipe.

Step 3: A BLP model is formulated and solved for the water network based on the known flows at each pipe and solved while satisfying the head constraints at each node.

Step 4: The resulting pipe sizes obtained in step 3 are compared with the assumed quantities in step 1. If they are the same, the optimization process has converged and the resulting pipe sizes are the final solution; otherwise, the resulting pipe sizes are assigned to the water network and steps 2, 3 and 4 are repeatedly performed until the convergence (resulting pipe sizes in step 3 are the same with the those used in step 2) is achieved.

Samani and Mottaghi (2006) used two relatively small looped WDS case studies to verify the effectiveness of their proposed BLP method, and reported that the performance of the BLP method was satisfactory in terms of accuracy and convergence based on results of two looped WDS case studies.

2.2.4 Analysis of deterministic optimization techniques

The advantage of these deterministic methods including LP, NLP is that they are able to provide local optimal solutions with great efficiency for treed water networks. This is because the flow distribution in the treed water network can be pre-determined since no loops are involved. Although these deterministic optimization approaches have been extended to deal with the optimization of small looped water networks, the majority of the applications remain in the research domain. Limitations of these approaches are given as follows.

Linear programming. The disadvantages of LP include:

- It is highly likely to become trapped by local optimal solutions due to its point by point movement (gradient based) in the search space;
- The final solution allows split pipe diameters, in which a link is composed of several different (usually two adjacent) pipe diameters and each diameter is associated with a particular length. This is impractical from an engineering perspective since it normally uses one pipe diameter for a link.

Nonlinear programming. For NLP, the drawbacks are that:

- It is also highly likely to be trapped by local optimal solutions in the same way that occurs for LP;
- It allows the continuous pipe diameters in the final solution, which is a severe disadvantage as only commercially discrete pipe diameters can be used in practice. Thus, a procedure is needed to round off the continuous pipe diameter to the nearest discrete one, which may lead to a sub-optimal solution or even an infeasible solution for the WDS design

Binary linear programming. The advantage of the BLP developed by Samani and Mottaghi (2006) over LP and NLP is that it is able to provide discrete pipe diameter solutions over complete segment of pipe lengths. However, the BLP approach is compromised by extreme inefficiency when dealing with relatively large WDS case studies (Savic and Cunha 2006). In addition, the global optimum for a looped WDS cannot be guaranteed as the final solution reached by the BLP approach is dependent on the initially assumed pipe diameters (Martínez 2006).

2.3 Evolutionary algorithms

Within the past two decades, evolutionary algorithms (EAs) have frequently been used to optimize WDSs. EAs are able to handle discrete search spaces directly and are less likely to be trapped at local optima. The search strategy of EAs differs from deterministic optimization techniques (such as LP or NLP) in that EAs explore the search space in a manner broadly based on stochastic evolution rather than on gradient information. A

number of EAs have been developed for optimizing WDS design, and the first significant publication of each EA is provided in Table 2.1. These EAs have been successfully applied to a number of WDS design optimization problems, yielding better quality solutions than deterministic optimization techniques.

Table 2.1 Types of previously used EAs applied to WDS optimization

Algorithm ¹	First reference ²
Genetic algorithm (GA)	Simpson et al. (1994)
Simulated annealing (SA)	Loganathan et al. (1995)
Tabu search (TS)	Lippai et al. (1999)
Harmony search (HS)	Geem et al. (2002)
Shuffled frog leaping algorithm (SFLA)	Eusuff and Lansey (2003)
Ant colony optimization (ACO)	Maier et al. (2003)
ANN metamodels	Broad et al. (2005)
Particle swarm optimization (PSO)	Suribabu and Neelakantan (2006)
Scatter search (SS)	Lin et al. (2007)
Cross-entropy algorithm (CE)	Perelman and Ostfeld (2007)
Differential evolution (DE)	Suribabu (2010)
Honey-Bee Mating Optimization (HB)	Mohan and Babu (2010)
Genetic Heritage Evolution by Stochastic Transmission (GHEST)	Bolognesi et al. (2010)

¹All mentioned algorithms in this table are referred to Evolutionary Algorithms for ease of reference, although the metaheuristics TS and SA are not strictly EAs. ²Only the first significant paper for each EA applied to WDS optimization is provided

The EAs investigated in this research include the genetic algorithm (GA), which is the most frequently used EA, and the differential evolution algorithm (DE), which is a relatively new EA that has recently received attention in terms of WDS optimization.

2.3.1 Genetic algorithms

Amongst EAs presented in Table 2.1, GAs have gained popularity due to their ease of implementation and satisfactory search ability. **Nicklow et al. (2010)** presented a comprehensive review on the GA applications to various water resources planning and management problems during the last two decades. It was concluded by Nicklow et al. (2010) that the GA has been consistently proven to be flexible and powerful in solving complex water resources problems.

The GA is a stochastic search technique based on artificial evolution (Holland, 1975). Three operators including selection, crossover and mutation, are commonly used in the GA to constitute the evolution. **Simpson et al. (1994)** first used GA to solve the optimization problem of WDSs. In their work, a performance comparison was made between GAs and deterministic optimization techniques (LP and NLP) in terms of finding optimal solutions for WDSs. It was reported by Simpson et al. (1994) that GAs are far more effective for providing good quality solutions than deterministic optimization techniques. In addition, a standard GA implementation for WDS optimization design was elaborated on by Simpson et al. (1994).

Subsequently, considerable research has been undertaken to improve the performance of GAs in terms of WDS optimization. **Dandy et al. (1996)** proposed an improved GA for WDS optimization. Compared to the standard GA, three modifications were made by Dandy et al. (1996) for the improved GA. These include

- Introducing a variable scaling power of the fitness function into the GA, accentuating the small differences between string fitness in the later generations when the Roulette wheel selection method is used. This method is able to lead the GA to explore the best region of the solution space when highly fit strings are dominating in the later generations.
- Using a Gray coding scheme rather than binary coding. Adjacent codes representing nearby designs in the solution space are guaranteed by the Gray coding method, thereby avoiding the Hamming cliff and making the GA perform better.
- Implementing an adjacency mutation in addition to bitwise mutation to allow the GA to locally explore. The improved GA was demonstrated to be more effective in finding better quality solutions for the case study used by Dandy et al. (1996).

Vairavamoorthy and Ali (2000) applied an integer coding method in the formulation of GA strings, thereby avoiding the redundant states often found when using binary or Gray coding. Deb (2000) introduced a constraint tournament selection algorithm to facilitate the GA to effectively handle the constraints. The basic algorithm when comparing two solutions in a constraint tournament selection is given as follows:

- A feasible solution is selected when compared with an infeasible solution.
- The solution with a smaller value of the objective function value (if cost is being minimised) is preferred between two feasible solutions.
- The solution with less constraint violation is preferred between two infeasible solutions.

Using this method, the comparison between the solutions in a tournament never happens in terms of both objective function and penalty function. In the first case, the solution with no head violation is preferred to the one with a head violation and does not take the value of the objective function into account. In the second case, the two solutions are compared based on the objective values and the one with a smaller value is selected as both solutions satisfy the constraints. In the last case, the solution with less head violation is selected and the value of the objective function is not considered. Thus, unlike traditional tournament selection, there is no need to specify a penalty multiplier in the proposed method.

Wu et al. (2001) introduced a fast messy genetic algorithm (mGA) to deal with the optimization of water networks, which showed a significant improvement in terms of efficiency and robustness compared to the standard GA. **Vairavamoorthy and Ali (2005)** proposed a pipe index vector based GA for WDS optimization design. In their work, a pipe index vector was established to assess the relative importance of the pipes in terms of their impact on the hydraulic performance of the pipe network. This pipe index reduced the search space for the GA and guided the GA search to promising regions where the optimal solutions were likely to be.

2.3.2 Differential evolution

The differential evolution (DE) algorithm, introduced by **Storn and Price (1995)**, has been found to be a relatively simple but powerful EA for global optimization. More recently, the DE algorithm has received much attention as a method of dealing with WDS optimization problems (**Suribabu 2010**). Three operators are involved in the DE during optimization including mutation, crossover and selection operators. The process

names are similar to those the commonly used when talking about GA. However, there are significant differences in the order of application and form of these operators.

DE differs significantly from a GA in the mutation process due to the fact that the mutant solution is generated by adding the weighted difference between several random population members to another random member of the population. Three parameters need to be pre-specified for the use of DE including the population size (N), mutation weighting factor (F) and crossover rate (CR). In addition to these three parameters, a particular mutation strategy needs to be selected for the use of DE among a number of possibilities (Price et al. 2005). Since DE is a relatively new optimization algorithm in the water community, the basic process of standard DE is reviewed in the sub-sections that follow (Storn and Price 1995).

2.3.2.1. Initialization

DE is a population based stochastic search technique. Thus, a set of members of the initial population is required to initialise the DE search. Normally, each initial population $X_{i,0} = \{ x_{i,0}^1, x_{i,0}^2, \dots, x_{i,0}^D \}$ is generated by randomising individuals from a uniform distribution within the search space, that is

$$x_{i,0}^j = x_{\min}^j + rand(0, 1)(x_{\max}^j - x_{\min}^j) \quad i=1, 2, \dots, N, j=1, 2, \dots, D \quad (2.17)$$

where $x_{i,0}^j$ = the initial value of the j^{th} parameter for the i^{th} individual in the initial population, x_{\min}^j and x_{\max}^j = the minimum and maximum bounds of the j^{th} parameter respectively, $rand(0, 1)$ represents a uniform distributed random variable in the range $[0, 1]$, while N and D = population size and dimension of the vector respectively. The population size is not changed during the DE process.

2.3.2.2. Mutation

DE is mainly defined by its mutation approach, compared with GAs, in that a mutant vector $V_{i,G}$, with respect to each individual $X_{i,G}$, is produced by adding the weighted difference (with weight F) between several random population members to a third member from the current population. Each individual $X_{i,G}$ associated with a mutant vector is denoted as a target vector. Five frequently used mutation strategies in DE are provided:

DE1-*Rand1*:

$$V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) \quad (2.18)$$

where $V_{i,G}$ is the mutant vector with respect to the target vector of $X_{i,G}$ at generation G . $X_{r1,G}$, $X_{r2,G}$ and $X_{r3,G}$ are three vectors randomly selected from the current population G . As shown in Equation (2.18), DE1 generates a mutant vector $V_{i,G}$ for each vector i by adding the weighted difference of two randomly selected vectors to a third vector. The random integers $r1$, $r2$ and $r3$ are different values from the population of size N . F is the weighted difference factor within the range $[0, 1]$.

DE2-*Best1*:

$$V_{i,G} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) \quad (2.19)$$

DE2 is similar to DE1 in terms of producing the mutant vector except that the third vector that is to be perturbed is the best individual of the current generation ($X_{best,G}$).

DE3-*Best2*:

$$V_{i,G} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G}) \quad (2.20)$$

where $X_{best,G}$ is the best individual of the current generation G . DE3 uses two weighted differences of four randomly selected individuals and the best individual to produce the mutant vector. The random integers $r1$, $r2$, $r3$ and $r4$ are different values chosen from the population of size N .

DE4-*CurrentToBest2*:

$$V_{i,G} = X_{r1,G} + F(X_{best,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G}) \quad (2.21)$$

Like DE3, DE4 also employs two weighted difference individuals, but one is the weighted difference between the best individual and a random individual. In addition, for DE4, the individual to be perturbed is a random individual rather than the best individual that used in DE3.

DE5-*rand2*:

$$V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) + F(X_{r4,G} - X_{r5,G}) \quad (2.22)$$

DE5 is quite similar to DE3, only differing in that the individual to be perturbed is a random individual from the population of DE5, while the individual to be perturbed is the current best individual for DE3. The integers $r1$, $r2$, $r3$, $r4$ and $r5$ are different values randomly selected from the population of size N .

2.3.2.3. Crossover

After the application of the mutation operator, a trial vector $U_{i,G}$ is generated through selecting solution component values of from either $X_{i,G}$ or $V_{i,G}$. In the basic DE version (Storn and Price 1995), uniform crossover is employed as:

$$u_{i,G}^j = \begin{cases} v_{i,G}^j, & \text{if } \text{rand}(0,1)_j \leq CR \\ x_{i,G}^j, & \text{otherwise} \end{cases} \quad (2.23)$$

where $u_{i,G}^j$, $v_{i,G}^j$, $x_{i,G}^j$ = the j^{th} parameter for the i^{th} trial vector, mutant vector and target vector respectively, CR is the crossover rate within the range of $[0, 1]$, $\text{rand}(0,1)_j$ is a random number between 0 and 1 generated for each parameter j . If $\text{rand}(0,1)_j$ is smaller than CR , the parameter $v_{i,G}^j$ in the mutant vector is copied to the trial vector, otherwise, the parameter $x_{i,G}^j$ in the target vector is copied to the trial vector.

2.3.2.4. Selection

After crossover, all the trial vectors are evaluated using the objective function $f(U_{i,G})$ and are compared with their corresponding trial vector objective function $f(X_{i,G})$. The vector with a lower objective function value (given a minimisation problem) survives for the next generation. That is

$$X_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases} \quad (2.24)$$

where $X_{i,G+1}$ is the i^{th} individual at the generation $G+1$.

Mutation, crossover and selection are repeatedly applied generation by generation until a stopping criterion (normally a maximum number of allowable evaluations) is satisfied.

Suribabu (2010) first introduced DE algorithm to optimize the WDS design. A total of four WDS case studies were used in his study to assess the effectiveness of the DE algorithm for optimizing WDS design. The results obtained by Suribabu (2010) clearly showed that DE significantly outperformed other EAs such as GAs and ACOs in terms of efficiently finding optimal solutions. **Vasan and Simonovic (2010)** developed a DENET optimization model to tackle the WDS optimization problem, in which a DE algorithm was combined with the network simulation model EPANET2.0 (Rossman 2000). The efficiency and robustness of the DENET was tested based on two benchmark WDS problems. It was reported by Vasan and Simonovic (2010) that DE was able to provide good quality optimal solutions with great efficiency based on results obtained for the two benchmark WDS problems.

2.3.2.5. Parameter sensitivity analysis for differential evolution

Research has been undertaken to systematically analyze the influence of the control parameters on the performance of DE applied to numerical optimization problems, and provide guidelines for selecting appropriate control parameters (Storn and price 1995; Price et al. 2005; Liu and Lampinen 2005). It is argued by these researchers that DE, with $1D \leq N \leq 10D$ (where D is the number of decision variables) $0.5 \leq F \leq 1.0$, $0.8 \leq CR \leq 1.0$ shows generally good performance in convergence properties. In addition, DE with $N=10D$, $F=0.5$ and $CR=0.9$ are recommended as universally suited control parameters for different numerical optimization problems. However, as with most EAs, the optimal setting of these parameters is heavily reliant on the properties of the fitness landscape associated with the problems that are being optimized. The numerical optimization problems that have been used to verify the effectiveness of DE (Storn and price 1995; Vesterstrom and Thomsen 2004; Price et al. 2005; Liu and Lampinen 2005) all have continuous search spaces, while the WDS optimization problem is a discrete search space problem as only commercially available pipe diameters can be used for the WDS

design. The global optimal solutions for these numerical optimization problems lie at the centre of the search space or along the coordinate axes. In addition, there is no linkage among the different variables in a numerical optimization problem. Whereas, for the optimization of the WDS design, the global optimal solution for the WDS design normally lies at the boundary of the whole search space and different variables interact with one another. Thus, the WDS optimization problem has a by far more complex fitness landscape than the numerical optimization problem. Consequently, recommendations for parameter guidelines based on the numerical optimization problems cannot necessarily be directly transferred to the WDS optimization problem.

Vasan and Simonovic (2010), and Suribabu (2010) concluded that the performance of a DE algorithm was at least as good as, if not better, than other EAs such as GAs and Ant Colony Optimization. However, **Dandy et al. (2010)** compared performance of GAs and DE in terms of optimizing WDSs and stated that GAs gave better results overall than DE. This contradiction can be explained by the fact that the different parameter values including N , F and CR may be used in these DE applications.

Zheng et al. (2011e) undertook a systematic parameter analysis for the DE algorithm in terms of WDS optimization. The parameters involved in their work were the mutation weighting factor (F) and the crossover rate (CR), which are considered to be the most important parameters to influence the DE algorithm's performance. The researchers concluded that the performance of DE is dependent on these two parameter values and that the appropriate DE parameter values are optimization problem dependent. Thus, a trial-and-error process is required to determine the preferable parameter values when DE is applied to a given WDS optimization problem. **Zheng et al. (2011c)** have also investigated the effectiveness of the five available mutation strategies (see Equations 2.18 to 2.22) in terms of WDS optimization and concluded that the mutation strategy given in Equation (2.18) exhibited the overall best performance.

2.3.3 Analysis of evolutionary algorithms

Advantages of EAs compared to deterministic methods can be concluded as follows in the context of optimizing WDS design.

- EAs are able to cover more regions of the search space than deterministic methods such as LP and NLP as they are stochastic optimization techniques. As a result, it is more likely that EAs will reach good optimal solutions for WDS optimization problems.
- EAs are capable of handling the discrete search space of the WDS design problem directly. This is of great benefit as they are able to produce practical final solutions, with each pipe being assigned a commercially discrete diameter. While split pipe solutions or continuous pipe diameters are included in the final solutions generated by deterministic methods (LP and NLP), both are impractical in practice.
- In contrast with deterministic methods, EAs can provide a set of solutions at the end of each run. These solutions are slightly different in cost but completely different in design. Thus the practitioner can select the more practical design from the options based on objectives which cannot be expressed explicitly during optimization.
- EAs can be modified to deal with multi-objective WDS design problems, while deterministic approaches are only limited for single objective optimization problems.

However, there are also limitations of EAs when applied to WDS optimization problems. The efficiency of EAs, for example, is frequently of concern, especially when dealing with relatively large and complex WDS optimization problems where simulation model run times are long. The majority of EAs are population based search algorithms and a hydraulic simulation model is normally required to evaluate each individual of the EA population, resulting in a large computational overhead. The inefficiency of EAs when dealing with large-scale problems has also been clearly stated by Nicklow et al. (2010).

Another issue in the use of EAs is that a number of parameter values need to be tuned when EAs are applied to various optimization problems. Table 2.2 gives a summary of

number of parameters that need to be selected for different EAs that have been reported for WDS design optimization (Tolson et al. 2009). As can be seen in Table 2.2, the number of parameters varies from three to eight. The performance of these EAs has been demonstrated to be heavily dependent on the parameter values used and suitable parameter values are dependent on the optimization problem under consideration. Thus, it requires significant effort, normally by trial and error, for practitioners to determine the most appropriate parameter values for these EAs in order to apply them to different optimization problems. The resulting computational budget is unavoidably large.

Table 2.2 Parameter statistics for various EAs (Tolson et al. 2009)

Algorithm	Reference	Number of reported parameters
GENOME	Reca and Martinez (2006)	8
PSO variant	Montalvo et al. (2008)	8
SFLANET	Eusuff and Lansey (2003)	6
GHEST	Bolognesi et al. (2010)	6
HS	Geem (2006)	5
MSATS	Reca et al. (2007)	5
GA	Simpson et al (1994)	5
MMAS-ACO	Zecchin et al. (2007)	4
PSHS	Geem (2009)	4
DE	Suribabu (2010)	4
CE	Perelman and Ostfeld (2007)	3

The solution quality cannot be guaranteed for EAs when dealing with large case studies. **Zheng et al. (2011d)** investigated the search ability of GAs applied to a number of case studies with the number of decision variables ranging from 21 to 1050 pipes. The performance assessment of the GAs was made by comparing the optimal solution found by GAs and the estimated global optimal solution for each case study. The results of this study are given in Table 2.3.

The results recorded in Table 2.3 demonstrate that the GA was able to find the estimated global optimal solutions (current best known solutions) for case studies with 21 and 42 decision variables. The GA exhibited reasonably good performance on the case study with 105 decision variables; the best solution it deviates only 0.53% from the estimated global optimal solution.

Table 2.3 Results of GA runs applied to a gradually increasing number of decision variables based on the New York Tunnels Problem (NYTP) network (Zheng et al. 2011d)

NOTE:

This figure/table/image has been removed to comply with copyright regulations. It is included in the print copy of the thesis held by the University of Adelaide Library.

As shown in Table 2.3, the best solution found by the GA deviated further from the estimated global optimal solution as the number of decision variables increased. This shows that the optimization problem becomes more and more intractable for GAs as the number of decision variables increases. When the number of decision variables increases to 1050, the best solution found by the GA was 16% higher than the estimated global optimal solution. Thus, it can be concluded that GAs are able to perform well on relatively small case studies (for a small number of decision variables) in terms of solution quality, whereas solution quality deteriorates for GAs when dealing with relatively larger networks.

In a conclusion, factors including (i) the inefficiency, (ii) the large effort required to tune parameter values and (iii) the deterioration of the solution quality are major concerns for EAs when dealing with real-world sized water networks, for which a large number of pipes and other components are normally involved. Thus, it is desirable to develop advanced optimization techniques to overcome these limitations to enable a generic application of optimization techniques for WDS design.

2.4 Hybrid optimization techniques

A number of hybrid optimization techniques have been developed in recent years to overcome disadvantages of EAs (inefficiency and solution deterioration) applied to relatively large WDS optimization problems. Typically, two approaches are used to enhance the search performance of EAs, including (1) refining an EA's search performance by combining it with traditional optimization techniques (LP or NLP) or local search procedures and (2) guiding an EA's search by providing it with initial near-optimal seeding estimates.

2.4.1 Combining EAs with deterministic optimization methods

When EAs are combined with deterministic optimization methods, they are generally used first to locate the approximate regions of the optimal solutions for the problems that are being optimized. A traditional or local search method is then employed to find the minimum solution within the localized search space region identified by the EAs. A few hybrid optimization models combining the EAs and traditional methods have been developed and improvements have been reported in terms of WDS optimization.

Tolson et al. (2009) developed a hybrid discrete dynamically dimensioned search (HD-DDS) algorithm to optimize the design of WDSs. The HD-DDS combines an evolutionary search method with two local search approaches: a one-pipe search and a two-pipe search. In their work, the metaheuristic search method is first used to explore broadly in the whole search space specified by a WDS design problem. Then one-pipe and two-pipe search approaches are employed to further polish the final solutions produced by the evolutionary search method.

Four WDS case studies with number of decision variables ranging from 21 to 454 were used to assess the performance of the HD-DDS, and it was reported by Tolson et al. (2009) that the HD-DDs exhibited a superior performance than GA and ACO methods in terms of efficiency and solution quality. In addition, it was found by Tolson et al. (2009)

that the local search methods were effective at improving the final solutions obtained by the evolutionary search method of the HD-DDS approach.

Krapivka and Ostfeld (2009) proposed a coupled GA-LP scheme for the least-cost pipe sizing of water networks. In this method, the optimization problem is decomposed into an “inner” and an “outer” problem. The “inner” LP is formulated and solved for a fixed set of flows, while the flows are altered in the “outer” using a GA. In their proposed optimization approach, an enumeration technique is initially used to identify all possible spanning trees for a looped water network. Then an LP solver is employed to optimize the pipe diameter sizes for each spanning tree to allow the least-cost tree to be determined. Lastly, the spanning tree chords are locked into the minimum permissible pipe diameters and the least-cost spanning tree is further optimized using the proposed coupled GA-LP technique.

The main advantage of this approach is that the search space handled by the GA-LP is reduced as the chords of the spanning tree are set to be the minimum allowable pipe sizes and removed as decision variables. However, this approach is computationally expensive for finding the least-cost spanning tree since all possible spanning trees need to be evaluated. The method is therefore limited in practical applications by the fact that it is impossible to evaluate all the spanning trees for a relatively large water network, and the global optimal solution for the original water network could be missed as the spanning tree chords are fixed by the minimum allowable pipe sizes in this method. An additional criticism is that a split-pipe approach is used in the proposed optimization technique.

Cisty (2010) proposed another combined GA and LP (GA-LP) model for solving WDS design problems. In this proposed GA-LP method, a GA is used to generate branched networks for a complex looped network, and LP is used to optimize each branched network. The proposed GA-LP method utilizes the fact that the LP is suitable for solving branched networks and GA is effective in dealing with networks with a small number of decision variables. This GA-LP was tested on three WDS case studies and proven to be

robust and efficient. However, split pipe solutions are still included in the final optimal solution, which is a severe limitation for practical application.

Haghighi et al. (2011) combined a simple GA with BLP for a WDS optimization design. In this GA-BLP method, a water network is first converted to a tree by removing one pipe from each primary loop and hence a total of NL pipes are removed, where NL is the number of loops in the water network. Then a set of N diameter combinations for the NL pipes is randomly generated using commercially available pipe diameters to form the initial population of the GA, where N is the population size of the GA. For each individual in the GA with different diameter combinations for the NL pipes, an iterative procedure using BLP combined with a hydraulic solver (EAPNET) is used to optimize the remaining tree (the NL pipes are not included in the BLP optimization).

The optimum pipe diameters obtained from the iterative BLP optimization for the tree are returned to the GA along with the corresponding cost. This cost in combination with the cost of the NL pipes handled by the GA provides the total cost of the original water network. This total cost is used to calculate the fitness of the GA individual. Subsequently, the GA operators (selection, crossover and mutation) are performed to evolve the initial solutions to achieve the final optimal solutions.

In the GA-BLP method (Haghighi et al. 2011), the GA is only used to deal with the NL pipes, while BLP is employed to tackle the optimization of the tree that was obtained by removing NL pipes. Thus, efficiency of the GA optimization is expected to be improved as the GA only handles NL pipes rather than the total number pipes in the original whole network (NL is normally significantly smaller than the total number pipes). However, the computational effort required for iterative BLP optimization in this GA-BLP approach is massive when dealing with large water networks since BLP has previously been found to be extremely inefficient when tackling large optimization problems (Savic and Cunha 2006; Martínez 2006).

2.4.2 Providing EAs with good estimates

The efficiency of an EA is improved if its search is initialised with good starting points since it requires less time to find optimal solutions. This approach allows the EA to focus on exploring the neighboring region that is specified by the good initial estimates, thereby speeding up the convergence speed. In this approach, an EA or traditional optimization technique is first used to explore the search space in an approximate way in order to identify an approximate optimal solution. The resulting solution is then used to seed another EA in order to attempt to locate better solutions. Since the EA is seeded with good initial estimates, better solutions can be generated at a low computational cost. This has been demonstrated in a number of studies (Grefenstette 1987, Harik and Goldberg 2000).

Keedwell and Khu (2006) proposed an optimization approach that combined a local representative cellular automata (CA) and a GA (CANDA-GA) for optimizing the design of WDSs. In CANDA-GA, the CA is used to find the approximate optimal solutions and the GA is seeded with these approximate optimal solutions in order to reach better solutions. It was reported by Keedwell and Khu (2006) that the CANDA-GA showed significant improvement in efficiency compared to a GA without any estimates based on two real-world WDS case studies with 632 and 1277 decision variables. However, premature convergence was observed by Keedwell and Khu (2006) for the CANDA-GA method.

2.4.3 Analysis of hybrid optimization techniques

The majority of currently available hybrid optimization techniques remain in the research domain due to their limitations. For the GA-LP method proposed by Krapivka and Ostfeld (2009), it is extremely inefficient to find the least-cost spanning tree for the looped water network. For the GA-LP method developed by Cisty (2010), split pipe solutions are generated, which is not practical. For the GA-BLP method proposed by Haghighi et al. (2011), it is extremely inefficient to deal with the large tree network with BLP algorithm. For the HD-DDS method (Tolson et al. 2009), although its performance

is superior to that of GAs and ACO, it requires considerable computational resources when dealing with relatively large case studies. For example, HD-DDS used 30 million evaluations budget to find an optimal solution for a WDS case study with 454 pipes as shown in Tolson et al. (2009). For the CANDGA (Keedwell and Khu 2006), premature convergence is a concern and the efficiency needs to be further improved.

In a conclusion, the majority of currently available hybrid optimization techniques have disadvantages in terms of WDS optimization. This results in their application being limited in terms of their capacity to deal with real-world sized WDS design problems.

2.5 Graph theory applications in water network design

Normally, a WDS can be viewed as a connected graph $G(V,E)$, where V is a set of links and E is a set of nodes in the water distribution network. Thus, it is natural to introduce graph theory algorithms to enable the WDN analysis. Traditionally, graph theory has been used for water network connectivity and reliability analysis. **Gupta and Prasad (2000)** used the linear graph theory for analysis of pipe networks. **Deuerlein (2008)** proposed a graph theory algorithm to decompose the WDN into forest, bridges and blocks. This method provides a tool to simplify complex WDNs and provides a better understanding of the interactions between their different parts.

In terms of WDS design optimization, **Kessler et al. (1990)** developed a graph theory based algorithm to optimize the design of WDSs. In their work, the design process consists of three distinct stages. In the first stage alternative paths are allocated using graph theory algorithms. In the second stage the minimum hydraulic capacity (diameters) of each path is determined using a LP model. In the third stage the obtained solution from the second stage is tested by a network solver for various demand patterns.

Sonak and Bhawe (1993) introduced a combined graph decomposition-LP algorithm for WDN design. In this combined algorithm, all the trees of the looped WDS are first identified by a graph theory algorithm and optimized by a LP, allowing the global optimum tree solution to be located. The final optimal solution for the original WDS is

then determined by assigning the chords of the global optimum tree the minimum allowable pipe diameters. **Savic et al. (1995)** used graph theory to partition the water network into ‘tree’ and ‘co-tree’ to enable an optimization problem that involved minimising the heads by setting regular valves.

Kadu et al. (2008) proposed a genetic algorithm (GA) combined with a graph theory algorithm to optimize water distribution systems. In their method, graph theory is used to identify the critical path for each node in order to reduce the search space for the genetic algorithm. **Krapivka and Ostfeld (2009)** proposed a coupled GA-LP scheme for the least-cost pipe sizing of water networks. A spanning tree identification algorithm is introduced in their work.

Improvements in terms of efficiency and solution quality have been consistently reported by the researchers when these optimization techniques are combined with graph theory algorithms and applied to WDS case studies. Graph theory is normally used to identify the critical path or the spanning tree for the WDN in the majority of graph theory based optimization techniques.

2.6 Research gaps

Based on the literature review, areas in the field of the WDS optimization that would benefit from further investigation are as follows.

- Although a number of optimization techniques have been successfully applied to optimize the design of WDSs, limitations exist for each of them when dealing with real-world water networks. Traditional optimization techniques (LP, NLP and BLP) often converge at local optimal solutions due to the nonsmoothness of the search space of the WDS optimization problem. EAs require a large number of network evaluations to find optimal solutions, resulting in an expensive computational overhead, especially for relatively large case studies. In addition, the solution quality found by EAs is inferior when dealing with relatively larger WDSs.

- The majority of currently available hybrid optimization techniques (combine EAs with deterministic methods) remain in the research domain due to their limitations. This makes it difficult, if not impossible, for these hybrid optimization techniques to tackle the optimization problems of real-world WDSs.
- Although graph theory has been used in work with water networks, the majority of the studies involving graphs and WDSs are concerned with network connectivity analysis. A few attempts have been undertaken to conduct the optimization of WDSs with the incorporation of graph decomposition techniques. However, these methods still remain in the research domain and cannot be used to deal with the real-world WDS optimization due to their severe limitations.
- The majority of optimization techniques have been evaluated using small benchmark WDS case studies. It is desirable to assess the performance of these and new techniques using relatively large or real-world sized water networks. There is no clear definition yet on the typical number of pipes for a real-world sized water network. In this research, we made the assumption that WDSs that have 100 pipes or more are considered to be real-sized water networks.

The research outlined in this thesis has been undertaken to address these current shortcomings in WDS design, as explained in the Section 1.1 of Chapter 1.

Chapter 3. Journal Paper 1-Dynamically Expanding Choice Table GA

3.1 Synopsis

A dynamically expanding choice table approach to genetic algorithm optimization of water distribution systems

Genetic algorithms (GAs) have been frequently used to find optimal solutions for the water distribution system (WDS) design. A significant issue within the use of the GA when dealing with the WDS optimization problem is the intensive computational overhead. This has been addressed in Section 2.3.4 of Chapter 2. Thus, it is desirable to improve the efficiency of the GAs, especially when dealing with real-world water networks, for which a large number of pipes are involved. A dynamically expanding choice table GA is developed in this research in order to enhance the search efficiency of the GAs.

Typically, all available diameters in the complete choice table for a decision variable are considered as potential choices for each pipe of the network when a GA is applied to optimize a WDS design. An example of a typical choice table is given in Table 3.1. Binary coding and integer coding for each pipe size are shown in the second and third columns respectively. If there are a total of eight different diameters in a choice table for a pipe, the GA will generally have a random selection of the eight different diameters in the population of GA strings. The GA starts by exploring the entire solution space in order to reach the lowest cost solution. All regions within the solution space are considered to be equally important in the conventional GA, and hence, much computational effort is wasted on investigating infeasible or unnecessarily high cost regions within the search space.

In this Chapter, a dynamically expanding choice table method is proposed to reduce the search space so that the GA can concentrate on promising regions of the search space. Initially, all the diameters in the full choice table are sorted from the smallest to largest and each pipe is given a diameter choice table.

Table 3.1 A typical choice table

Nominal Diameter (mm)	Binary Coding	Integer Coding	Unit Cost (\$/m)
150	000	1	49
200	001	2	63
250	010	3	95
300	011	4	133
375	100	5	171
450	101	6	220
500	110	7	270
600	111	8	330

In this newly proposed method, only a small portion of pipe sizes in the full choice table for each pipe are used to generate solutions randomly in the initial population of the GA. During the run of the GA, if most of the members of the population in a generation select the smallest diameter for a particular pipe from its corresponding reduced size choice table, a smaller diameter is added to the pipe's choice table and the choice table has been dynamically expanded.

To the contrary, if most of the members of the GA population prefer the largest diameter for a pipe from its reduced size choice table, a larger diameter is added to expand the current choice table for this pipe. As a result, each pipe selects its own tailored choice table in the later generations of the GA. If the majority of members in the population select the smallest or largest diameter for a particular pipe at the extremity of the full choice table, this pipe is locked in to be the smallest or largest pipe size and is then removed as a decision variable.

This work has been published on *Journal of Water Resources Planning and Management* and the paper is provided here.

Citation of Paper

Zheng, F., A. R. Simpson, and A. C. Zecchin (2011b). "Dynamically expanding choice-table approach to genetic algorithm optimization of water distribution systems." *Journal of Water Resources Planning and Management*, 137(6), 547-551

STATEMENT OF AUTHORSHIP (PUBLISHED)

Journal paper title: **A dynamically expanding choice table approach to genetic algorithm optimization of water distribution systems.**

Authors: Feifei Zheng¹, Angus Simpson² and Aaron Zecchin³

Corresponding author:

¹Feifei Zheng (Candidate): PhD student, School of Civil, Environmental and Mining Engineering, University of Adelaide

Wrote the manuscript, performed all analysis, developed the model and theory and acted as corresponding author.

I hereby certify that the statement of contribution is accurate.

Signed _____ Date _____

²Angus Simpson: Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide

Supervised the development of the model and reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

³Aaron Zecchin: Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide

Reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

Journal of Water Resources Planning and Management, 137(6), 547-551, 2011.

3.2 Journal Paper 1: A dynamically expanding choice table approach to genetic algorithm optimization of water distribution systems (Published in the *Journal of Water Resources Planning and Management*)

Feifei Zheng, Angus R. Simpson and Aaron C. Zecchin

ABSTRACT

This paper proposes a modified genetic algorithm (GA) for optimization of water distribution systems. A method of dynamically expanding pipe choice table selections and reducing the number of decision variables is introduced that occurs during a GA run. Based on the progressive selection, an initially reduced size choice table for each decision variable is allowed to dynamically expand and then the number of decision variables is gradually reduced. This process enables the GA search to concentrate on promising regions of the search space. The dynamically expanding choice table genetic algorithm (GA_{DECT}) has been applied to a benchmark case study, the New York Tunnels Problem. The results obtained show that the GA_{DECT} yields a superior performance in terms of solution quality and computational efficiency.

CE Database subject headings: Optimization; Water distribution systems; Algorithms.

INTRODUCTION

Evolutionary algorithms have been introduced over the last 15 years to seek the least-cost design of water distribution systems. Among them, genetic algorithm (GA) optimization has gained popularity in terms of optimal design of water distribution systems because of its robustness and search performance (Simpson et al. 1994; Savic and Walters 1997). Many methods have been developed by researchers to improve the performance of GAs. A creeping mutation operator, variable power scaling of the fitness function and Gray coding (Dandy et al. 1996) were incorporated into the GA and were shown to be more effective. Vairavamoorthy and Ali (2000) applied integer coding in GAs to avoid the problem of redundant states often found when using binary or Gray codings. Wu et al. (2001) introduced a fast messy genetic algorithm to deal with optimization of water

networks, showing significant improvement in terms of efficiency and robustness. Vairavamoorthy and Ali (2005) used a pipe index method to modify GA-based pipe optimization. Other evolutionary optimization approaches have also been developed. Eusuff and Lansey (2003) proposed a shuffled frog leaping algorithm (SFLA) which showed improvement on the convergence speed in the context of optimal design of water distribution systems. Maier et al. (2003) applied ant colony optimization approach to optimize water distribution systems. Zecchin et al. (2006) proposed a Max-Min Ant System optimization (MMAS) and compared results obtained by GAs.

THE MODIFIED GENETIC ALGORITHM

Dynamically expanding choice tables

Typically all available diameters in the complete choice table for a decision variable are considered as potential choices for each pipe of the network when a GA is applied to optimize a WDS design. All regions within the solution space are considered to be equally important in the conventional GA, and hence, much computational effort is wasted on investigating infeasible or unnecessarily high cost regions within the search space.

In this research, a dynamically expanding choice table method is proposed to reduce the search space so that the GA can concentrate on promising regions of the search space. Initially, all the diameters in the full choice table are sorted from the smallest to largest and each pipe is given a diameter choice table. In the new method, only a subset of pipe sizes in the full choice table for each pipe (say the 3 successive middle sizes) are used to generate solutions randomly in the GA's initial population. During the GA run, if most of the members of the population in a generation have taken on the smallest diameter for a particular pipe from its corresponding reduced size choice table, this implies that this pipe diameter potentially can be further reduced in size to further reduce the cost of the whole network. Consequently, a smaller diameter is added to the pipe's current choice table and the choice table has been dynamically expanded. The same principle can be

applied to the larger diameter options in a choice table. As a result, each decision variable in terms of pipe diameter size selects its own tailored choice table in the later generations.

Reduction of the number of decision variables

If the majority of members in a population select the diameter size for a particular pipe at the extremity of the full choice table, this pipe is locked to be the selected pipe size and then removed as a decision variable (whether it is either the smallest or largest diameter options). This process is used to dynamically remove such decision variables that cannot be further evolved as they have already converged at one extremity of the choice table. Therefore, the GA is able to more effectively and efficiently search the reduced search space, and focus on regions that show promise.

In summary, there are five cases that may occur for a choice table as shown in Fig. 1. Assume that the full choice table is made up of pipe diameters D1 to D10 ranked from the smallest to the largest diameter. An initial reduced size choice table including D5, D6 and D7 (middle column in Fig. 1) is used to randomly generate the initial population of GA.

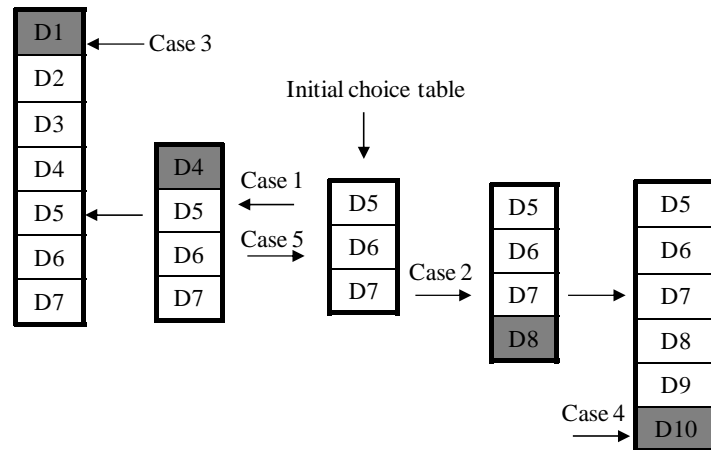


Figure 1 An example of expanding of a choice table and reduction of decision variables

The following threshold percentages are defined: (1) for expanding the choice table (P_e) (2) for removing decision variables (P_r) and (3) for when the majority of population

members select the middle size of the current choice table during the GA run (P_s). Five cases are given as follows.

Case 1: For a particular pipe, if more than P_e percent of the members in a population select the smallest size (D5) of the current choice table (middle column of Fig. 1), a smaller pipe size D4 is *added* to the choice table (the second column from left in Fig. 1). Diameters of D4 and D5 are then randomly reselected for this pipe for all the members in the GA population.

Case 2: If more than P_e percent of the members in a population select the largest size (D7) of the current choice table (middle column of Fig. 1), a larger pipe size D8 is *added* to the choice table (the second column from right in Fig. 1). Diameters of D7 and D8 are then randomly reselected for this pipe for all the members in the GA population.

Now consider the situation where the choice table has been eventually expanded to include either the smallest or largest pipe:

Case 3: If more than P_r percent of the members in a population select the smallest size (D1) of the choice table (the first column from left in Fig. 1), this pipe is *removed* as a decision variable and the diameter for this pipe is locked at the minimum pipe size (D1).

Case 4: If more than P_r percent of the members in a population select the largest size (D10) of the choice table (the last column on the far right in Fig. 1), this pipe is *removed* as a decision variable and the diameter for this pipe is locked at the maximum pipe size (D10).

Now consider the situation where the majority of the pipes are the pipe size from the middle of the current choice table for that pipe:

Case 5: If more than P_s percent of population members select the middle size (D6) of the current choice table for a particular pipe during the GA run, all the pipe sizes in the current choice table are randomly *reselected* for this pipe in all members of the whole population. This process is used to maintain the

population diversity, as occurs with the common mutation operator. However, case 5 is quite different from the normal mutation operator in that it only occurs when most of the population members select the middle pipe size diameter from its corresponding choice table.

CASE STUDY

The dynamically expanding choice table genetic algorithm (GA_{DECT}) was developed in C++ and combined with the EPANET2 hydraulic network solver. A total of 1000 independent optimization runs based on different random number seeds have been performed for New York Tunnel Problem (NYTP). The parameters settings used in GA_{DECT} are given in Table 1. Constraint tournament selection was used in GA_{DECT} (Deb 2000).

Table 1 GA_{DECT} parameter values for the NYTP case study

Parameter	Value
Population size (N)	100
Maximum number of evaluations	100,000
Probability of crossover (P_c)	0.9
Probability of bitwise mutation (P_m)	0.0
Threshold percentage for expanding the choice table (P_e)	65%
Threshold percentage for removing decision variables (P_r)	95%
Threshold percentage for reselection (P_s)	70%

Case Study: New York Tunnels Problem

The New York Tunnels Problem (NYTP) has 21 existing tunnels and 20 nodes fed by the fixed-head reservoir. Details of this network, including the layout, the head constraints, pipe choices and costs, and water demands can be found in Dandy et al. (1996). The objective is to determine which pipes should be installed in parallel with the existing pipes to minimize the cost while satisfying the minimum head requirement at all nodes. The entire choice table for the NYTP case study involved 16 choices of pipe diameters consisting of {0, 36, 48, 60, 72, 84, 96, 108, 120, 132, 144, 156, 168, 180, 192, and 204} inches.

An initial choice table with the diameters of {48, 60, 72} inches for each pipe was used to seed the initial population in the GA_{DECT} for the NYTP case study. One requirement of the proposed GA_{DECT} is that the threshold percentages (P_e , P_r and P_s) need to be specified. As given in Table 1, the parameter settings for the NYTP case study were as follows: $P_e=65\%$ (that is, expansion of the choice table occurred if more than 65% of the members selected the largest or smallest pipe size for a pipe from its choice table); $P_r=95\%$ (that is, if more than 95% of the members for a particular pipe have selected the smallest or the largest diameter size, this pipe is locked in to be the smallest or largest diameter and then removed as a decision variable); $P_s=70\%$ (that is, if more than 70% of the members selected a particular middle size for a pipe from its choice table, all the sizes in the current choice table are randomly reselected for this pipe for the whole population). An example of the initial choice table and the final choice table for a typical GA_{DECT} run applied to the NYTP, after dynamic expansion plus the decision variable removal, are shown in Table 2.

As can be seen from Table 2, the second column is the initial choice table of {48, 60, 72} inches for diameters for each pipe and the third column is the final choice table for each pipe at the end of GA run. The final column is the least-cost solution found by the GA_{DECT} with a cost of \$38.64 million (the current best known-least-cost solution). It is observed from Table 2 that choice tables for individual pipes were expanded differently during the GA run, despite the fact that they all started with the same initially reduced size choice table. The pipes labeled with a hash were removed as decision variables, as a pipe size of zero was selected during the GA run. From column 3 of Table 2, the total search space covered by the GA_{DECT} is given by $5^9 \times 7^9 \times 9^2 \times 11 \approx 7.0224 \times 10^{16}$, which is only a small fraction ($3.62 \times 10^{-9}\%$) of the size of the original solution space.

As can be seen from Table 2, some pipes (such as pipe 4, 6, 10, 11, 12, 13, 14, 15 and 20) moved towards the smaller pipe sizes during the GA run and finally were dropped as decision variables with a pipe size of zero, indicating that it was not economic for these pipes to be duplicated. However, several pipes (such as pipe 7, 16, 17, 18, 19, 21) were

assigned larger sizes within the GA process, implying that these pipes were the potential candidates for duplication. It is noted that choice tables of some pipes (such as pipe 1, 2, 3, 5, 8, 9) expanded to larger diameters at the beginning and then to smaller diameters afterwards, showing that these pipes were indentified to be potential duplicates initially, but were eliminated from consideration in the later generations of the GA.

Table 2 An example of the expansion of choice tables and removing decision variables during the GA_{DECT} process applied to the NYTP

Links	Choice table for pipe diameters (inches)		Final solution (inches)
	Initial choice	End	
1 [#]	48, 60, 72	0, 36, 48, 60, 72, 84, 96	0
2 [#]	48, 60, 72	0, 36, 48, 60, 72, 84, 96	0
3 [#]	48, 60, 72	0, 36, 48, 60, 72, 84, 96	0
4 [#]	48, 60, 72	0, 36, 48, 60, 72	0
5 [#]	48, 60, 72	0, 36, 48, 60, 72, 84, 96	0
6 [#]	48, 60, 72	0, 36, 48, 60, 72	0
7	48, 60, 72	48, 60, 72, 84, 96, 108, 120, 132, 144, 156, 168	144
8 [#]	48, 60, 72	0, 36, 48, 60, 72, 84, 96	0
9 [#]	48, 60, 72	0, 36, 48, 60, 72, 84, 96	0
10 [#]	48, 60, 72	0, 36, 48, 60, 72	0
11 [#]	48, 60, 72	0, 36, 48, 60, 72	0
12 [#]	48, 60, 72	0, 36, 48, 60, 72	0
13 [#]	48, 60, 72	0, 36, 48, 60, 72	0
14 [#]	48, 60, 72	0, 36, 48, 60, 72	0
15 [#]	48, 60, 72	0, 36, 48, 60, 72	0
16	48, 60, 72	48, 60, 72, 84, 96, 108, 120, 132, 144	96
17	48, 60, 72	48, 60, 72, 84, 96, 108, 120, 132, 144	96
18	48, 60, 72	48, 60, 72, 84, 96, 108, 120	84
19	48, 60, 72	48, 60, 72, 84, 96, 108, 120	72
20 [#]	48, 60, 72	0, 36, 48, 60, 72	0
21	48, 60, 72	48, 60, 72, 84, 96, 108, 120	72
Cost (\$M)			38.64

Pipe was locked in at zero size and eliminated as a decision variable during the GA_{DECT} process.

The dynamic reduction of the number of the decision variables for a typical GA_{DECT} run is shown in Fig. 2. At stage A in Fig. 2, there were 21 decision variables. After 16 generations (at stage B), pipe 11 was the first pipe dropped out as a decision variable with a size of zero. The following sequence of pipes involving 4, 10, 12, 13, 14, 15 and 20 were consecutively eliminated. Thus, only 13 decision variables were left at stage C

after 44 generations. Subsequently, pipes 2, 3, 5, 6 and 8 were removed as decision variables from stage C to D. After 176 generations (at stage E), only six decision variables were left, which were pipes 7, 16, 17, 18, 19 and 21. In the final stage, GA_{DECT} dealt with a reduced search space size and hence worked more efficiently.

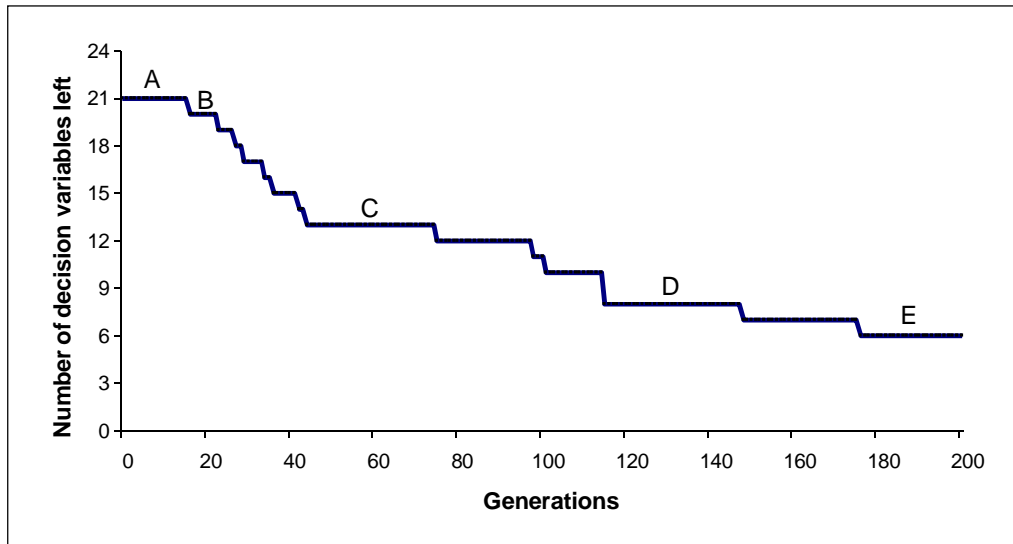


Figure 2 Example of dynamical reduction of number of decision variables

RESULTS AND DISCUSSION

For the NYTP cases study, the current best known solution with a value of \$38.64 million was first found by Maier et al. (2003) and this solution has been also found by the proposed GA_{DECT}. Fig. 3 gives a summary of a range of different sets of threshold values for GA_{DECT} applied to the NYTP case study. The GA_{DECT} program with each set of threshold values was performed for 1000 runs using different random number seeds. As can be seen from Fig. 3, GA_{DECT} with relatively high threshold percentages is able to find the best known solution with higher frequency, but at the expense of increased computational overhead. It was found that GA_{DECT} with $P_e=65\%$, $P_f=95\%$ and $P_s=70\%$ exhibited overall well with an appropriate balance between performance in terms of frequency that the best solution was found and computational efficiency based on 1000 different runs.

The results for GA_{DECT} ($P_e=65\%$, $P_r=95\%$ and $P_s=70\%$) runs are given in Table 3. In order to enable a comparison of performance, the results of other optimization techniques that have previously applied to the NYTP case study are also included in Table 3.

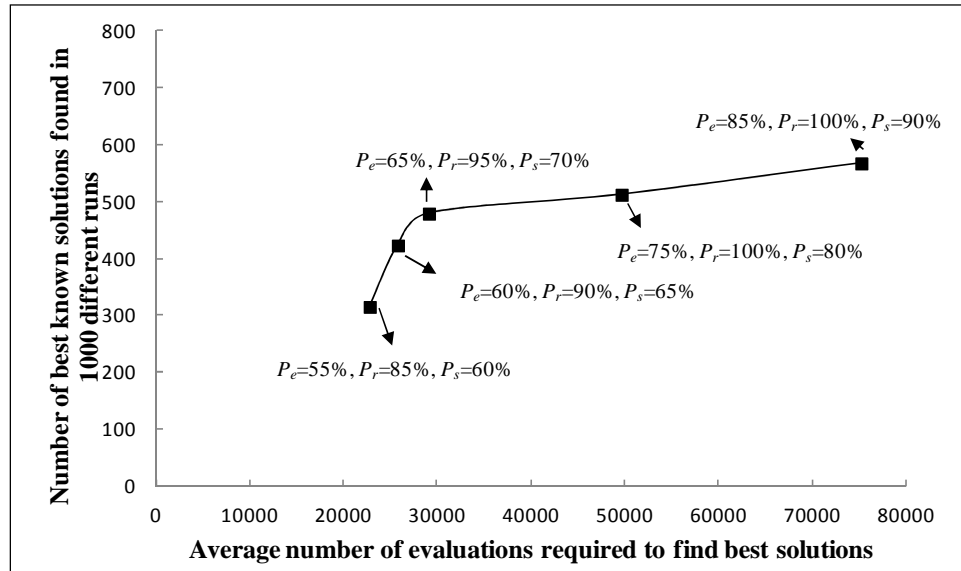


Figure 3 Results of GA_{DECT} with different set of parameter values applied to the NYTP case study

Table 3 Comparison of algorithmic performance applied to the NYTP case study

Algorithm	No. of runs	Best solution (\$M)	Average cost (\$M)	No. of average evaluations	No. of best solution found
GA_{DECT} ($P_e=65\%$, $P_r=95\%$ and $P_s=70\%$)	1000	38.64	39.06	29,101	479
Improved GA ¹	5	38.80	38.98	143,790	NA
MMAS ²	20	38.64	38.84	30,711	NA
ACO ³	3	38.64	NA	13,928	NA
Messy GA ⁴	5	38.80	39.09	48,427	NA
PSO ⁵	30	38.64	38.93	NA	10
DE ⁵	30	38.64	40.33	NA	22

¹Dandy et al. (1996). ²Zecchin et al. (2006). ³Maier et al. (2003). ⁴Wu and Simpson (2001).

⁵Dandy et al. (2010). NA means “not available”

The best solution found by Improved GA (Dandy et al. 1996) and Messy GA (Wu and Simpson 2001) was \$38.80 million, which deviates 0.414% from the best known solution. In terms of efficiency, the proposed GA_{DECT} outperformed the other optimization techniques, but had slightly more average evaluations than the ACO (Maier

et al. 2003). However, it is highlighted that there were only three different ACO runs used, whilst a total of 1000 different GA_{DECT} runs were performed in this study. The average cost solution produced by GA_{DECT}, based on 1000 different runs, is \$39.06 million, which only deviates 1.087% from the known-least-cost solution. Even though the average cost solution provided by MMAS (Zecchin et al. 2006) and particle swarm optimization (PSO) (Dandy et al. 2010) are slightly lower than that of GA_{DECT}, the number of random number seeds are only 20 and 30 respectively, The GA_{DECT} was able to locate the current best solution 479 times out of a total of 1000 different runs, a higher frequency in finding optimal solutions than the PSO but slightly lower than that found by DE (Dandy et al. 2010).

CONCLUSION

A dynamically expanding choice table approach has been developed to enhance the performance of GA optimization for water distribution systems. The proposed approach provides a guide for the GA search to focus within regions of good fitness values. Thus, the search time is reduced and the optimal solution is more likely to be found. It is noted that, from the results of NYTP case study, the GA_{DECT} performed better than, or at least as good as, other optimization techniques presented in this paper such as other GA variants, ACO, MMAS and PSO methods.

REFERENCES

- Dandy, G. C., Simpson, A. R., and Murphy, L. J. (1996). "An improved genetic algorithm for pipe network optimization." *Water Resources Research*, 32(2), 449-457.
- Dandy, G. C., Wilkins, A., and Rohrlach, H. (2010). "A methodology for comparing evolutionary algorithms for optimizing water distribution systems." Proceedings of the 12th *Water Distribution System Analysis Symposium*, Tucson, Arizona.
- Deb, K. (2000). "An efficient constraint handling method for genetic algorithms." *J Comput. Methods Appl. Mech. Engrg.*, 186,311-338.

- Eusuff, M. M., and Lansey, K. E. (2003). "Optimization of water distribution network design using shuffled frog leaping algorithm." *J. Water Resources Planning and Management*, 129(3), 210-225.
- Maier, H. R., et al. (2003). "Ant colony optimization for the design of water distribution systems." *J. Water Resources Planning and Management*, 129(3), 200-209.
- Simpson, A. R., Dandy, G. C., and Murphy, L. J. (1994). "Genetic Algorithms compared to other techniques for pipe optimization." *J. Water Resources Planning and Management*, 120(4), 423-443.
- Savic, D. A., and Waters G.A. (1997). "Genetic Algorithms for least-cost design of water distribution networks." *J. Water Resource Planning and Management*, 123(2), 67-77.
- Vairavamoorthy, K., and Ali, M. E. (2000). "Optimal Design of water distribution systems using genetic algorithms." *Computer-aided Civil and Infrastructure Engineering*, 15(2), 374-382.
- Vairavamoorthy, K., and Ali, M. E. (2005). "Pipe index vector: A method to improve genetic-algorithm-based pipe optimization." *J. Hydraul. Eng.*, 131(12), 1117-1125.
- Wu, Z. Y., Boulos, P. F., Orr, C. H., and Ro, J.J. (2001). "Using genetic algorithms to rehabilitate distribution systems." *J. Am. Water Works Assoc.*, 93(11),74-85.
- Wu, Z. Y., and Simpson, A. R. (2001). "Competent genetic-evolutionary optimization of water distribution systems." *Journal of Computing in Civil Engineering*, 15(2), 89-101.
- Zecchin, A. C., Simpson, A. R., and Maier, H. R. (2006). "Application of two ant colony optimization algorithms to water distribution system optimization." *J. Mathematical and Computer Modelling*, 44,451-668.

Chapter 4. Journal Paper 2-A Dither Creeping Mutation GA for WDS Optimization

4.1 Synopsis

A non-crossover dither creeping mutation genetic algorithm for water distribution system optimization

In Chapter 3, a dynamically expanding choice table genetic algorithm (GA_{DECT}) was proposed to optimize the design for water distribution systems (WDSs). It was demonstrated in Chapter 3 that the efficiency of the GA_{DECT} is improved compared to the standard GA. However, a number of parameter values need to be calibrated for the GA_{DECT} , which causes difficulties for the practicing water engineers wanting to implement the GA_{DECT} to tackle real-world sized WDS. This is because it is time consuming to tune the parameter values for evolutionary algorithms and specific knowledge is required to determine the appropriate parameter values (see discussion in Section 2.3.4 of Chapter 2)

In this current research, a non-crossover dither creeping mutation-based genetic algorithm (CMBGA) for water distribution system (WDS) optimization is developed and analyzed. This CMBGA differs from the classic GA optimization in that it does *not* utilize the crossover operator, but instead only uses selection and a proposed dither creeping mutation operator. The creeping mutation rate in the proposed dither creeping mutation operator is randomly generated in a range rather than being set to a fixed value. In addition, the dither mutation rate is applied at an individual chromosome level rather than the generation level. The dither creeping mutation probability is set to take values from a small range that is centered about $1/ND$ (ND =number of decision variables of the optimization problem being considered). The reason for adopting this range is that a mutation probability of $1/ND$ has been demonstrated to be an effective value and is normally used for the GA.

Genetic algorithms have usually been previously thought to be highly dependent on crossover. The research reported in this paper shows clearly that the performance of the

GA can be achieved without crossover and that mutation, used in the right way, is just as effective. This is the first known work to develop a non-crossover and mutation only based genetic algorithm for WDS design.

An important objective of this research is to compare the performance of the proposed CMBGA to other standard GA variants. This systematic comparison amongst GA variants has proven important, and it serves to highlight the relative importance of the GA mechanisms of mutation and crossover in yielding an effective search. The proposed GA has shown significant improvements compared to four other GA variants in terms of the quality of the optimal solutions based on four WDS case studies used in this research. Thus the non-crossover dither creeping mutation based GA is a preferred tool for water distribution system optimization in contrast to standard GA variants. Additionally, another advantage of the proposed CMBGA over other GA variants is that it does not involve as much elaborate tuning of the parameter values.

This work has been published on *Journal of Water Resources Planning and Management* and the paper is provided here.

Citation of Paper

Zheng, F., A. R. Simpson, A. C. Zecchin, M. F. Lambert (2013). “A non-crossover dither creeping mutation genetic algorithm for pipe network optimization.” *Journal of Water Resources Planning and Management*, doi: 10.1061/(ASCE)WR.1943-5452.0000351.

STATEMENT OF AUTHORSHIP (PUBLISHED)

Journal paper title: **A non-crossover dither creeping mutation genetic algorithm for pipe network optimization.**

Authors: Feifei Zheng¹, Aaron Zecchin², Angus Simpson³, and Martin Lambert⁴

Corresponding author:

¹Feifei Zheng (Candidate): PhD student, School of Civil, Environmental and Mining Engineering, University of Adelaide

Wrote the manuscript, developed the model and acted as corresponding author.

I hereby certify that the statement of contribution is accurate.

Signed _____ Date _____

²Angus Simpson: Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide

Supervised the development of the model and reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

³Aaron Zecchin: Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide

Supervised the development of the model and reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

⁴Martin Lambert: Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

Journal of Water Resources Planning and Management, doi: 10.1061/(ASCE)WR.1943-5452.0000351

4.2 Journal Paper 2: A non-crossover dither creeping mutation genetic algorithm for pipe network optimization (Published in the *Journal of Water Resources Planning and Management*)

Feifei Zheng, Aaron C. Zecchin, Angus R. Simpson and Martin F. Lambert

Abstract

A non-crossover dither creeping mutation-based genetic algorithm (CMBGA) for water distribution system optimization has been developed and is analyzed. CMBGA differs from classic GA optimization as it does not utilize crossover, but instead only uses selection and dither creeping mutation. The creeping mutation rate is randomly generated in a range rather than being set to a fixed value for each individual. An objective of this paper is to compare the performance of the CMBGA with four other GA variants. The results based on four case studies show that the CMBGA exhibits considerable improvement over the considered GA variants. The CMBGA shows a very significant improvement in optimization for the Hanoi Problem and the Go Yang network compared to all previously published results. A main advantage of the proposed CMBGA over the majority of the other evolutionary algorithms is that it avoids the need for an extensive parameter calibration phase.

Keywords: Optimization; water distribution systems; genetic algorithms; creeping mutation; dither mutation.

1. Introduction

The non-linear constraints and the discrete combinatorial decision space of water distribution systems (WDSs) bring a significant challenge when optimizing their design. A number of optimization techniques have been previously applied to optimal water network design, such as complete enumeration (Gessler 1985), linear programming (Alperovits and Shamir 1977, Morgan and Goulter 1985, Fujiwara et al. 1987) and non-linear programming (Lansey and Mays 1989, Fujiwara and Khang 1990). The complete enumeration approach is able to guarantee that the global optimal solution is reached.

However, the computational overhead is huge since all possible solutions need to be evaluated. Other optimization techniques, such as linear or non-linear programming are often trapped at local optimal solutions.

Evolutionary algorithms (EAs) have been used to optimize WDSs since the early 1990s. Examples of these studies include the following: Murphy and Simpson (1992) introduced genetic algorithms (GAs) to the water community, specifically for water network optimization; Cunha and Sousa (2001) used simulated annealing to optimize WDSs; Geem et al. (2002) developed a harmony search model for optimizing WDSs; Eusuff and Lansey (2003) proposed a shuffled frog leaping algorithm (SFLA) for network optimization; Maier et al. (2003) applied an ant colony optimization approach to optimize WDSs; Tolson et al. (2009) developed a hybrid discrete dynamically dimensioned search (HD-DDS) approach to optimize the WDSs; and Suribabu (2010) employed the differential evolution (DE) to the optimization of WDSs. These techniques have been successfully applied to a number of optimization problems and have been demonstrated to be more effective in finding optimal solutions compared with traditional optimization techniques.

Amongst these EAs, GAs have gained popularity due to their ease of implementation and search ability (Simpson et al. 1994, Savic and Walters 1997)). Much research has been undertaken to enhance the performance of GAs. A creeping mutation operator, variable power scaling of the fitness function and Gray coding were incorporated into the GA and were shown to be more efficient (Dandy et al. 1996). Vairavamoorthy and Ali (2000) applied integer coding in GAs to avoid the problem of redundant states often found when using binary or Gray codings. Wu and Simpson (2001) introduced a fast messy genetic algorithm (fmGA) to deal with the optimization of water networks, showing significant improvement in terms of efficiency and robustness. A pipe index method proposed by Vairavamoorthy and Ali (2005) was able to guide the GA search into the promising regions, thus enabling the GA to provide optimal solutions in less search time. Zheng et al. (2011a) developed a modified GA for water distribution system design. In their work,

a method of dynamically expanding pipe choice table selections and reducing the number of decision variables was introduced to improve GA's performance.

The GAs used by the water community are derived from Holland (1975), with the crossover operator being considered to be the dominant operator while mutation has been considered to be a second order operator. Thus high crossover probabilities and low mutation probabilities have been suggested for a better performance of GAs for the optimization of WDSs (Simpson et al. 1994, Savic and Walters 1997, Deb 2001). A typical parameter combination for GA optimization WDS is a crossover probability of 0.9 and a mutation probability of 0.01 (Simpson et al. 1994, Dandy et al. 1996, Vairavamoorthy and Ali 2000).

In contrast, some other EAs such as Evolutionary Strategy (ES) (Rechenberg 1965) and Evolutionary Programming (EP) (Fogel et al. 1966) have concentrated on mutation as the main driving evolution operator. ES algorithms with adaptive mutation rates have been found to be effective when dealing with some optimization tasks (Rechenberg 1965). Fogel and Atmar (1990) strongly suggested that crossover has no general advantage over mutation. As a result, mutation-based GAs have been proposed to solve some optimization problems (Falco et al. 2002, Dai et al. 2002). Although there exists a large body of conventional wisdom concerning the relative importance of crossover and mutation, no explicit conclusion has been made on this issue to date. In addition, it is reported in Spears (1993) that the relative importance of crossover and mutation is heavily dependent on the properties of the fitness landscape associated with the optimization problem. Consequently, conclusions that have been made based on other optimization problems cannot necessarily be easily transferred to WDS optimization directly.

This paper aims to develop and investigate a non-crossover dither creeping mutation-based GA (CMBGA) to optimize the design of WDSs. The term dither comes from differential evolution algorithm as its use will be explained later on (Das et al. 2005). In the proposed CMBGA, only the selection and dither creeping mutation operators are

applied. The performance of the proposed CMBGA is assessed in this paper using a large range of different random number seeds.

2. Background analysis

The most commonly used GA in the water community is a crossover-based GA. A typical parameter combination for a GA optimization WDS is a crossover probability of 0.9 and a mutation probability of 0.01 (Simpson et al. 1994; Dandy et al. 1996; Vairavamoorthy and Ali 2000). This version of the GA is based on Holland (1975) and uses bitwise mutation of strings while crossover is used as the primary search mechanism. Mutation has traditionally been viewed as secondary operator while being considered only useful in maintaining diversity of the population. Holland formalized his GA using the Schema Theorem (the theory of building blocks) to provide a theoretical background justification of crossover. The building block hypothesis has often been used as a basis for theoretical and experimental work on GAs (Goldberg 1989, Deb 2001). Goldberg (1989) suggested that crossover was the dominant operator in GAs as it was able to efficiently assemble the short, low-order and high performance schemata or building blocks. Syswerda (1989) has argued that the building block hypothesis lacks theoretical justification. Fogel (2000) and Zheng et al. (2010) found that uniform crossover outperformed two-point crossover, which in turn outperformed one-point crossover on many of the optimization case studies. Those results contradict the building blocks hypothesis that one-point and two-point crossover should perform better than uniform crossover because they are much less disruptive of the short and low schemata.

The critical issue that exists when comparing the crossover- and mutation-based GAs is as to which operator is the dominant operator for driving evolution. Much work has been done previously to identify the dominant operator in GAs (Vose 1994, Palmes et al. 2005). Proponents of Holland's version of GAs have claimed that crossover is a more powerful operator compared with mutation based on a number of experimental results (Schaffer and Eshelman 1991). In contrast, others have asserted that mutation is the dominant operator for driving evolution (Vose 1994, Palmes et al 2005). However, the

empirical comparisons made by one group of researchers have often been disputed by other groups, and no theoretical justification has been universally claimed on this issue.

Spears (1993) asserted that there were some important individual characteristics of each of these two operators that were not captured by the other operator. Spears (1993) conducted his analysis based on hyperplanes (building blocks) and defined two potential roles of a genetic operator: disruption (exploration) and construction (exploitation). He provided a theoretical justification that crossover was more effective for constructing high order building blocks from lower building blocks in comparison to mutation, indicating that crossover was more powerful in terms of construction; while mutation was more powerful in terms of disruption. Crossover emphasizes the evolutionary information exchange between individuals, thus it is able to maximize accumulated payoff and exhibits high simultaneous levels of preservation, indicating more exploitation. In contrast, mutation emphasizes preservation of the behavior links between parent and offspring, thus it provides higher levels of exploration. Therefore the mean behaviour of a GA with crossover outperformed the mean behaviour with a GA without crossover while a GA without crossover outperformed a GA with crossover in terms of seeking optimal solutions (Fogel and Atmar 1990). Thibert-Plante and Charbonneau (2007) found that crossover was not particularly helpful in producing better solutions, while it markedly improved the overall evolutionary stability. Zheng et al. (2010) demonstrated that a GA without mutation or an extremely low mutation rate tended to converge prematurely.

The mutation operator used in the Holland's GA has been viewed to be oversimplified by some researchers (Vose 1994) and can be greatly modified to enhance its performance (Spears 1993). Falco et al. (2002) employed a modified-mutation-based GA to deal with several test functions and a comparison was made between the modified-mutation-based GA and the traditional crossover-based GAs. The results achieved showed that the modified-mutation-based GA outperformed the traditional crossover-based GA. Dai (2002) developed a non-crossover GA and applied it to the travelling salesman problem

(TSP) and showed that the non-crossover GA outperformed the crossover-based GA in terms of solution quality and efficiency.

As the non-crossover GA shows potential merits in other field, it is desirable to provide an analysis for the non-crossover GA applied to WDS optimization problems. This is the first known work in terms of applying the non-crossover GA to WDS design.

3. The proposed non-crossover dither creeping mutation-based GA

The non-crossover dither creeping mutation genetic algorithm (CMBGA) proposed in this paper is characterized by the fact that crossover is not used. Additionally, a dither creeping mutation operator is introduced into the CMBGA to replace the commonly used bitwise mutation operator. A flowchart of the proposed CMBGA applied to WDS optimization is illustrated in Figure 1 and the details of the proposed CMBGA are discussed in the following sections. The CMBGA run is stopped when the criterion is satisfied. In the proposed CMBGA, a maximum number of allowable evaluations is used as the stopping criterion.

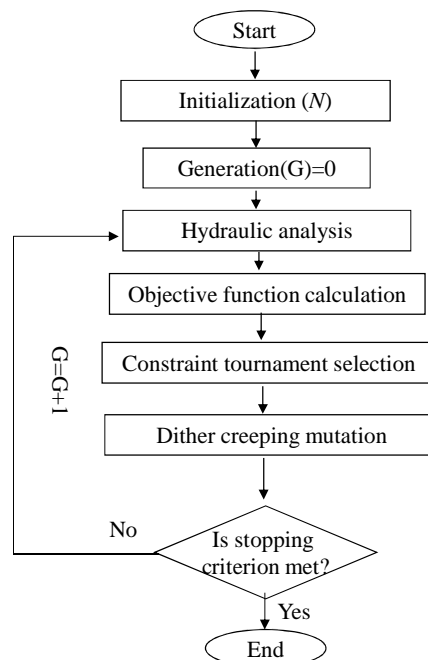


Figure 1 Flowchart of the proposed CMBGA

3.1 Initialization

An initial random population of N solutions is generated by uniformly randomizing individuals within the search space as:

$$x_{i,0}^j = U_i^j(0, K - 1) \quad j=1, 2, \dots, ND, i=1, 2, \dots, N \quad (1)$$

where $x_{i,0}^j$ represents the initial value of the j^{th} parameter in the i^{th} individual at the initial population, where an individual is given by $X_{i,0} = [x_{i,0}^1, x_{i,0}^2, \dots, x_{i,0}^D]^T$. U_i^j represents a randomly generated integer variable within the range of 0 to $K-1$ for the j^{th} parameter in the i^{th} individual. The symbols N , ND and K are population size, number of decision variables and number of pipe diameter choices respectively.

3.2 Hydraulic analysis

For each network design, a steady state hydraulic solver is used to compute the heads at each node for the given water demands. The actual head for each node is compared with its corresponding minimum allowable head, thereby computing the head deficit (if any). The head deficits for every node are cumulated and this value $P_{i,G}$ is recorded for its corresponding network design to be used in the selection phase.

3.3 Objective function calculation

The integer strings are decoded into the corresponding pipe diameters and hence N network designs are produced. The total material and construction cost for each network design is computed as:

$$f = \sum_{j=1}^{ND} L_j C^j \quad (2)$$

where f is the objective function value for the individual i at the generation G , L_j represents the length of the pipe j and C^j is the cost per unit for the pipe diameter of pipe j in the individual i .

3.4 Selection

Constraint tournament selection (Deb 2000, Prasad and Park 2004, Tolson et al. 2009) is used to determine the individuals that survive to the next generation (a noted advantage of this method is that it does not require a penalty multiplier parameter). For two candidate solutions $X_{A,G}$ and $X_{B,G}$, the selection algorithm is given as:

$$X'_{G+1} = \begin{cases} \operatorname{argmin}_{X \in (X_{A,G}, X_{B,G})} f(X), & \text{if } X_{A,G} \text{ and } X_{B,G} \text{ are both feasible solutions.} \\ \operatorname{argmin}_{X \in (X_{A,G}, X_{B,G})} P(X), & \text{otherwise.} \end{cases} \quad (3)$$

where X'_{G+1} is the individual at generation $G+1$ which is either $X_{A,G}$ or $X_{B,G}$, $f(X)$ is the objective function value for string X , and $P(X)$ is the cumulative head deficit for string X . If a vector X is a feasible solution, $P(X)=0$. As can be seen from Equation (3), the solution with a smaller value of objective function is selected between two feasible solutions. A feasible solution is selected ($P(X)=0$) when compared with an infeasible solution ($P(X)>0$); The solution with less head constraint violation is chosen between two infeasible solutions.

3.5 Dither creeping mutation

Creeping mutation

Davis and Coombs (1987) first introduced a creeping mutation operator into their GAs for designing communication networks. A creeping mutation operator, in addition to bitwise mutation, was subsequently employed by Dandy et al. (1996) for WDS optimization. The basic idea of creeping mutation for WDS optimization is as follows: the creeping mutation operator mutates a selected substring to an adjacent pipe size, where conditional probabilities of downward (P_d) and upward (P_u) movement are employed. For example, $P_d=0.6$ means that there is a 60% probability of the creeping mutation operator will change the selected pipe size to the next adjacent smaller size, thus implying a 40% probability of creeping mutation moving the current pipe size to the

next larger size (as $P_u=1-P_d$). This creeping mutation approach (Dandy et al. 1996) is termed the traditional creeping mutation in this paper.

Dither mutation

The dither mutation strategy was used in differential evolution algorithm by Das et al. (2005). In their work, rather than providing a fixed value, a randomly generated value of the mutation weighting factor (F) was used in the mutation operator (the mutation weighting factor (F) used in differential evolution algorithm plays the same role as the mutation probability used in the GA). The dither differential evolution algorithm was applied to a number of mathematic optimization test functions (Das et al. 2005). The results of their study showed that the dither mutation strategy improved the convergence properties of differential evolution algorithm.

Dither creeping mutation

The dither creeping mutation is proposed in this paper to combine the creeping mutation and the dither mutation strategy. Within the proposed dither creeping mutation mechanism, each string, $i=1, \dots, N$, is first assigned a probability (P_{dcm}^i), where $P_{dcm}^i \in [P_{dcm}^{\min}, P_{dcm}^{\max}]$ is a uniform random variable. Each bit of each string i is selected with a probability of P_{dcm}^i to be mutated. Then the selected bit has a probability P_d of being mutated to the adjacent bit value below and a probability $1-P_d$ of being mutated to the adjacent bit value above. For a bit that is already set to the smallest (largest) value, upward (or downward) mutation is allowed only. The dither creeping mutation algorithm used in the proposed CMBGA is given in Figure 2. For K pipe diameter choices, the integer numbers from 0 to $K-1$ are associated with each pipe diameter, order from the smallest to the largest.

The proposed dither creeping mutation used in this paper is novel in that the mutation probability used for each string is uniformly randomly generated rather than being set to a fixed value. Thus different strings in the proposed CMBGA will be subject to different creeping mutation probabilities at the same generation and the same string will be also

subject to varied mutation probabilities at different stages. This differs significantly to the creeping mutation GA used by Dandy et al. (1996), for which, a fixed mutation probability was used throughout all the optimization and all the strings were subject to an identical mutation probability.

```

FOR  $i=1, 2, \dots, N$ 
     $P_{dcm}^i = P_{dcm}^{\min} + rand_i \times (P_{dcm}^{\max} - P_{dcm}^{\min})$ 
    FOR  $j=1, 2, \dots, ND$ 
        IF  $Rand_j < P_{dcm}^i$ 
            IF  $Rand_p \leq P_d$ 
                 $x_{i,G}^j = \max[0, x_{i,G}^j - 1]$ 
            ELSE
                 $x_{i,G}^j = \min[K - 1, x_{i,G}^j + 1]$ 
            END IF
        END FOR
    END FOR
END FOR

```

Figure 2 Dither creeping mutation algorithm

In Figure 2, P_{dcm}^i is the dither creeping mutation probability; P_{dcm}^{\max} and P_{dcm}^{\min} are the maximum and minimum allowable dither creeping mutation probabilities; $x_{i,G}^j$ is the i^{th} bit of the string in the proposed GA, $i=1, \dots, N$; $rand_i$, $Rand_j$ and $Rand_p$ are uniformly distributed random variables between 0 and 1; and P_d is conditional probability of downward mutation.

It is noted that the ES (Rechenberg 1965) and the proposed creeping mutation-based GA (CMBGA) proposed here have the same feature in that both of them do not utilize crossover operator. However, there exist some important differences between these two optimization algorithms. ES (such as $(\mu+\lambda)$ ES) normally selects the best μ individuals from the total $(\mu+\lambda)$ individuals to become parents for the next generation (Rechenberg 1965), where μ is the population size (parents) and λ is the number of offspring produced by the μ parents. In contrast all N individuals of the next generation are selected from the

N parents utilizing constraint tournament selection strategy (Deb 2001) for the proposed CMBGA. Real values are typically used to represent the strings in ES, while integer coding is utilized in the proposed CMBGA. A self-adaptive mutation strategy is normally used for ES (such as 1/5 success rule proposed by Rechenberg (1965)), while a dither creeping mutation strategy is adopted in the proposed CMBGA.

4. Case studies

Four case studies from the literature are used to investigate the effectiveness of the proposed CMBGA. These include the New York Tunnels Problem (NYTP) (Dandy et al. 1996), the Go Yang water network (GYN) (Kim et al. 1994), the Hanoi Problem (HP) (Fujiwara and Khang 1990) and the Balerma network (BN) (Reca and Martínez 2006). The CMBGA has been coded in C++ and combined with the EPANET2 hydraulic network solver.

In this study, the dither creeping mutation rate takes values from a small range that is centered about $1/ND$, where ND is number of decision variables for the WDS that is being optimized. This is motivated by the fact that a mutation probability of $1/ND$ has been demonstrated to be an effective value and is normally used for the GA (Goldberg 1989). A small interval of size $O(1/ND)$ is used to form the lower and upper bounds of the range. For example, for a WDS optimization problem with the $1/ND \approx 0.05$, the range of $P_{dcm} \in [0.03, 0.07]$ is used for the proposed CMBGA. The number of decision variables, the range for the dither creeping mutation probability (P_{dcm}), the population size and the maximum number of allowable evaluations for each case study are given in Table 1.

Table 1 Summary of case study characteristics and parameters of the CMBGA

WDS case study	Number of decision variables (ND)	$1/ND$	Range for $P_{dcm}^{(1)}$	Population size (N)	Maximum number of allowable evaluations
NYTP	21	≈ 0.05	[0.03, 0.07]	100	50,000
GYN	30	≈ 0.03	[0.01, 0.05]	100	20,000
HP	34	≈ 0.03	[0.01, 0.05]	100	100,000
BN	454	≈ 0.002	[0.001, 0.004]	500	10,000,000

⁽¹⁾Dither creeping mutation probability.

5. Preliminary sensitivity analysis

5.1 Sensitivity analysis of the probability of downward mutation

A sensitivity analysis of the conditional probability of downward mutation (P_d) is conducted. Values of $P_d=0.4, 0.5$ and 0.6 were used for the CMBGA applied to each case study, with all other parameter values being held constant. In order to present a reliable comparison, 1000 trial runs with different random number seeds were performed for the NYTP, GYN and HP case studies, and 100 runs for the BN case study. The properties of the best solution found versus the evaluation number for the CMBGA with different P_d values applied to the NYTP case study is shown in Figure 3.

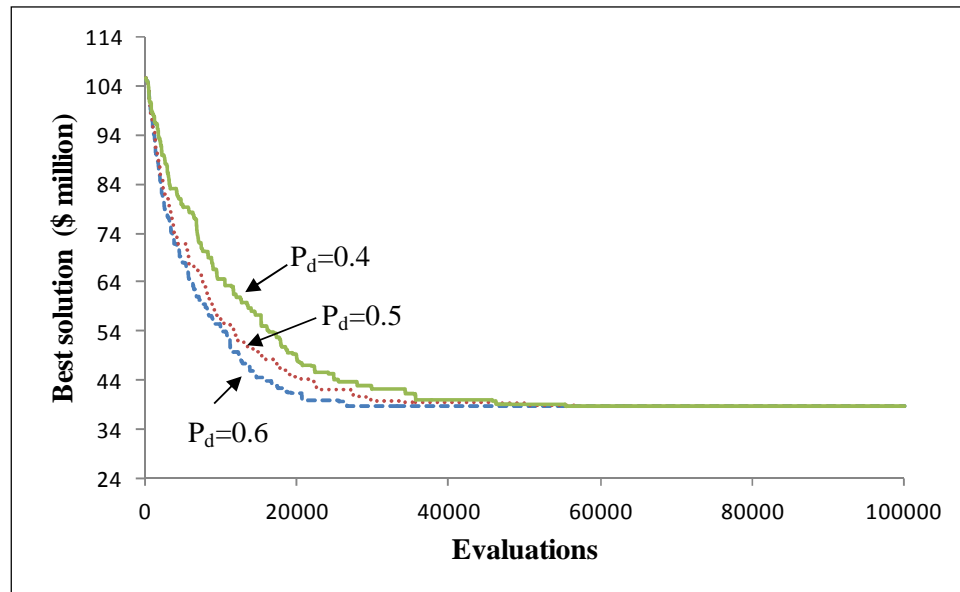


Figure 3 Example trial results for the best solution found versus evaluation number for the CMBGA with different P_d applied to the NYTP case study

From Figure 3, it is seen that the CMBGA with $P_d=0.6$ converged fastest, whereas for a value of $P_d=0.4$, it tended to converge the slowest. The CMBGA with $P_d=0.6$ and $P_d=0.5$ found the same final best solution with a cost of \$38.64 million, the current best known solution for the NYTP (Maier et al. 2003). The CMBGA with $P_d=0.4$ found a final best solution with a cost of \$38.80 million. The results for the trial runs are provided in Table 2.

It is observed from Table 2 that the CMBGA exhibited a similar performance for each value of P_d on each case study in terms of solution quality. The current best known solutions for the GYN and HP case studies, with costs of \$1.770 million (Tolson et al. 2009) and \$6.081 million (Reca and Martínez 2006) respectively, were found by the proposed CMBGA with different P_d values. In comparing efficiency (average number of evaluations to find the optimum), the $P_d=0.5$ runs performed the most efficiently for the HP case study. For the NYTP, GYN and BN case studies, the CMBGA with $P_d=0.6$ outperformed that for $P_d=0.4$ and 0.5. This can be explained by the fact that the final best solutions for the NYTP, GYN and BN case studies contain many minimum pipe diameters, and hence a conditional probability of downward mutation of $P_d=0.6$ is able to speed up the convergence.

Table 2 Results of CMBGA with different probabilities of downward mutation

WDS case study	Conditional probabilities of downward mutation (P_d)	Best solution found (M)	Number of best solutions found in R runs	Average cost of solutions (M)	Average evaluations to find best solutions for R runs
NYTP ($N=100$ $R=1000$)	0.4	\$38.64	565	\$39.61	52,345
	0.5	\$38.64	623	\$38.82	42,385
	0.6	\$38.64	673	\$38.81	26,512
GYN ($N=100$ $R=1000$)	0.4	\$1.770	845	\$1.774	14,357
	0.5	\$1.770	1000	\$1.770	12,453
	0.6	\$1.770	867	\$1.772	9,847
HP ($N=100$ $R=1000$)	0.4	\$6.081	640	\$6.198	94,371
	0.5	\$6.081	820	\$6.112	70,423
	0.6	\$6.081	814	\$6.117	89,056
BN ($N=500$ $R=100$)	0.4	€ 1.971	1	€ 2.065	8.7×10^6
	0.5	€ 1.963	1	€ 2.060	7.6×10^6
	0.6	€ 1.969	1	€ 2.001	6.3×10^6

R =number of runs using different random number seeds. N =population size.

It is concluded from this sensitivity study that P_d is a relatively robust parameter as a slight change of this parameter does not significantly affect the search performance of the CMBGA. As a result, in this paper, a probability of downward mutation (P_d) of 0.5 is adopted.

6. CMBGA performance discussion and comparison

The performance of the CMBGA for each case study is now discussed, and compared with other GA variants considered in this paper and also previously published results.

6.1 Performance of CMBGA compared with other GA variants

A total of four other GA variants with fine-tuned calibrated parameters have been studied in this paper in order to enable the comparison with the proposed CMBGA. These include a non-crossover traditional creeping mutation based GA (GA1), a crossover-based GA with bitwise mutation (GA2), a non-crossover bitwise-mutation based GA (GA3) and a crossover-based GA with creeping mutation (GA4). GA2 and GA4 are two normally used standard GA variants, whereas GA1 and GA3 were included in this work in order to investigate the performance of the GA variants for which only the traditional creeping mutation and only bitwise mutation were employed (compared to the proposed CMBGA where the dither creeping mutation operator is used).

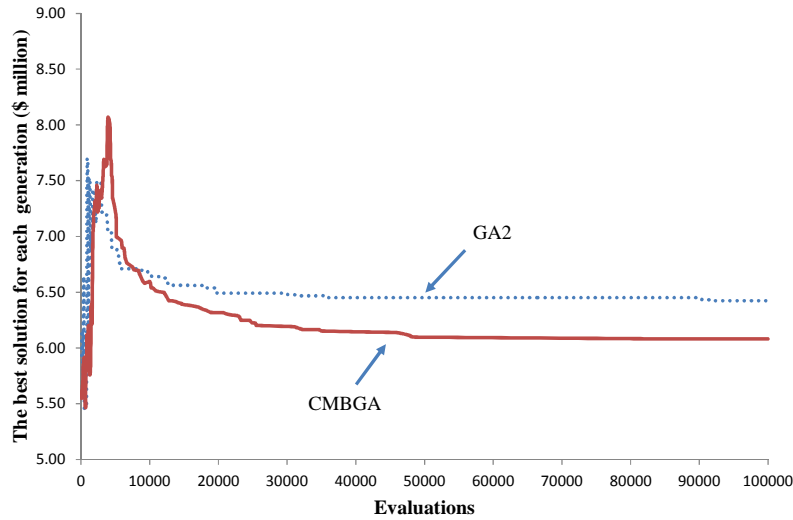
For each case study, each GA variant used the same population size and the same maximum allowable number of evaluations (outlined in Table 1). Integer coding, constraint tournament selection (tournament size=2) and an elite count of 2 were used for all the GA variants. The elite count is the number of individuals with the best fitness values in the current generation that are guaranteed to survive to the next generation (Gibbs et al. 2008). The other parameter values for the four GA variants applied to each case study are given in Table 3. These parameter values have been fine-tuned by a calibration process for each case study to give the best performance. A typical run for both CMBGA and GA2 (normally used GA) for the HP case study is presented in Figure 4.

As can be seen from Figure 4, the proposed CMBGA tends to converge faster than the GA2 with tuned parameter values. In addition, the final solution found by the CMBGA for the HP case study is significantly lower than that generated by the GA2. Similar results were obtained for other case studies and hence are not given.

Table 3 Fine-tuned calibrated parameter values for the four GA variants applied to each case study

Case study	GA1	GA2	GA3	GA4
NYTP	$P_{cm}=0.05$, Non-crossover	$P_c=0.5$, $P_m=0.03$, two-point crossover	$P_m=0.03$, Non-crossover	$P_c=0.9$, $P_{cm}=0.04$, two-point crossover
GYN	$P_{cm}=0.02$, Non-crossover	$P_c=0.8$, $P_m=0.02$, two-point crossover	$P_m=0.03$, Non-crossover	$P_c=0.8$, $P_{cm}=0.03$, two-point crossover
HP	$P_{cm}=0.02$, Non-crossover	$P_c=0.6$, $P_m=0.02$, two-point crossover	$P_m=0.01$ Non-crossover	$P_c=0.6$, $P_{cm}=0.02$, two-point crossover
BN	$P_{cm}=0.003$, Non-crossover	$P_c=0.6$, $P_m=0.002$, two-point crossover	$P_m=0.002$ Non-crossover	$P_c=0.7$, $P_{cm}=0.002$, two-point crossover

P_c : crossover probability. P_m : bitwise mutation probability. P_{cm} : traditional creeping mutation probability.

**Figure 4 Example trial results for the best solution found versus evaluation number for the CMBGA and GA2 applied to the HP case study**

For each GA variant, a total of 1000 trial runs with different starting random number seeds were performed for the NYTP, GYN and HP case studies, and 100 runs were used for the BN case study. The results of CMBGA and the four other GA variants with the calibrated parameter values are given in Table 4.

From the results, it is clearly seen that the proposed CMBGA consistently outperformed all the other GA variants in terms of solution quality and efficiency. In particular, the

proposed CMBGA found the best known solution for the HP case study with 82% based on 1000 different runs, while the GA1 and GA4 located the best known solution with only 39% and 2% respectively, and GA2 and GA3 were unable to find this best solution. For the GYN case study, the proposed CMBGA found the current best known solution with a 100% success rate based on 1000 different runs, which is higher than all the other GA variants presented in this paper. For the large BN case study, the proposed CMBGA was able to find better solutions than the other GA variants as shown in Table 4, while requiring fewer evaluations.

Table 4 The performance comparison of four GA variants against the proposed CMBGA

Case studies ¹	Algorithm	Best solution found (M)	Percent of best solutions found in R runs	Average cost (M)	No. of average evaluations for R runs
NYTP ($R=1000$, $N=100$)	CMBGA ^a	\$38.64	62%	\$38.82	42,385
	GA1 ^b	\$38.64	61%	\$38.82	46,850
	GA4 ^c	\$38.64	50%	\$39.04	44,324
	GA2 ^d	\$38.64	45%	\$39.16	49,950
	GA3 ^e	\$38.64	7%	\$40.07	57,469
GYN ($R=1000$, $N=100$)	CMBGA ^a	\$1.770	100%	\$1.770	12,453
	GA1 ^b	\$1.770	94%	\$1.770	15,661
	GA4 ^c	\$1.770	80%	\$1.775	16,987
	GA2 ^d	\$1.770	42%	\$1.825	19,387
	GA3 ^e	\$1.770	15%	\$1.997	19,657
HP ($R=1000$, $N=100$)	CMBGA ^a	\$6.081	82%	\$6.112	70,423
	GA1 ^b	\$6.081	39%	\$6.136	68,492
	GA4 ^c	\$6.081	2%	\$6.264	70,164
	GA2 ^d	\$6.099	0%	\$6.329	68,568
	GA3 ^e	\$6.113	0%	\$6.259	73,695
BN ($R=100$, $N=500$)	CMBGA ^a	€ 1.963	1%	€ 2.060	7.6×10^6
	GA1 ^b	€ 1.967	0%	€ 2.060	8.4×10^6
	GA4 ^c	€ 2.057	0%	€ 2.111	8.7×10^6
	GA2 ^d	€ 2.069	0%	€ 2.113	8.6×10^6
	GA3 ^e	€ 2.105	0%	€ 2.234	8.9×10^6

¹Ranked based on Column 4 for the NYTP, GYN and HP case studies, while Column 3 for the BN case study. R =number of runs using different random number seeds. N =population size.

^aNon-crossover dither creeping mutation based GA (CMBGA); ^bNon-crossover and traditional creeping mutation based GA; ^cCrossover and traditional creeping mutation GA; ^dCrossover and bitwise mutation GA. ^eNon-crossover and bitwise mutation GA

It was noted that the only difference between the CMBGA and the GA1 was that the dither creeping mutation was used for the CMBGA, while the traditional creeping mutation approach was implemented for GA1. This shows that the dither creeping mutation strategy is more effective than the traditional creeping mutation method as the CMBGA consistently yielded a better performance than GA1 with the best calibrated parameter value for each case study in terms of solution quality and efficiency.

The GA1 was found to perform the second best (only worse than the proposed CMBGA) for each case study. The only difference between GA1 and GA4 was that the crossover was not used for GA1, while the crossover was utilized for GA4. This shows that, for the four WDS case studies considered, the non-crossover GA with the traditional creeping mutation is more effective than the crossover based GA with traditional creeping mutation.

It is observed that the crossover-based GA with creeping mutation (GA4) consistently performed better than the crossover-based GA with bitwise mutation (GA2) for all the case studies. This suggests that creeping mutation is more effective than the bitwise mutation. This is also evidenced by the fact of that the bitwise mutation-based GA (GA3) was found to perform the worst as can be seen from Table 4. Both the proposed CMBGA and GA3 utilized only the mutation operator (both are non-crossover based GA variants). However, the proposed CMBGA significantly outperformed GA3 for each case study. This can be attributed to the fact that the dither creeping mutation was used in the proposed CMBGA, while only simple bitwise mutation was used in GA3.

In concluding the comparison, a final note on parameter calibration effort for each algorithm is required. For the proposed CMBGA, the dither creeping mutation rate (P_{dcm}) is set to be a range around the inverse value of the number of decision variables for the optimization problem, and hence no calibration is required. Although $P_d=0.6$ was found to be able to speed up the convergence of the CMBGA applied to some case studies, the CMBGA with $P_d=0.5$ showed a consistently reasonably good performance for each case study. Therefore, $P_d=0.5$ is recommended for application of the proposed CMBGA. For

the traditional GA or the GA with traditional creeping mutation, the crossover rate and mutation rate (bitwise or creeping mutation rate) require tuning. Thus, the proposed CMBGA removes the need of the users to utilize a trial-and-error process to determine the effective parameters of the GA in order to tackle different WDS optimization problems.

6.2 Performance of CMBGA compared with previously published results

The performance of the proposed CMBGA for each case study is now compared with other previously reported optimization techniques in terms of search ability and efficiency. The results obtained by some evolutionary optimization algorithms mentioned in this section are based on the algorithm parameters that have been tuned by the authors based on an extensive calibration process. These include the Improved GA (Dandy et al. 1996), Messy GA (Wu and Simpson 2001), GENOME (Reca and Martínez 2006), harmony search (HS) (Geem 2006b), Max-Min Ant System (MMAS) (Zecchin et al. 2007), particle-swarm harmony search (PSHS) (Geem 2009), genetic heritage evolution by stochastic transmission (GHEST) (Bolognesi et al. 2010) and differential evolution (DE) (Suribabu 2010). In contrast, the results from the hybrid discrete dynamically dimensioned (HD-DDS) method (Tolson et al. 2009) were based on a default parameter value and hence no parameter tuning was undertaken. From the particle swarm optimization (PSO) variant algorithm (Montalvo et al. 2008), however, it is unclear that whether the parameters were tuned or not. For each case study, the results of GA1, GA2, GA3 and GA4 with calibrated parameter values are also presented to enable a comparison with other published results.

New York Tunnels Problem (NYTP)

A comparison of algorithmic performance for the NYTP case study is given in Table 5. In terms of percent of trials with different random number seeds of finding the best solution, the HD-DDS (Tolson et al. 2009) and DE (Suribabu 2010) performed better than the proposed CMBGA, however, the average cost over all runs was only negligibly different. In terms of search efficiency, as measured by the average number of

evaluations, the proposed CMBGA performed worse than the DE, HD-DDS and MMAS, while better than other EAs given in Table 5.

Table 5 The performance comparison of the proposed CMBGA with previously published results for the NYTP case study

Algorithm ¹	No. of runs (R)	Best solution (\$M)	Percent of best solutions found in R runs	Average cost (\$M)	Maximum allowable evaluations	Average evaluations to find best solutions for R runs
HD-DDS ²	50	38.64	86%	38.64	50,000	13,000
DE ³	300	38.64	71%	NA	10,000	5,494
CMBGA	1000	38.64	62%	38.82	100,000	42,385
GA1	1000	38.64	61%	38.82	100,000	46,850
MMAS ⁴	20	38.64	60%	38.84	50,000	30,700
GA4	1000	38.64	50%	39.04	100,000	44,324
GA2	1000	38.64	45%	39.16	100,000	49,950
PSO variant ⁵	2000	38.64	30%	NA	80,000	NA
GA3	1000	38.64	7%	40.07	100,000	57,469
Messy GA ⁶	5	38.80	0	39.09	NA	48,427
Improved GA ⁷	5	38.80	0	38.98	200,000	143,790

¹Ranked based on Column 4. ²Tolson et al. (2009). ³Suribabu (2010). ⁴Zecchin et al. (2007). ⁵Montalvo et al. (2008). ⁶Wu and Simpson (2001). ⁷Dandy et al. (1996). NA means “not available”.

Go Yang Network (GYN)

Table 6 shows the performance comparison of different optimization techniques applied to the GYN case study. The current best known solution (\$1.770 million) was first located by HD-DDS and proposed CMBGA found this best solution 100% compared with 32% found by the HD-DDS. It is noted that the computational budget used by HD-DDS was 10,000, while 100,000 was used for the proposed CMBGA method.

Table 6 The performance comparison of the proposed CMBGA with previously published results for the GYN case study

Algorithm ¹	No. of runs (R)	Best solution (\$M)	Percent of trials with best solution found in R runs	Average cost (\$M)	Maximum allowable evaluations	Average evaluations to find best solutions for R runs
CMBGA	1000	1.770	100%	1.770	100,000	12,453
HD-DDS ²	50	1.770	32%	1.775	10,000	NA
NLP ³	NA	1.791	0	NA	NA	NA

¹Ranked based on Column 4. ²Tolson et al. (2009). ³Kim et al. (1994). NA means “not available”.

Hanoi Problem (HP)

The HP network has been investigated in many studies and is of particular interest as the region of the search space that contains feasible solutions is extremely small. The performance of different optimization techniques previously applied to the HP, including the results of the proposed CMBGA, are given in Table 7.

In comparing the algorithmic performance, it can be seen that the CMBGA achieved the highest percentage of best solutions found with a value of 82%, which is significantly higher than the other previously published algorithms. As the computational budget used by the DE (Suribabu 2010) was 10,000 while 100,000 used by the proposed CMBGA, we cannot conclude the proposed CMBGA performs better than the DE algorithm in terms of percentage of the best solutions found. The proposed CMBGA produced the lowest average solution with a value of \$6.112 million, which deviates only 0.51% from the best known solution. This illustrates that the CMBGA has a robust search strategy that is relatively effective in exploring the search space for highly constrained problems.

Table 7 The performance comparison of the proposed CMBGA with previously published results for the HP case study

Algorithm ¹	No. of runs (R)	Best solution (\$M)	Percent of best solutions found in R runs	Average cost (\$M)	Maximum allowable evaluations	Average evaluations to find best solutions for R runs
CMBGA	1000	6.081	82%	6.112	100,000	70,423
GA1	1000	6.081	39%	6.136	100,000	68,492
DE ²	300	6.081	21%	NA	10,000	6,244
GENOME ³	10	6.081	10%	6.248	100,000	NA
HD-DDS ⁴	50	6.081	8%	6.252	100,000	$\leq 100,000$
PSO variant ⁵	2000	6.081	5%	6.310	80,000	NA
GA4	1000	6.081	2%	6.264	100,000	70,164
HS ⁶	18	6.081	NA	6.139	NA	NA
PSHS ⁷	81	6.081	NA	6.340	NA	NA
GA2	1000	6.099	0%	6.329	100,000	68,568
GA3	1000	6.113	0%	6.259	100,000	73,659
MMAS ⁸	20	6.134	0%	6.386	100,000	85,600

¹Ranked based on Column 4. ²Suribabu (2010). ³Reca and Martinez (2006). ⁴Tolson et al. (2009). ⁵Montalvo et al. (2008). ⁶Geem (2006b). ⁷Geem (2009). ⁸Zecchin et al. (2007). NA means “not available”.

Balerma network (BN)

Table 8 give performance comparisons among different optimization techniques applied to the BN case study. As shown in Table 8, the current best known solution for the BN case study was found by Zheng et al. (2011b) using a coupled NLP and DE algorithm with a cost of € 1.923 million using 1,427,850 evaluations. Tolson et al. (2009) proposed the HD-DDS to find a best solution of € 1.940 million using 30 million evaluations budget. The CMBGA located a best solution with value of € 1.963 million for the BN case study, which is lower than the GHEST (Bolognesi et al. 2010), HS (Geem 2009) and GENOME (Reca and Martinez 2006). The average cost solution produced by the proposed CMBGA was lower than that obtained by GENOME.

In comparing the quality of the best solution for the BN case study, the proposed CMBGA was not as efficient as the NLP-DE (Zheng et al. 2011b) or HD-DDS (Tolson et al. 2009). However, both the NLP-DE and HD-DDS are hybrid optimization techniques. For the HD-DDS method, an EA was combined with two local search techniques (one pipe and two pipes search methods), while for the NLP-DE approach, the DE was combined with a NLP.

Table 8 The performance comparison of the proposed CMBGA with previously published results for the BN case study

Algorithm ¹	No. of runs (R)	Best solution (€ M)	Percent of best solutions found in R runs	Average cost (€ M)	Maximum allowable evaluations	Average evaluations to find best solutions for R runs
NLP-DE ²	10	1.923	10%	1.927	10^7	1.4×10^6
HD-DDS ³	1	1.940	10%	2.014	3×10^7	NA
CMBGA	100	1.963	1%	2.060	10^7	7.6×10^6
GA1	100	1.967	1%	2.060	10^7	8.4×10^6
GHEST ⁴	10	2.002	10%	2.055	NA	2.54×10^5
HS ⁵	NA	2.018	NA	NA	10^7	NA
GA4	100	2.057	1%	2.111	10^7	8.7×10^6
GA2	100	2.069	1%	2.113	10^7	8.6×10^6
GA3	100	2.105	1%	2.234	10^7	8.9×10^6
GENOME ⁶	10	2.302	NA	2.334	10^7	NA

¹Ranked based on Column 3. ²Zheng et al. (2011b). ³Tolson et al (2009). ⁴Bolognesi et al. (2010). ⁵Geem (2009). ⁶Reca and Martinez (2006). NA means “not available”.

7. Conclusions

Within this paper, a dither creeping mutation based GA with no crossover (CMBGA) has been proposed. It differs significantly from the commonly used GA approach as no crossover operator is used. A big advantage of CMBGA is its simplicity and that it requires the tuning of fewer parameters compared with the traditional GA. It should be noted that the proposed CMBGA is a variant of the GA with constraint tournament selection (but with the crossover probability set to be zero) and its effectiveness has been demonstrated for pipe network optimization in this paper.

The proposed CMBGA has been compared with the other four GA variants based on four case studies. The results obtained show that the proposed CMBGA exhibits improvements in efficiently finding optimal solutions for the four case studies compared with the other GA variants studied in this paper. In addition, it has been concluded from this study that the dither creeping mutation approach is more effective than the traditional creeping method, which in turn, is better than the bitwise mutation method.

The proposed CMBGA has also been compared with other EAs that have been previously applied to the four case studies. The proposed CMBGA shows a comparable performance to the other EAs, but it is not as efficient as the DE (Suribabu 2010) and HD-DDS (Tolson et al. 2009) for the NYTP case study. For the GYN case study, the proposed CMBGA was able to find the current best known solution with a success rate of 100% based on 1000 different runs, which is significantly higher than other algorithms (the other GA variants and all previously published results for this problem). For the HP case study, the CMBGA significantly outperformed the other EAs as it found the current best known solution for this case study with the highest success rate (82%). For the large case study (BN case study), the proposed CMBGA was able to find satisfactory results.

References

- Alperovits, E. and Shamir, U., 1977. Design of optimal water distribution systems. *Water Resour. Res.*, 13 (6), 885-900.

- Bolognesi, A., Bragalli, C., Marchi, A. and Artina, S., 2010. Genetic heritage evolution by stochastic transmission in the optimal design of water distribution networks. *Advances in Engineering Software*, 41 (5), 792-801.
- Cunha, M.D.C. and Sousa, J., 2001. Hydraulic infrastructures design using simulated annealing. *Journal of Infrastructure Systems*, 7 (1), 32-39.
- Dai, X., Zou, R., Sun, R., Feng, R. and Huihe, S., 2002. Convergence properties of non-crossover genetic algorithms. *Intelligent Control and Automation, Proceedings of the 4th World Congress.*, Shanghai, P. R. China, 3, 1822-1826.
- Dandy, G.C., Simpson, A.R. and Murphy, L.J., 1996. An improved genetic algorithm for pipe network optimization. *Water Resour. Res.*, 32 (2), 449-458.
- Das, S., Konar, K. and Chakraborty, U.K., 2005. Two improved differential evolution schemes for faster global search. In *Proceedings of Conference on Genetic and Evolutionary Computation (GECCO)*, 991-998.
- Davis, L. and Coombs, S., 1987. Genetic algorithms and communication link speed design: Constraints and operators. *Proceedings of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum, Hillsdale, 257-260.
- Deb, K., 2001. *Multi-objective Optimization Using Evolutionary Algorithm*, Wiley, London.
- Deb, K., 2000. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186 (2-4), 311-338.
- Eusuff, M.M. and Lansey, K.E., 2003. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 129 (3), 210-225.
- Falco, I., Della Cioppa, A. and Tarantino, E., 2002. Mutation-based genetic algorithm: Performance evaluation. *Applied Soft Computing*, 1 (4), 285-299.
- Fogel, L.J., Owens, A.J. and Walsh, M.J., 1966. *Artificial Intelligence Through Simulated Evolution*, Wiley, New York.
- Fogel, D.B., and Atmar, J.W., 1990. Comparing genetic operators with Gaussian mutation in simulated evolutionary process using linear systems. *Biological Cybernetics*, 63, 111-114.
- Fogel, D.B., 2000. *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*. *IEEE Press*, New York, pp. 140.
- Fujiwara, O., Jenchaimahakoon, B. and Edirisinghe, N.C.P., 1987. A modified linear programming gradient method for optimal design of looped water distribution networks. *Water Resour. Res.*, 23 (6), 977-982.

- Fujiwara, O. and Khang, D.B., 1990. A two-phase decomposition method for optimal design of looped water distribution networks. *Water Resour. Res.*, 26 (4), 539-549.
- Gessler, J., 1985. Pipe network optimization by enumeration. *Proc., Computer Applications in Water Resources*, ASCE, New York, 572-581.
- Geem, Z.W., 2006(b). Improved harmony search from ensemble of music players search. *Lecture Notes in Artificial Intelligence*, 4251, 86-93.
- Geem, Z.W., Kim, J.H. and Loganathan, G.V., 2002. Harmony search optimization: application to pipe network design. *Int. J. Model. Simulat.*, 22(2), 125-133.
- Geem, Z.W., 2009. Particle-swarm harmony search for water network design. *Engineering Optimization*, 41 (4), 297-311.
- Gibbs, M.S., Dandy, G.C. and Maier, H.R., 2008. A genetic algorithm calibration method based on convergence due to genetic drift. *Inf. Sci.*, 178 (14), 2857-2869.
- Goldberg, D.E., 1989. *Genetic Algorithm in Search, Optimization and Machine Learning*. MA: Addison-Wesley.
- Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Mich.
- Lansey, K.E. and Mays, L.W., 1989. Optimization model for water distribution system design. *Journal of Hydraulic Engineering*, 115 (10), 1401-1418.
- Maier, H.R., Simpson, A.R., Zecchin, A.C., Foong, W.K., Phang, K.Y., Seah, H.Y. and Tan, C.L., 2003. Ant colony optimization for design of water distribution systems. *Journal of Water Resources Planning and Management*, 129 (3), 200-209.
- Montalvo, I., Izquierdo, J., Pérez, R. and Tung, M.M., 2008. A diversity-enriched variant of discrete PSO applied to the design of water distribution networks. *Eng. Optim.*, 40(7), 655-668.
- Morgan, D.R. and Goulter, I.C., 1985. Optimal urban water distribution design. *Water Resour. Res.*, 21 (5), 642-652.
- Murphy, L.J. and Simpson, A.R., 1992. Pipe optimization using genetic algorithms. *Research Report No. R93*, Department of Civil Engineering, University of Adelaide.
- Palmes, P.P., Hayasaka, T. and Usui, S., 2005. Mutation-based genetic neural network. *Neural Networks, IEEE Transactions on Neural Networks*, 16 (3), 587-600.

- Rechenberg, I., 1965. *Cybernetic solution path of an experimental problem*. Royal Aircraft Establishment, Library Translation Number 1122, Farnborough, England.
- Reca, J. and Martínez, J., 2006. Genetic algorithms for the design of looped irrigation water distribution networks. *Water Resour. Res.*, 42 (5), W05416.
- Reca, J., Martínez, J., Gil, C. and Baños, R., 2008. Application of several meta-heuristic techniques to the optimization of real looped water distribution networks. *Water Resources Management*, 22 (10), 1367-1379.
- Savic, D.A. and Walters, G.A., 1997. Genetic algorithms for least-cost design of water distribution networks. *Journal of Water Resources Planning and Management*, 123 (2), 67-77.
- Schaffer, J.D. and Eshelman, L.J., 1991. On crossover as an evolutionarily viable strategy. *Proceeding of the Fourth International Conference on Genetic Algorithms*, LaJolla, Morgan Kaufmann, 61-68.
- Simpson, A.R., Dandy, G.C. and Murphy, L.J., 1994. Genetic algorithms compared to other techniques for pipe optimization. *Journal of Water Resources Planning and Management*, 120 (4), 423-443.
- Suribabu, C.R., 2010. Differential evolution algorithm for optimal design of water distribution networks. *Journal of Hydroinformatics*, 12 (1), 66-82.
- Spears, W.M., 1993. Crossover or Mutation. *Proceedings of the Foundations of Genetic Algorithms*, Morgan Kaufmann, 2, 221-237.
- Syswerda, G., 1989. Uniform crossover in genetic algorithm. *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann, 2-9.
- Thibert-Plante, X., and Charbonneau, P., 2007. Crossover and evolutionary stability in the Prisoner's Dilemma. *Evolutionary Computation*, 15(3), 321-344.
- Tolson, B.A., Asadzadeh, M., Maier, H.R. and Zecchin, A., 2009. Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization. *Water Resour. Res.*, 45 (12), W12416.
- Vairavamoorthy, K. and Ali, M., 2000. Optimal design of water distribution systems using genetic algorithms. *Computer-Aided Civil and Infrastructure Engineering*, 15 (5), 374-382.
- Vairavamoorthy, K. and Ali, M., 2005. Pipe index vector: A method to improve genetic-algorithm-based pipe optimization. *Journal of Hydraulic Engineering*, 131 (12), 1117-1125.

- Vose, M.D., 1994. A closer look at mutation in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 10, 423-434.
- Wu, Z.Y. and Simpson, A.R., 2001. Competent genetic-evolutionary optimization of water distribution systems. *Journal of Computing in Civil Engineering*, 15 (2), 89-101.
- Zecchin, A.C., Maier, H.R., Simpson, A.R., Leonard, M. and Nixon, J.B., 2007. Ant colony optimization applied to water distribution system design: Comparative study of five algorithms. *Journal of Water Resources Planning and Management*, 133 (1), 87-92.
- Zheng, F., Simpson, A.R. and Zecchin, A.C., 2011(a). Dynamically expanding choice table approach to genetic algorithm optimization of water distribution systems *J. Water Resources Planning and Management*, in press.
- Zheng, F., Simpson, A.R. and Zecchin, A.C., 2011(b). A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems. *Water Resour. Res.*, 47 (8), W08531.

Chapter 5. Journal Paper 3-A Self-adaptive DE for WDS Optimization

5.1 Synopsis

A self-adaptive differential evolution algorithm applied to water distribution system optimization

Chapters 3 and 4 focus on improving the performance of genetic algorithms (GAs) in terms of water distribution system (WDS) optimization. Another type of evolutionary algorithm that has been investigated in this research is the differential evolution (DE) algorithm. DE is a relatively new optimization algorithm that has been recently introduced for dealing with the WDS optimal design problems. A review of DE was elaborated upon in Section 2.3.2 of Chapter 2.

DE has been viewed as a promising optimization technique due to its excellent performance when applied to WDS optimization problems. A performance comparison between the standard DE and the standard GA applied to two benchmark WDS case studies is presented in Figure 5.1 and 5.2 respectively. These two case studies are the New York Tunnels Problem (NYTP) and the Hanoi Problem (HP). The details of these two case studies are included in the paper of this Chapter (Chapter 5).

The control parameters for each algorithm have been tuned by a trial-and-error approach and the selected parameter values are presented in Table 5.1. The starting random number seeds for the DE and GA run were identical for each case study in order to enable a fair comparison.

Table 5.1 Parameter values for each algorithm applied to two case studies

Algorithms	NYTP	HP
DE	N=100, F=0.5, CR=0.6	N=100, F=0.7, CR=0.8
GA	N=100, P _c =0.6, P _m =0.03	N=100, P _c =0.5, P _m =0.02

N=population size; *F*=mutation weighting factor used for DE; *CR*=crossover rate used for DE; *P_c*=crossover probability used for the GA and *P_m*=bitwise mutation rate used for the GA.

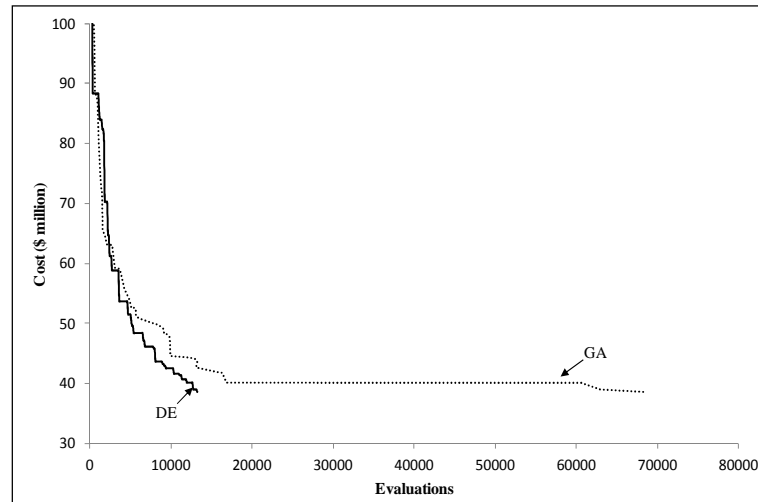


Figure 5.1 A convergence comparison between DE and GA applied to the NYTP case study

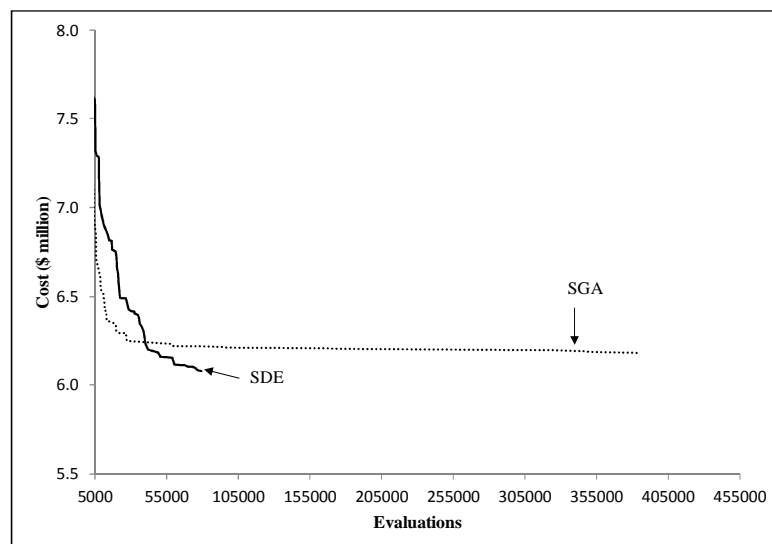


Figure 5.2 A convergence comparison between DE and GA applied to the HP case study

As clearly shown in Figures 5.1 and 5.2, DE significantly outperformed the GA for both case studies in terms of the convergence speed of the whole search process. It is observed from Figure 5.1 that DE and the GA have a similar convergence speed during early generations for the NYTP case study. For the HP case study, the GA exhibits a slightly faster convergence speed than the DE algorithm at the early stages. However, the GA tends to be stagnate at the later generations for both case studies as shown in Figures 5.1

and 5.2, while the DE algorithm still maintains a high convergence speed. This implies that DE is more robust than the GA at the later generations. All solutions for the SDE and SGA applied to the HP case study are presented in Figure 5.3.

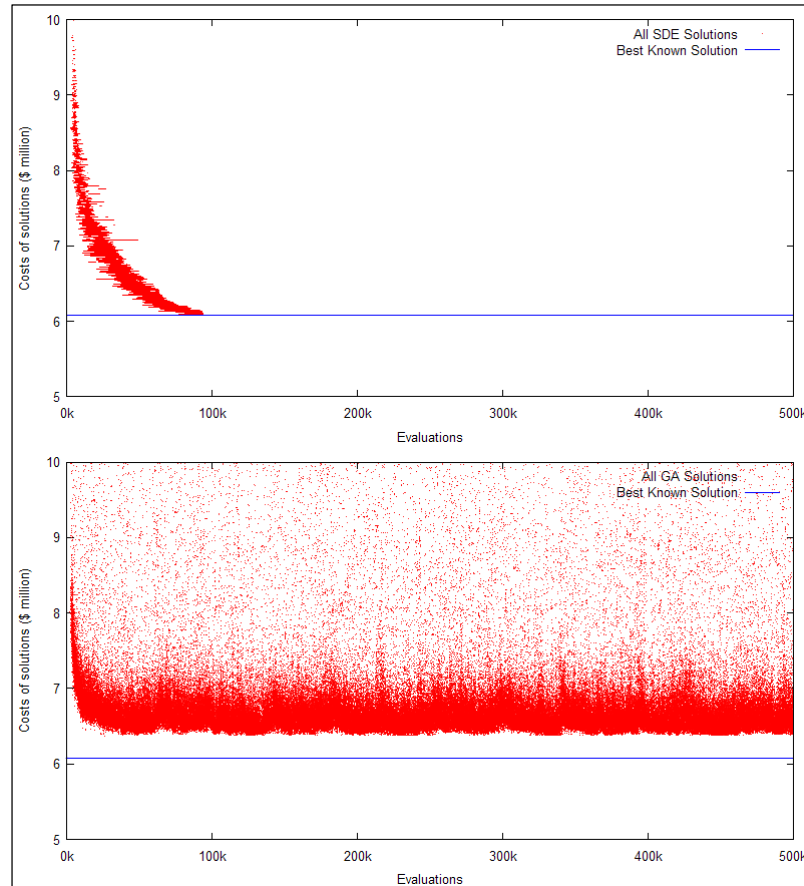


Figure 5.3 Solutions of the DE and SGA applied to the HP case study

As shown in Figure 5.3, the solutions generated by the GA are scattered while the solutions yielded by DE (denoted as SDE in Figure 5.3) tend to converge one final solution. This illustrates the significant difference between the GA and the DE algorithm. All the GA solutions are significantly higher in cost than the best known solution for the HP case study (\$6.081 million), while all DE solutions finally converged to the current best known solution.

It has been widely recognized that a GA is able to converge quickly at the early stages of an optimization process, while tending to be stagnate at the later stage of the whole

search process. This agrees with observations made in the current research (see Figures 5.1, 5.2 and 5.3). In contrast, DE exhibits consistently good convergence speed throughout the whole search process as shown in Figures 5.1, 5.2 and 5.3. Thus, DE can be viewed as a more suitable optimization technique for use when designing water networks. However, the performance of DE is extremely sensitive to the control parameters (especially the F and CR) used, which has been clearly stated in Section 2.3.2.5 of Chapter 2. These parameters need to be fine-tuned for different optimization problems as they are generally problem-dependent. This causes difficulties in implementing the DE algorithm to deal with real-world sized optimization problems, because tuning the parameter values is computationally expensive, especially when dealing with relatively large water networks.

In order to reduce the effort required to tune the parameter values of the DE algorithm, a self-adaptive DE algorithm is proposed in this research. Three new contributions in this thesis are included in the proposed SADE algorithm:

- Instead of pre-specification, the control parameters of F and CR are encoded into the chromosome of the SADE algorithm and hence are adapted by means of evolution.
- The F and CR values of the SADE algorithm apply at the individual level rather than at the generational level normally used by the traditional DE algorithm.
- A new convergence criterion is proposed for the SADE algorithm as the termination condition, thereby avoiding pre-specifying a fixed number of generations or computational budget to terminate the evolution. The only parameter value that needs to be provided for the proposed SADE, therefore, is the population size (N). The population size is a relatively easy parameter to adjust since a slight variation of its value does not appear to significantly impact the performance of the SADE. In addition, it was proven in the current research that a population size within $[1D, 6D]$ is an approximate heuristic for the proposed SADE applied to WDS case studies.

The proposed SADE provides a robust tool for the optimization of the WDS design (or rehabilitation of an existing WDS). This is because that (i) the proposed SADE algorithm does not require as much fine-tuning of parameter values nor pre-specification of a computational budget; and (2) the proposed SADE algorithm is able to find optimal solutions with good quality and great efficiency.

The proposed SADE algorithm differs from the self-adaptive DE algorithm (denoted as jDE) developed by Brest et al. (2006) in that: (i) the F and CR parameters that are able to generate better solutions are directly passed onto the next generation in the proposed SADE, in contrast they survive in the next generation with a probability of $1-\tau_1$ and $1-\tau_2$ ($0 < \tau_1, \tau_2 < 1$) respectively in the jDE; (ii) the jDE has two more parameters, than the proposed SADE, that need to be specified (τ_1 and τ_2); and (iii) the convergence criterion developed for the proposed SADE removes the need to pre-specify the computational budget, while a computational budget needs to be pre-set for the jDE (Brest et al. 2006).

It is necessary to define the traditional DE here in order to enable the comparison with the proposed SADE. The traditional DE is the DE algorithm uses the mutation rate (F) and crossover rate (CR) at the generational level, both of which need to be tuned. In addition, the computational budget for the traditional DE needs to be pre-specified. The previous DE applications including Vasani and Simonovic (2010), Suribabu (2010) and Dandy et al. (2012) are traditional DE algorithms.

This work has been published on *Journal of Computing in Civil Engineering* and the paper is provided here.

Citation of Paper

Zheng, F., A. C. Zecchin and A. R. Simpson (2012a). “A self adaptive differential evolution algorithm applied to water distribution system optimization.” *Journal of Computing in Civil Engineering*, doi:10.1061/(ASCE)CP.1943-5487.0000208.

STATEMENT OF AUTHORSHIP (PUBLISHED)

Journal paper title: **A self-adaptive differential evolution algorithm applied to water distribution system optimization.**

Authors: Feifei Zheng¹, Aaron Zecchin² and Angus Simpson³

Corresponding author:

¹Feifei Zheng (Candidate): PhD student, School of Civil, Environmental and Mining Engineering, University of Adelaide

Wrote the manuscript, performed all analysis, developed the model and theory and acted as corresponding author.

I hereby certify that the statement of contribution is accurate.

Signed _____ Date _____

²Aaron Zecchin: Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide

Reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

³Angus Simpson: Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide

Reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

Journal of Computing in Civil Engineering, doi:10.1061/(ASCE)CP.1943-5487.0000208.

5.2 Journal Paper 3: A self-adaptive differential evolution algorithm applied to water distribution system optimization (Published in the *Journal of Computing in Civil Engineering*)

Feifei Zheng, Aaron C. Zecchin and Angus R. Simpson

ABSTRACT

Differential evolution (DE) is a relatively new technique that has recently been used to optimize the design for water distribution systems (WDSs). Several parameters need to be determined in the use of DE, including: population size, N ; mutation weighting factor, F ; crossover rate, CR and a particular mutation strategy. It has been demonstrated that the search behavior of DE is especially sensitive to the F and CR values. These parameters need to be fine-tuned for different optimization problems as they are generally problem-dependent. A self-adaptive differential evolution (SADE) algorithm is proposed to optimize the design of WDSs. Three new contributions are included in the proposed SADE algorithm: (i) instead of pre-specification, the control parameters of F and CR are encoded into the chromosome of the SADE algorithm and hence are adapted by means of evolution; (ii) F and CR values of the SADE algorithm apply at the individual level rather than the generational level normally used by the traditional DE algorithm; and (iii) a new convergence criterion is proposed for the SADE algorithm as the termination condition, thereby avoiding pre-specifying a fixed number of generations or computational budget to terminate the evolution. Four WDS case studies have been used to demonstrate the effectiveness of the proposed SADE algorithm. The results obtained show that the proposed algorithm exhibits good performance in terms of solution quality and efficiency. The advantage of the proposed SADE algorithm is that it reduces the effort required to fine-tune algorithm parameter values.

CE Database subject headings: optimization; water distribution systems; differential evolution.

Author Keywords: optimization; differential evolution; water distribution systems.

INTRODUCTION

Water distribution systems (WDSs) are one of the most expensive public infrastructure works as they require a high level of capital investment for construction and a continuing investment for maintenance. Research into the optimal design of WDSs is motivated, therefore, by the possibility of substantial cost savings. The optimal design of a WDS involves identifying the lowest cost pipe network that is able to provide the required demand and head pressure for each individual supply node. The design of WDSs poses challenges for optimization tools for two main reasons: (i) the nonlinear relationships between pipe discharges and head losses introduce complex nonlinear constraints into the optimization problem, and (ii) the discrete pipe diameters lead to a combinatorial optimization problem.

Historically, a number of traditional optimization techniques have been applied to water network optimal design, such as linear programming (Alperovits and Shamir 1977; Quindry et al. 1981; Fujiwara et al. 1987) and non-linear programming (Lansey and Mays 1989; Fujiwara and Khang 1990). However, due to the multi-modal nature of the fitness landscape for the optimization of water distribution system problem, these methods are more likely to converge on local optimal solutions, where the final solutions are highly sensitive to the initial starting point (Eiger et al. 1994). In addition, the final solutions may include continuous pipe sizes or split pipes, which is a significant practical limitation.

Evolutionary algorithms (EAs) have been popular alternatives for optimizing WDS designs as they are able to handle a discrete search space directly, and are less likely to be trapped by local optimal solutions. The search strategy of EAs differs from the traditional optimization techniques, such as linear programming or non-linear programming, in that they explore broadly across the search space using a population-based stochastic evolution algorithm, where no gradient information is required.

Over the last two decades, a number of EAs have been employed to optimize the design of WDSs, such as genetic algorithms (Murphy and Simpson 1992; Simpson et al. 1994; Dandy et al. 1996; Savic and Walters 1997); simulated annealing (Cunha and Sousa 2001); harmony search (Geem et al. 2002); shuffled frog leaping algorithm (Eusuff and Lansey 2003); Ant Colony Optimization (Maier et al. 2003); particle swarm optimization (Suribabu and Neelakantan 2006); cross entropy (Perelman and Ostfeld, 2007); and scatter search (Lin et al. 2007). These techniques have been successfully applied to a number of WDS optimization problems and have been demonstrated to be more effective in finding optimal solutions compared to traditional optimization techniques. It has been noticed that the performance of all these EAs, in terms of robustness and efficiency, are significantly affected by the algorithm parameter settings, which need to be adjusted for different optimization problems. It has been reported by Tolson et al. (2009) that the number of parameters that need to be fine-tuned for different optimization problems for these EAs varies from 3 to 8. These do not include a termination criterion parameter that also needs to be pre-specified to end the EA run (i.e. normally the maximum number of allowable evaluations or generations). The appropriate parameters of EAs are varied for different optimization problems and normally are adjusted by trial and error. Thus, it is extremely computationally expensive to determine the proper parameter values for a newly given WDS case study.

Differential evolution (DE), proposed by Storn and Price (1995), has recently been used to optimize WDSs (Suribabu 2010; Dandy et al. 2010). There are three important operators involved in the application of the DE algorithm: a mutation operator, a crossover operator and a selection operator. These operators are similar to a genetic algorithm (GA), but DE algorithms differ significantly from a GA in the mutation process, in that the mutant solution is generated by adding the weighted difference between two random population members to a third member.

A total of four parameters need to be pre-determined in the use of DE, including: population size, N ; mutation weighting factor, F ; crossover rate, CR ; and a particular

mutation strategy. It has been demonstrated that the performance of DE is governed by these parameters (especially the F and CR) based on a number of numerical optimization case studies (Storn and Price 1995; Vesterstrom and Thomsen 2004). In terms of optimizing WDSs, Suribabu (2010) and Vasan and Simonovic (2010) concluded that the performance of DE algorithms was at least as good as, if not better, than other EAs such as GAs and Ant Colony Optimization. While Dandy et al. (2010) has stated that GAs give better results overall than DE algorithms in terms of solution quality and efficiency. The contradiction of results reported by Suribabu (2010) and Dandy et al. (2010) can be explained by the fact that the different parameter values including N , F and CR are used in these DE applications. In addition, Suribabu (2010) investigated the effectiveness of the DE using a number of different F and CR combinations (N is constant) applied to WDS optimization problems. His results show that the performance of the DE algorithm applied to the WDS optimization is highly dependent on the parameter values selected. As these control parameters are problem dependent, using the DE algorithm effectively is time consuming since appropriate parameter values have to be established for each new WDS case study.

Investigations have been undertaken to avoid pre-specifying parameter values in EAs. Bäck et al. (1991) introduced a self-adaptive algorithm to dynamically adjust the mutation probability in the evolution strategy. Eiben et al. (1999) gave a systematic analysis of a self-adaptation strategy for the parameters of EAs. Wu and Simpson (2002) and Wu and Walski (2005) proposed a self-adaptive penalty approach GA for pipeline optimization. The penalty multiplier was encoded onto each member of the population, thereby allowing the penalty multiplier to evolve over the course of the GA optimization. Thus, there is no need to pre-specify a penalty multiplier before running the GA run. Gibbs et al. (2010) provided an estimate of population size for GA applications based on the genetic drift. Tolson et al (2009) developed a hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for WDS optimization and proposed the HD-DDS as a parameter-setting-free algorithm. Geem and Sim (2010) proposed a parameter-setting-free harmony search algorithm to optimize the design of WDSs.

Brest et al. (2006) proposed a self-adaptive strategy to evolve the F and CR values of the DE algorithm, which is called jDE. In the jDE algorithm, the F and CR values were adjusted by introducing two new parameters τ_1 and τ_2 . They concluded that the self-adaptive DE algorithm performed better than the traditional DE algorithm in terms of convergence speed and final solution quality based on testing a number of numerical benchmark optimization problems.

In this paper, a new self-adaptive differential evolution (SADE) algorithm is proposed. A total of three novel aspects are involved in the proposed SADE algorithm, which are (i) control parameters of F and CR are encoded into the chromosome of the SADE algorithm rather than pre-specification and hence are adapted by means of evolution; (ii) F and CR values of the SADE algorithm apply at the individual level, which differs to the traditional DE algorithm that F and CR values applied at the generational level; and (iii) a new convergence criterion is proposed for the SADE algorithm as the termination condition in order to avoid pre-specifying a fixed number of generations or evaluations to terminate the evolution.

The F and CR are encoded onto the solution string and hence are subject to evolution in the proposed SADE algorithm. Each individual in the initial population is assigned with randomly generated F and CR values within a given range. The better values of F and CR that produce fitter offspring are directly passed onto the next generation. If the F and CR values are unable to yield better offspring, these two values are randomly regenerated within the given range for the next generation. This newly proposed SADE differs with the jDE algorithm (Brest et al. 2006). For the jDE algorithm used in Brest et al. (2006), the F and CR values survive to the next generation with a particular probability τ_1 and τ_2 ($0 < \tau_1, \tau_2 < 1$) respectively. With a probability of $1 - \tau_1$ and $1 - \tau_2$, the F and CR values are randomly re-initialized to new values within the given range for the next generation respectively. The τ_1 and τ_2 values need to be pre-specified and hence two new parameters were introduced in the jDE algorithm proposed by Brest et al. (2006). The self-adaptive strategy proposed in this paper allows the F and CR values that are able

to yield fitter offspring are more likely to survive longer over generations during the running of the algorithm, which in turn, generates further better offspring. The details of the proposed SADE algorithm are presented later in this paper.

The F and CR values in traditional DE algorithms (Storn and Price 1995) and the DE algorithms applied to the WDS optimization (Suribabu 2010; Dandy et al. 2010; Zheng et al. 2011) are typically applied at the generation level during optimization. This implies all the individuals are therefore subject to identical mutation weighting and crossover strength. As with Brest et al. (2006), the F and CR values in the proposed SADE algorithm are applied at the individual level and hence different individuals within a population may have different mutation weightings and crossover rates applied. This approach was motivated by the fact that different individuals in a generation will be at varying distances from the optimal solutions and therefore require different mutation and crossover strength. For the individuals at greater distances from the optimal solutions, a relatively large F and CR is probably appropriate, while in contrast, for the individuals at relatively short distances from the optimal solutions, a relatively smaller F and CR may be suitable. Thus, the search performance of the proposed SADE algorithm is expected to improve as different individuals are associated with different F and CR values by means of evolution.

For EAs, the convergence condition is usually a fixed number of generations reached (limit of computational budget) or a predefined small value reached between two consecutive generations in terms of objective function values (Deb 2001). In the case of WDS optimization problems, the maximum number of allowable evaluations or generations is normally used as the termination condition (Savic and Walters 1997; Tolson et al. 2009; Suribabu 2010; Dandy et al. 2010). However, the appropriate number of allowable evaluations or generations is optimization problem-dependent and hence generally determined by trial and error. Moreover, the evolution time to reach the same final solutions of EAs applied to the same optimization problem with different starting points is also different. This unavoidably results in computational waste when the budget

is greater than required or computational insufficiency when the budget is smaller than required. In addition to the self-adaptive strategy, a new convergence criterion is proposed in this paper for the SADE algorithm to eliminate the need to preset the computational budget and thereby avoid computational excess and insufficiency. The details of the proposed convergence criterion are given in the next section.

SELF-ADAPTIVE DIFFERENTIAL EVOLUTION

Figure 1 illustrates the flowchart of the proposed SADE algorithm to be discussed in the following sections.

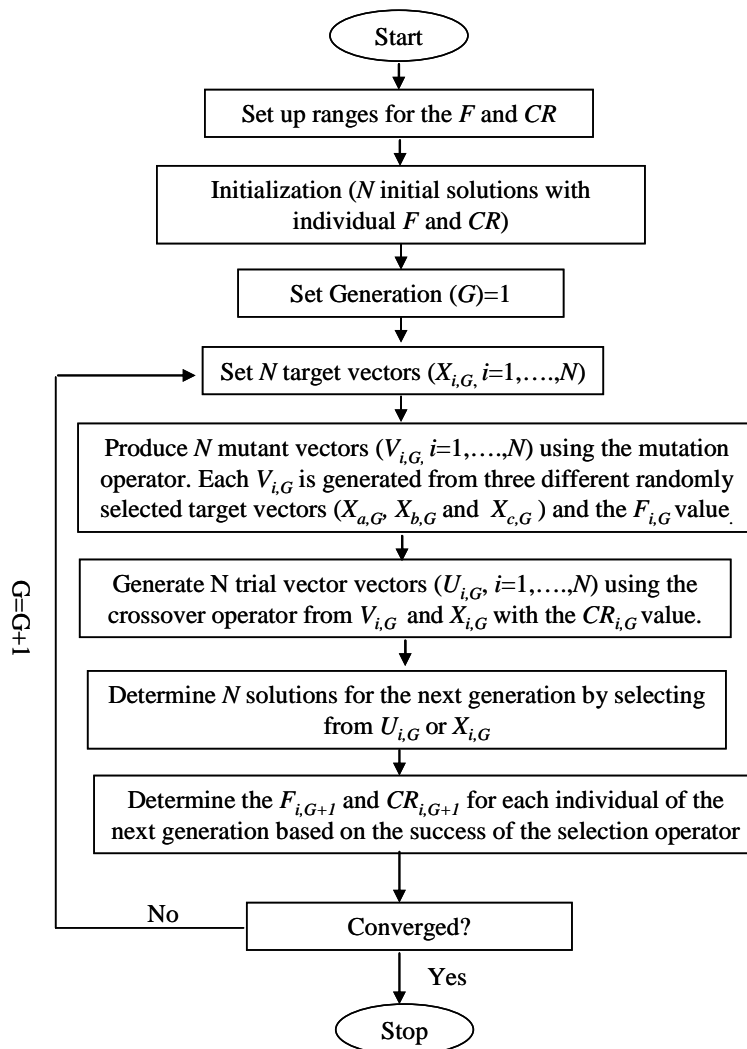


Figure 1 Flowchart of the proposed SADE algorithm

Initialization

The SADE algorithm is a population based stochastic search technique. Thus, an initial population is required to start the DE algorithm search. Normally, each initial population $X_{i,0} = \{ x_{i,0}^1, x_{i,0}^2, \dots, x_{i,0}^D \}$ is generated by uniformly randomizing individuals within the search space. In addition, initial values of the mutation factor F and crossover rate CR are randomly generated within a given range for each initial individual real-valued string. The initialization rule is given by:

$$x_{i,0}^j = x_{\min}^j + Rand_1(x_{\max}^j - x_{\min}^j) \quad i=1, 2, \dots, N, j=1, 2, \dots, D$$

$$F_{i,0} = F_l + Rand_2(F_u - F_l) \quad (1)$$

$$CR_{i,0} = CR_l + Rand_3(CR_u - CR_l)$$

where $x_{i,0}^j$ represents the initial value of the j^{th} parameter in the i^{th} individual at the initial population; x_{\min}^j and x_{\max}^j are the minimum and maximum bounds of the j^{th} parameter; $F_{i,0}$ and $CR_{i,0}$ are the initial values for the i^{th} individual; F_l and F_u are the minimum and maximum lower and upper bounds of the mutation weighting factor; CR_l and CR_u are the minimum and maximum lower and upper bounds of the crossover rate; $Rand_1$, $Rand_2$ and $Rand_3$ represent three independently uniformly distributed random variables in the range $[0, 1]$; N and D are population size and dimension of the vector (number of decision variables) respectively. The population size N is not changed during the SADE evolution process.

In the proposed SADE algorithm, the F and CR values are appended to the actual solution strings as shown in Figure 2.

$X_{1,G}$	$F_{1,G}$	$CR_{1,G}$
$X_{2,G}$	$F_{2,G}$	$CR_{3,G}$
...
$X_{N,G}$	$F_{N,G}$	$CR_{N,G}$

Figure 2 Encoding for the proposed SADE algorithm

where G is the generation number and $G=0$ is the initial generation. These F and CR values will evolve along with their corresponding actual solutions.

Mutation

Before the mutation operator is applied, each vector $X_{i,G}$ in the current population is treated as the target vector. Corresponding to each target vector, a mutant vector $V_{i,G} = \{v_{i,G}^1, v_{i,G}^2, \dots, v_{i,G}^D\}$ is generated by adding the weighted difference between two random vectors to a third vector (the base vector) from the current population (D is the number of decision variables). The $F_{i,G}$ value of each target vector $X_{i,G}$ is used to generate the mutant vector, which is given by:

$$V_{i,G} = X_{a,G} + F_{i,G}(X_{b,G} - X_{c,G}) \quad (2)$$

where $X_{a,G}$, $X_{b,G}$, $X_{c,G}$ are three vectors randomly selected from the current population ($a \neq b \neq c$). These three indices are randomly generated for each mutant vector $V_{i,G}$. A total of N mutant vectors, one for each target vector in the population, are produced using Equation (2).

Crossover

A trial vector $U_{i,G} = \{u_{i,G}^1, u_{i,G}^2, \dots, u_{i,G}^D\}$ is produced by selecting solution component values from either mutant vector ($V_{i,G}$) or its corresponding target vector ($X_{i,G}$) using a crossover process that is similar to uniform crossover. Thus, each component within the trial vector $U_{i,G}$ becomes:

$$u_{i,G}^j = \begin{cases} v_{i,G}^j, & \text{if } Rand_2 \leq CR_{i,G} \\ x_{i,G}^j, & \text{otherwise} \end{cases} \quad (3)$$

where $u_{i,G}^j$, $v_{i,G}^j$, $x_{i,G}^j$ are the j^{th} parameters in the i^{th} trial vector, mutant vector and target vector respectively. If $Rand_2$ is smaller than $CR_{i,G}$ ($0 \leq CR_{i,G} \leq 1$), the value $v_{i,G}^j$ in the mutant vector is copied to the trial vector. Otherwise, the value $x_{i,G}^j$ in the target vector is copied to the trial vector. A total of N mutant vectors $V_{i,G}$ and their corresponding target vectors $X_{i,G}$ are crossed over to generate N trial vectors using Equation (3).

Selection

After crossover, the objective function $f(U_{i,G})$ for each trial vector is evaluated. Then each trial vector $U_{i,G}$ is compared with the corresponding target vector $X_{i,G}$ in terms of objective function values. The vector with a smaller objective function value (given that a minimization problem is being considered) survives into the next generation ($X_{i,G+1}$).

That is

$$X_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases} \quad (4)$$

Thus, N solutions are selected utilizing Equation (4) to form the next generation.

The F and CR values in this proposed SADE algorithm are subject to the selection operator. If a combination of $F_{i,G}$ and $CR_{i,G}$ is able to generate a better solution $U_{i,G}$ compared to $X_{i,G}$, these two values are given to $X_{i,G+1}$ and survive to the next generation; in contrast, if $F_{i,G}$ and $CR_{i,G}$ generate a worse solution $U_{i,G}$ than $X_{i,G}$, then new randomly generated F and CR values are given to $X_{i,G+1}$. The F and CR selections for the next generation are given by:

$$F_{i,G+1} = \begin{cases} F_{i,G}, & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ F_l + Rand_3(F_u - F_l), & \text{if } f(U_{i,G}) > f(X_{i,G}) \end{cases} \quad (5)$$

$$CR_{i,G+1} = \begin{cases} CR_{i,G}, & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ CR_l + Rand_4(CR_u - CR_l), & \text{if } f(U_{i,G}) > f(X_{i,G}) \end{cases}$$

where $Rand_3$ and $Rand_4$ are independently generated random numbers in the range of $[0, 1]$.

As can be seen from Equations (1) to (5), the F and CR values are applied at the individual level and adjusted by means of evolution in the proposed SADE algorithm. It should be noted that neither the population size (N) and mutation strategy have not been included in the self-adaptation of the proposed SADE algorithm. For the population size (N), a sensitivity study has been undertaken to investigate its impact on the proposed SADE's performance in terms of WDS optimization. For the mutation strategy, it has

been demonstrated that the mutation strategy given in Equation (2) is most effective among a number of various mutation strategies introduced by Storn and Price 1995 (Zheng et al. 2011). Thus, the mutation strategy given in Equation (2) is used for the proposed SADE algorithm.

Convergence criterion

In the proposed SADE algorithm, the coefficient of variation ($C_{v,G}$) of the objective function values for the current DE population of solutions is used as the convergence criterion. The coefficient of variation is a concept commonly used in hydrology (Haan 1977). That is:

$$C_{v,G} = \frac{s_G}{\overline{OBJ}_G} = \sum_{i=1}^N \left(\frac{\sqrt{\frac{1}{N-1} \sum_{i=1}^N (OBJ_{i,G} - \overline{OBJ}_G)^2}}{\overline{OBJ}_G} \right) \quad (6)$$

where $C_{v,G}$ is the coefficient of variation of the objective function value based on all individuals at generation G ; s_G is the standard deviation for the N (population size) objective function values at population G . \overline{OBJ}_G is the average objective function value at generation of G ; The $C_{v,G}$ value reflects the convergence property of the SADE algorithm that has been run as when s_G approaches zero then all individuals of the population are similar in objective function values. The coefficient of variation is used to effectively non-dimensionalize the standard deviation with respect to the mean so that values are comparable across different case studies. This is an important advantage of the proposed new convergence criterion.

If $C_{v,G} < \text{Tol}$ (where Tol is an appropriately small value, say 10^{-6}), it indicates that all the individuals in the current population at generation G have already located final solutions (usually they will all be identical) and no further improvement can be made. If $C_{v,G} > \text{Tol}$, it is likely that not all individuals have converged on the same final solution and that better solutions may be able to be found as the SADE algorithm continues to explore the search space.

This proposed convergence criterion is new and motivated by the fact that all individuals in the DE tend to converge at the same final solution (Price et al. 2005). This convergence criterion significantly differs to the method of using the objective function values between two consecutive generations to terminate the EA evolution (Deb 2001). In the proposed convergence criterion approach, the search of SADE is terminated when all the individuals in the DE locate the same or extremely close final solutions, rather than using the differences of objective function values between two consecutive generations.

Self-adaptive differential evolution applied to the WDS optimization

The basic SADE algorithm is a continuous global optimization search algorithm. Therefore, the algorithm must be modified to solve the discrete WDS optimization problem. In this study, the decision variables included in the proposed SADE are the integers that represent the set of discrete pipe diameters. However, real continuous values are created in the mutation process in the proposed SADE algorithm. In the proposed method, these real values are truncated to the nearest integer number and hence mapped to the corresponding pipe diameters for the hydraulic analysis.

A network solver is used to compute the hydraulic balance in the proposed SADE method. For each individual, the network solver is called to perform the hydraulic simulation based on the pipe diameters decoded from integer string of this individual. As such, the head at each node of the WDS that is being optimized is obtained for each individual of the SADE, which, in turn, is used to assess the feasibility of each individual solution (a minimum allowable head requirement at each node usually needs to be satisfied when designing a WDS).

Constraint tournament selection is used in the proposed SADE to handle the constraints and determine the individuals survived in the next generation (Deb 2000). The algorithm when comparing two solutions (one is the trial vector solution and the other is the target

vector solution in the proposed SADE) in constraint tournament selection is given as follows:

- 1 The feasible solution is selected when compared with an infeasible solution;
- 2 The solution with a smaller value of objective function value (if cost is being minimized) is preferred between two feasible solutions;
- 3 The solution with less constraint violation is preferred between two infeasible solutions.

With this method, the comparison between the solutions in a tournament never happens in terms of both objective function and penalty function. In the first case, the solution with no head violation is preferred to the one with head violation and does not take the value of objective function into account. In the second case, the two solutions are compared based on the objective values and the one with a smaller value is selected as both solutions satisfy the constraints. In the last case, the solution with less head violation is selected and the value of the objective function is not considered. Thus, unlike traditional tournament selection, there is no need to specify a penalty multiplier in this proposed method.

CASE STUDIES

The SADE algorithm was developed in C++ and combined with the EPANET2 network solver (Rossman 2000). Four WDS case studies have been used to investigate the effectiveness of the proposed algorithm. These include the New York Tunnels Problem (NYTP) (Dandy et al. 1996), the Hanoi Problem (HP) (Fujiwara and Khang 1990), the Double New York Tunnels Problem (NYTP2) (Zecchin et al. 2005) and the Balerna network (BN) (Reca and Martínez 2006). The number of decision variables and the search space size for each case study is given in Table 1.

The ranges for the F and CR are generally between 0 and 1 (Storn and Price 1995). The recommended range for F is [0.5, 1.0] and for CR is [0.8, 1.0] (Price et al. 2005; Liu and

Lampinen 2005) based on testing on numerical optimization problems. In order to demonstrate the effectiveness of the self-adaptive algorithm, relatively larger ranges for the F and CR values were used in the proposed SADE algorithm. Both F and CR values in the range of $[0.1, 0.9]$ were utilized for each case study. For the SADE algorithm applied to the WDS optimization, convergence is taken to have occurred for $C_{v,G} < \text{Tol}$. For the computer runs presented in this research the Tol was set to be 10^{-6} .

Table 1 Summary of case study characteristics

WDS case study	Number of decision variables	Number of total available tunnel or pipe diameters that can be used	Search space size
NYTP	21	16	1.934×10^{25}
HP	34	6	2.865×10^{26}
NYTP2	42	16	3.741×10^{50}
BN	454	10	10^{454}

CONVERGENCE CRITERION ANALYSIS

The $C_{v,G}$ values at each generation for three SADE algorithm runs with different starting random number seeds applied to the NYTP case study is illustrated in Figure 3.

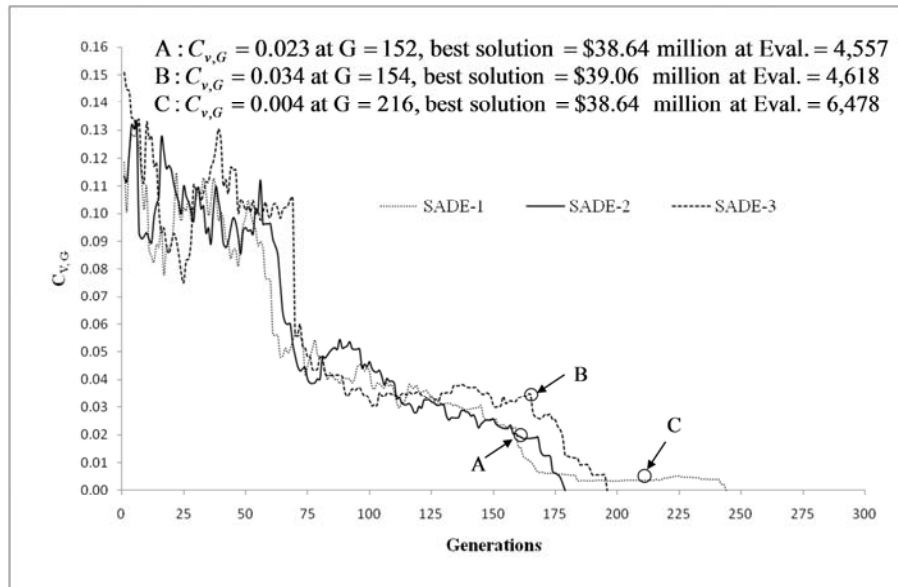


Figure 3 The $C_{v,G}$ values in each generation for three different SADE algorithm runs applied to the NYTP case study. Points A, B, and C reflect the points at which the best solution was found within each run.

When the SADE algorithm is run, as can be seen from Figure 3, the value of $C_{v,G}$ overall reduces as the number of generations increases. This shows that individuals in the SADE algorithm tend to be converging by means of evolution. The current best solution for the NYTP case study was first reported by Maier et al. (2003) with a cost of \$38.64 million. This best solution was initially found by SADE-2 run when $C_{v,G}=0.023$ at generation 152 (at 4,557 evaluations). Then all the individuals converged at this current best solution at generation 179 ($C_{v,G}<Tol$). The SADE-1 run first arrived at the current best solution when $C_{v,G}=0.004$ at generation 216 (at 6,478 evaluations) and finally converged at $C_{v,G}<Tol$ at generation 244. The SADE-3 run initially reached an optimal solution with a cost of \$39.06 million when $C_{v,G}=0.034$ at generation 154 (at 4,618 evaluations) and finally converged at this solution at generation 196. The SADE-3 was unable to reach the current best solution by the time the search was terminated at $C_{v,G}<Tol$.

From Figure 3, it can be seen that the SADE algorithm runs with different starting random number seeds consistently converged at $C_{v,G} < Tol$, although they require a different computational overhead. The search process varies for SADE runs starting with different random number seeds and hence each run may require different computational overheads to reach the same final solution. This is reflected by the fact that SADE-1 needed 244 generations for all individuals converge to the solution with a cost of \$38.64 million, while SADE-2 required 152 generations for all individuals to finally locate this solution. In this case, if a fixed computational budget is used to terminate the evolutions of EA runs, it is impossible to avoid the computational excess or insufficiency since each EA run with different starting random number seed requires different computational overhead. The proposed convergence criterion is able to overcome this disadvantage as convergence occurs based on the evolution feedback for each SADE run rather than specifying a fixed computational budget in advance. This allows SADE runs starting with different random number seeds to terminate their exploration at different numbers of generations purely based on the convergence criterion being satisfied.

It is also difficult to guarantee that each EA run with various starting random number seeds will find the same final solution. For the three different SADE runs given in Figure 3, SADE-1 and SADE-2 found the current best known solution (\$38.64 million) for the NYTP case study, while the best solution found by SADE-3 was \$39.06 million. The proposed convergence approach is able to indicate that no further improvement on the solution quality can be expected for the SADE-3 run although it has not arrived the current best known solution. This is because that all the individuals for the SADE-3 have converged at the identical final solution with a cost of \$39.06 million when $C_{v,G} < \text{Tol}$. Thus, providing a larger computational budget for the SADE-3 run for this particular random number seed would make no difference. Starting another SADE run with other starting random number seeds should be carried out if better solutions are required.

The convergence properties of the SADE algorithm in terms of $C_{v,G}$ applied to the other three case studies produced results similar to those exhibited by the NYTP case study and are therefore not given. From this study, it can be concluded that the proposed termination criterion with $C_{v,G} < \text{Tol}$ (see Equation (6)) for WDS optimization successfully avoids computation excess and insufficiency.

POPULATION SIZE STUDY

Table 2 gives the results of the proposed SADE applied to the four case studies with different population sizes. Multiple SADE runs with different random number seeds were performed for each case study in order to enable a reliable comparison.

The current best known solutions for the NYTP, HP and NYTP2 case studies were first reported by Maier et al. (2003), Reca and Martínez (2006) and Zecchin et al. (2005) with costs of \$38.64 million, \$6.081 million and \$77.28 million respectively. These current best known solutions were also found by the proposed SADE with different population sizes. The best solution found by the proposed SADE for BN case study was €1.983 million.

As shown in Table 2, in terms of percent with the best solution found and the average cost solution based on R runs with different starting random number seeds, the SADE algorithm with a larger population size performed better for each case study. However, the evaluations required to find optimal solutions and to converge using the proposed criterion ($C_{v,G} < \text{Tol}$: see Equation (6)) for the SADE with a larger population size are increased significantly as can be seen from Table 2. In considering both the solution quality and efficiency, population sizes of 50, 200, 100 and 500 were selected for the NYTP, HP, NYTP2 and BN case studies respectively. Note that for these population sizes selected that (i) the SADE algorithms exhibited good performance in solution quality and require a reasonably small computational overhead; and (ii) a further increase in population size for each case study only slightly improves the solution quality at the expense of a significantly increased computational overhead.

Table 2 Results of the SADE with different population sizes

Case study	Population size (N)	Best solution found ^a	Percent with the best solution found (%)	Average cost solution ^a	Average number of evaluations to find the final solutions	Average number of evaluations to converge
NYTP ($R=50$)	30	38.64	64	38.94	4,069	5,375
	50	38.64	92	38.64	6,584	9,227
	100	38.64	98	38.64	12,874	19,270
HP ($R=50$)	100	6.081	56	6.145	38,210	45,848
	200	6.081	84	6.090	60,532	74,876
	300	6.081	84	6.090	125,454	170,724
NYTP2 ($R=50$)	100	77.28	90	77.28	33,810	40,812
	200	77.28	98	77.28	70,196	87,592
	300	77.28	100	77.28	109,446	167,472
BN ($R=10$)	500	1.983	10	1.995	1.2×10^6	1.3×10^6
	1000	1.983	10	1.986	4.1×10^6	4.2×10^6
	2000	1.983	10	1.985	8.5×10^6	8.7×10^6

R =number of runs using different starting random number seeds. ^athe cost unit for the NYTP and HP case studies is \$ million and the cost unit for the BN case study is € million.

By comparing the number of decision variables (given in Table 1) and the selected population sizes for each case study (50 for the NYTP, 200 for the HP, 100 for the NYTP2: and 500 for the BN), an approximate heuristic guideline for the population size of the SADE algorithm applied to a WDS case study is within $[1D \ 6D]$, where D is the

number of decision variables for the WDS. This differs with the rule of thumb for the GAs in that the population size should be within $[5D \ 10D]$. The results of the SADE algorithm with population sizes of 50, 200, 100 and 500 for the NYTP, HP, NYTP2 and BN respectively are now used to compare results with other optimization techniques that have been previously applied to these four case studies.

SADE ALGORITHM PERFORMANCE COMPARISON AND DISCUSSION

Case study 1: New York Tunnels Problem (NYTP: 21 decision variables)

Table 3 gives the results of the proposed SADE and other previously published results for the NYTP case study. The results including the best solution found, the percentage of different runs with the best known solution found, the average cost solution and the average number of evaluations. The results in Table 3 are ranked based on the percent of trials with best solution found (the column 4).

Table 3 Summary of SADE and other EAs applied to the NYTP case study

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Algorithm ⁹	No. of runs	Best solution (\$M)	Percent of trials with best solution found	Average cost (\$M)	Average evaluations to find first occurrence of the best solution	Maximum allowable evaluations or evaluations for convergence ⁸
SADE ¹	50	38.64	92%	38.64	6,598	9,227
GHEST ²	60	38.64	92%	38.64	11,464	-
HD-DDS ³	50	38.64	86%	38.64	47,000	50,000
Suribabu DE ⁴	300	38.64	71%	NA	5,492	10,000
Scatter Search ⁵	100	38.64	65%	NA	57,583	-
MMAS ⁶	20	38.64	60%	38.84	30,700	50,000
PSO variant ⁷	2000	38.64	30%	NA	NA	80,000

¹Results from this study. ²Bolognesi et al. (2010). ³Tolson et al. (2009). ⁴Suribabu (2010). ⁵Lin et al. (2007). ⁶Zecchin et al. (2007). ⁷Montalvo et al. (2008). ⁸Average evaluations to final convergence. ⁹Results are ranked based on column (4).

As can be seen from Table 3, the proposed SADE algorithm was able to locate the current best solution with a frequency of 92%, which is the same or higher than other

EAs reported in Table 3. It should be highlighted that the proposed SADE algorithm is significantly more efficient than the majority of other EAs to find the optimal solutions in terms of average number of evaluations. As clearly shown in Table 3, the average number of evaluations required to find the first occurrence of optimal solutions based on 50 different SADE algorithm runs was 6,598, which is less than those required by the majority of other EAs given in Table 3. More importantly, the average number of evaluations required for final convergence of the SADE algorithm (when $C_{v,G} < \text{Tol}$) was 9,227, which is significantly less than the maximum number of allowable evaluations used for other EAs given in the last column of Table 3.

Case study 2: Hanoi Problem (HP: 34 decision variables)

Table 4 gives a performance summary of the proposed SADE algorithm and other optimization techniques applied to the HP case study. The proposed SADE algorithm found the current best solution for the HP case study with a success rate of 84%, which is an improvement compared to other EAs given in Table 4. The SADE algorithm also produced the lowest average cost solution over the 50 different runs as shown in Table 4 with a cost of \$6.090 million, which deviates only 0.15% from the known best solution.

Table 4 Summary of SADE and other EAs applied to the HP case study

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Algorithm ¹⁰	No. of runs	Best solution (\$M)	Percent of trials with best solution found	Average cost (\$M)	Average evaluations to find first occurrence of the best solution	Maximum allowable evaluations or evaluations for convergence ⁹
SADE ¹	50	6.081	84%	6.090	60,532	74,8769
Suribabu DE ²	300	6.081	80%	NA	48,724	100,000
Scatter Search ³	100	6.081	64%	NA	43,149	-
GHEST ⁴	60	6.081	38%	6.175	50,134	-
GENOME ⁵	10	6.081	10%	6.248	NA	150,000
HD-DDS ⁶	50	6.081	8%	6.252	100,000	100,000
PSO variant ⁷	2000	6.081	5%	6.310	NA	500,000
MMAS ⁸	20	6.134	0%	6.386	85,600	100,000

¹Results from this study. ²Suribabu (2010). ³Lin et al. (2007). ⁴Bolognesi et al. (2010). ⁵Reca and Martínez (2006). ⁶Tolson et al. (2009). ⁷Montalvo et al. (2008). ⁸Zecchin et al. (2007). ⁹Average evaluations to final convergence. ¹⁰Results are ranked based on column (4).

In terms of efficiency, the proposed SADE algorithm with an average number of evaluations of 60,532 did not perform as well as the DE (Suribabu 2010), Scatter Search algorithm (Lin et al. 2007) and GHEST (Bolognesi 2010). However, in terms of comparing the total computational overhead for each run, the average number of evaluations required for convergence (when $C_{v,G} < \text{Tol}$) of the proposed SADE algorithm was 74,876, which is less than the maximum number of evaluations used of the other EAs.

It should be highlighted that the results of other EAs in Table 4 were based on fine-tuning parameter values and only the final results with the calibrated parameter values are reported. In reality, adjusting the parameter values for these EAs by trial-and-error method required a large computational overhead. In contrast, for the proposed SADE, ranges of the F [0.1, 0.9] and CR [0.1, 0.9] were used for the HP case study and no tuning was conducted for these two parameters.

Case study 3: Double New York Tunnels Problem (NYTP2: 42 decision variables)

In order to enable a comparison with the proposed SADE, the traditional DE algorithm was also applied to the NYTP2 case study. The population size of 100 was also used in the traditional DE algorithm. Values of $F=0.5$ and $CR=0.6$ were found to be appropriate for the NYTP2 case study based on trials of different parameter values. The newly proposed convergence criteria was also used for the traditional DE. The results of the proposed SADE algorithm, the traditional DE algorithm and other optimization techniques that have been previously applied to the NYTP2 are given in Table 5.

The proposed SADE algorithm outperformed the traditional DE algorithm, the HD-DDS (Tolson et al. 2009) and MMAS (Zecchin et al. 2007) in terms of the percentage of trials with the best solution found. This is reflected from Table 5 that the proposed SADE found the current best solution for the NYTP2 case study with a frequency of 90%, which is higher than all the other EAs given in Table 5.

For the NYTP2 case study, the proposed SADE exhibited a notably better performance in terms of efficiency than other EAs presented in Table 5, as it required a significantly lesser average number of evaluations (33,810) to find the first occurrence of optimal solutions. The average evaluations required for convergence of 50 different SADE runs applied to the NYTP case study was 40,812. This shows the computational overhead for each proposed SADE run was significantly reduced compared with other EAs that terminated the run using a maximum number of allowable evaluations. A convergence comparison between the proposed SADE algorithm run and a traditional DE algorithm run with the same starting number seeds is illustrated in Figure 4.

Table 5 Summary of SADE and other EAs applied to the NYTP2 case study

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Algorithm ⁵	No. of runs	Best solution (\$M)	Percent of trials with best solution found	Average cost (\$M)	Average evaluations to find first occurrence of the best solution	Maximum allowable evaluations or evaluations for convergence ⁴
SADE ¹	50	77.28	90%	77.28	33,810	408,124
Traditional DE1 (F=0.5, CR=0.6) ¹	50	77.28	86%	77.28	70,104	874,574
HD-DDS ²	20	77.28	85%	77.28	310,000	300,000
MMAS ³	20	77.28	5%	78.20	238,300	300,000

¹Results from this study. ²Tolson et al. 2009. ³Zecchin et al. 2007. ⁴Average evaluations to final convergence. ⁵Results are ranked based on column (4).

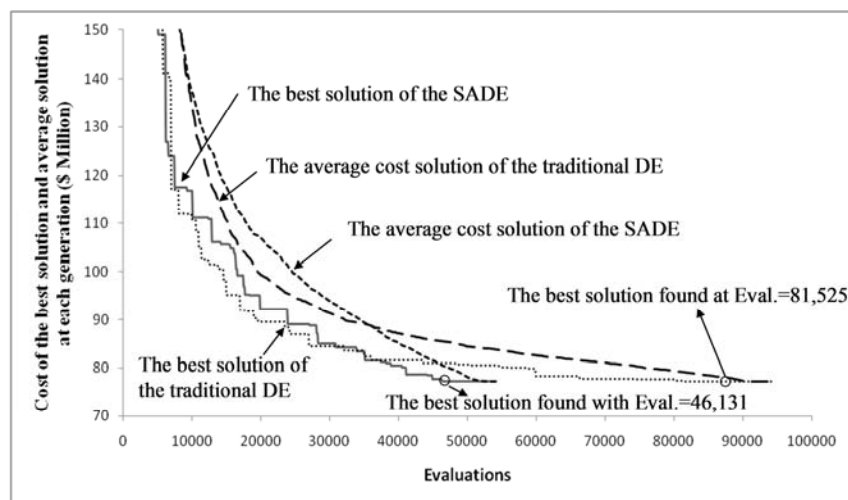


Figure 4 Convergence properties of the SADE and the traditional DE for the NYTP2 case study with the same random number seed of 100.

As can be seen from Figure 4, at evaluation numbers smaller than 30,000, the traditional DE algorithm found the best solution slightly faster than the proposed SADE algorithm when starting with the same random number seeds. In terms of comparing the average cost solution obtained at each generation, the traditional DE algorithm performed better than the proposed SADE algorithm at evaluation numbers smaller than 30,000 as it generated a lower average cost solution than the SADE algorithm. This is due to the fact that the F and CR values for the traditional DE algorithm have been fine-tuned, while the F and CR values in the SADE algorithm are initially randomly generated and in the early stages of generation have not yet self-adapted.

As clearly shown in Figure 4, the SADE algorithm was able to converge faster than the traditional DE algorithm in later generations (that is after 35,000 evaluations) in terms of finding the best solution as well as the best average cost solution. This is because the F and CR parameter values have been maturely evolved. Thus, the proposed SADE algorithm exhibits an improved performance for later generations. The proposed SADE algorithm found the current best solution at evaluation number 46,131 and converged at 54,100 evaluations based on the convergence criterion in Equation 6 ($C_{v,G} < Tol$), while the traditional DE algorithm found the current best solution for the NYTP2 case study with 81,525 evaluations and finally converged at 94,382 evaluations.

Case study 4: Balerma Network (BN: 454 decision variables)

In comparison, a traditional DE algorithm with a population size of 500, $F=0.3$ and $CR=0.5$ (these two values were selected after a number of fine-tuning trials) was performed for the BN case study. The newly proposed convergence criteria was used for the traditional DE applied to the BN case study. Table 6 outlines the performance comparison of the SADE algorithm with different CR ranges, the traditional DE algorithm with tuned parameter values and other optimization techniques that have been previously applied to the BN case study.

As shown in Table 6, the best solution found by the proposed SADE algorithm for the BN case study was €1.983 million, which is higher than the best known solution (€1.940

million) reported by Tolson et al. (2009) using HD-DDS method, but lower than solutions reported by other EAs given in Table 6. However, the HD-DDS (Tolson et al. 2009) yielded the best solution of €1.940 million requiring 30 million evaluations, while the SADE algorithm used only 1.3 million average evaluations to finally converge.

The average number of evaluation required for the SADE algorithm to first reach the optimal solutions was 1.2 million, which is less than those required by most of the EAs given in Table 6. While GHEST (Bolognesi et al. 2009) converged more quickly, the quality of the final solution was worse than that produced by the proposed SADE.

Table 6 Summary of SADE and other EAs applied to the BN case study

(1)	(2)	(3)	(4)	(4)	(5)	(6)
Algorithm ⁶	No. of runs	Best solution (€M)	Percent with the best solution found (%)	Average cost (€M)	Average evaluations to find first occurrence of the best solution	Maximum allowable evaluations or evaluations for convergence ^a
HD-DDS ²	1	1.940	-	NA	NA	30×10 ⁶
SADE ¹	10	1.983	10	1.995	1.2×10 ⁶	1.3×10 ⁶
Traditional DE(F=0.3,CR=0.5) ¹	10	1.998	10	2.031	2.3×10 ⁶	2.4×10 ^{6a}
GHEST ³	10	2.002	10	2.055	2.5×10 ⁵	NA
HS ⁴	NA	2.018	NA	NA	10 ⁷	10×10 ⁶
GENOME ⁵	10	2.302	10	2.334	NA	10×10 ⁶

¹Results from this study. ²Tolson et al. 2009. ³Bolognesi et al. (2010). ⁴Geem (2009). ⁵Reca and Martínez (2006). ⁶Results are ranked based on column (3). NA means not available. ^aAverage evaluations to final convergence.

Table 7 Summary of computational effort of the SADE for each case study

WDS case study	Number of different runs	Average number of evaluations required to find the best solution (AE1)	Average number of evaluations required to terminate the SADE runs based on the proposed convergence criterion (AE2)	Percent (AE1/AE2)
NYTP	50	6,584	9,227	71.4%
HP	50	60,532	74,876	80.8%
NYTP2	50	33,810	40,812	82.8%
BN	10	1.2×10 ⁶	1.3×10 ⁶	92.3%

Table 7 gives an analysis of the computational effort required to find the best solutions and the computational effort used to terminate the SADE run (when $C_{v,G} < \text{Tol}$) based on the proposed convergence criterion (see Equation (6)). It was found that the average

number of evaluations required to find the first occurrence of the best solution was around 80% of that required for final convergence ($C_{v,G} < \text{Tol}$) of the SADE runs.

CONCLUSION

The performance of all EAs is sensitive to the parameters used. Determining effective parameter values for each WDS optimization problem, therefore, requires a number of trials with different parameter values. This results in a significant increase in computational overhead and hence reduces the attractiveness of EAs being used in engineering practice.

The proposed self-adaptive DE algorithm (SADE) method overcomes the challenge mentioned above. A total of five contributions are presented in this paper in terms of novelty and the computational advantage of the proposed SADE algorithm, which are given as follows:

- (1) The proposed SADE encodes the parameters (F and CR) onto the strings to be automatically adjusted by means of evolution. Consequently, it reduces the effort required for the trial-and-error process normally used to determine the effective parameters for use in the DE algorithm.
- (2) The F and CR values of the proposed SADE algorithm are applied at the individual level rather than the generation level, which differs with the traditional DE algorithm applied to the WDS optimization design.
- (3) A new convergence criterion has been proposed in the SADE algorithm to avoid pre-specifying convergence conditions. This convergence criterion is based on the coefficient of variation such that $C_{v,G} < \text{Tol}$. It has been successfully implemented as the termination condition for the SADE algorithm applied to the WDS optimization. This represents a significant advantage compared to other EAs, where the maximum number of allowable evaluations is required to be pre-specified.

(4) The only parameter value that needs to be provided for the proposed SADE is the population size. The population size is a relatively easy parameter to adjust since a slight variation of its value does not significantly impact the performance of the SADE. In addition, it has been derived in this study that a population size within $[1D, 6D]$ is an approximate heuristic for the proposed SADE applied to WDS case studies, which differs to the rule of thumb for the GAs in that the population size should be within $[5D, 10D]$ (Deb 2001), where D is the number of decision variables for the WDS that is being optimized.

(5) A total of four WDS case studies with the number of decision variable ranging from 21 to 454 have been used to verify the effectiveness of the proposed SADE algorithm. For the NYTP, HP and NYTP2 case studies, the SADE performed the best in terms of the percent of the best solution found and exhibited improved performance in convergence speed compared to the majority of other reported EAs. For the large BN case study, the proposed SADE also exhibited a comparable performance to other EAs. It should be highlighted that the results of *other* EAs (excluding the new SADE algorithm as proposed in this paper) in Table 3 to 6 were based on fine-tuning parameter values and only the final results with the calibrated parameter values are reported. In reality, adjusting the parameter values for these EAs by trial-and-error required a large computational overhead. In contrast, for the proposed SADE, ranges of the F $[0.1, 0.9]$ and CR $[0.1, 0.9]$ were used for each case study and no tuning was needed to be conducted for these two parameters. Given this fact, it may be fair to draw a conclusion that the proposed SADE was able to yield optimal solutions with greater efficiency than other EAs.

The proposed SADE provides a robust optimization tool for the optimization of the design of WDSs (or rehabilitation of an existing WDS). This is because (i) the proposed SADE algorithm does not require as much fine-tuning of parameter values nor pre-specification of a computational budget; and (2) the proposed SADE algorithm is able to find optimal solutions with good quality and great efficiency. In addition, the proposed

SADE algorithm can also be used to tackle other water network management problems such as leakage hotspot detection (Wu and Sage 2000), optimal valve operation (Kang and Lansey 2010) and contaminant detection (Weickgenannt et al. 2010). The potential benefit of the proposed SADE algorithm compared to other EAs that have been used to deal with these water network management optimization problems is that it would need significantly less effort to adjust the parameter values. This is a huge advantage especially dealing with the real-time optimization problems for WDSs (Kang and Lansey 2010), in which decisions have to be made in extremely limited time.

The utility of the proposed SADE algorithm has been demonstrated using the least-cost single objective WDS optimization problems in this paper. A natural extension of this proposed self-adaptation algorithm is to extend it to deal with the multi-objective WDS optimization problems, for which in addition to the cost, other objectives such as the reliability or greenhouse gases are considered in order to provide more practical solutions for WDS design. This extension is the focus of future work.

REFERENCES

- Alperovits, E., and Shamir, U. (1977). "Design of Water Distribution Systems." *Water Resources Research*, 13(6), 885-900.
- Bäck, T., Hoffmeister, F., and Schwefel, H.-P. (1991). "A survey to evolution strategies." R. K. Belew and L. B. Booker, Eds., *Proc. 4th Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, San Mateo, Calif., 2-9.
- Bolognesi, A., et al. (2010). "Genetic heritage evolution by stochastic transmission in the optimal design of water distribution networks." *Advances in Engineering Software*, 41, 792-801.
- Brest, J., Greiner, S., Boskovic, B., Mernik, M., and Zumer, V. (2006). "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems." *IEEE Transactions on Evolutionary Computation*, 10(6), 646-657.
- Cunha, M. C., and Sousa, J. (2001). "Hydraulic infrastructures design using simulated annealing." *Journal of Infrastructure Systems*, 7(1), 32-38.

- Dandy, G. C., Simpson, A. R., and Murphy, L. J. (1996). "An improved genetic algorithm for pipe network optimization." *Water Resources Research*, 32(2), 449-457.
- Dandy, G. C., Wilkins, A., and Rohrlach, H. (2010). "A methodology for comparing evolutionary algorithms for optimizing water distribution systems." *Proceedings of the 12th Water Distribution System Analysis Symposium*, Tucson, Arizona
- Deb, K. (2000). "An efficient constraint handling method for genetic algorithms." *J Comput. Methods Appl. Mech. Engrg.*, 186,311-338.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithm*, Wiley, London.
- Eiger, G., Shamir, U., and Ben-Tal, A. (1994). "Optimal design of water distribution networks." *Water Resour. Res.*, 30(9), 2637-2646.
- Eusuff, M. M., and Lansey, K. E. (2003). "Optimization of water distribution network design using shuffled frog leaping algorithm." *J. Water Resources Planning and Management*, 129(3), 210-225.
- Eiben, A. E., Hinterding, R., and Michalewicz, Z. (1999). "Parameter control in evolutionary algorithms." *IEEE Transactions on Evolutionary Computation*, 3(2), pp.124-141.
- Eiger, G., Shamir, U., and Ben-Tal, A. (1994). "Optimal design of water distribution networks." *Water Resources Research*, 30(9), 2637-2646.
- Fujiwara, O., Jenchaimahakoon, B., and Edirisinghe, N. C. P. (1987). "A modified linear programming gradient method for optimal design of looped water distribution networks." *Water Resources Research*, 23(6), 977-982.
- Fujiwara, O., and Khang, D. B. (1990). "A two-phase decomposition method for optimal design of looped water distribution networks." *Water Resources Research*, 23(6), 977-982.
- Geem, Z. W., Kim, J. H., and Loganathan, G. V. (2002). "Harmony search optimization: application to pipe network design." *Int. J. Model. Simulat.*, 22(2), 125-133.
- Geem, Z. W. (2009). "Particle-swarm harmony search for water network design." *Engineering Optimization*, 41(5), 297-311.
- Geem, Z. W., and Sim, K. B. (2010) "Parameter-setting-free harmony search algorithm." *Applied Mathematics and Computation*, 3881~3889.

- Gibbs, M. S., Maier, H. R., and Dandy, G. C. (2010). "Comparison of genetic algorithm parameter setting methods for chlorine injection optimization." *Journal of Water Resources Planning and Management*, 136(2), 288-291.
- Haan, C. T. (1977). *Statistical methods in hydrology*, the Iowa State University Press, Ames, Iowa.
- Kang, D., and Lansey, K. (2010). "Real-time optimal valve operation and booster disinfection for water quality in water distribution systems." *Journal of Water Resources Planning and Management*, 136(4), 463-473.
- Lansey, K. E., and Mays, L. W. (1989). "Optimization model for water distribution system design." *J. Hydraul. Eng.*, 115(10), 1401-1417.
- Lin, M.-D., Liu, Y.-H., Liu, G.-F., and Chu, C.-W. (2007). "Scatter search heuristic for least-cost design of water distribution networks." *Engineering Optimization*, 39(7), 857-876.
- Liu, J., and Lampinen, J. (2005). "A fuzzy adaptive differential evolution algorithm." *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 9(6), 448- 462.
- Maier, H. R., et al. (2003). "Ant colony optimization for the design of water distribution systems." *J. Water Resources Planning and Management*, 129(3), 200-209.
- Montalvo, I., et al. (2008). "A diversity-enriched variant of discrete PSO applied to the design of water distribution networks." *Engineering Optimization*, 40(7), 655-668.
- Murphy, L. J., and Simpson, A. R. (1992). "Pipe optimization using genetic algorithms." *Research Report No. R93*, Department of Civil Engineering, University of Adelaide.
- Perelman, L., and Ostfeld, A. (2007). "An adaptive heuristic cross-entropy algorithm for optimal design of water distribution systems." *Engineering Optimization*, 39(4), 413-428.
- Price, V. K., Storn, M. R., and Lampinen, A. J. (2005). *Differential evolution: A practical approach to global optimization*, Springer, Berlin.
- Qin, A. K., Huang, V. L., and Suganthan, P. N. (2009). "Differential evolution algorithm with strategy adaptation for global numerical optimization." *IEEE Trans, Evol. Compt.*, 13(2), 398-417.
- Quindry, G. E., Brill, E. D., and Liebman, J. C. (1981). "Optimization of looped water distribution systems." *J. Environmental Engineering, Div. Am. Soc. Civ. Eng.*, 107 (EE4).

- Reca, J., and Martínez, J. (2006). “Genetic algorithms for the design of looped irrigation water distribution networks.” *Water Resources Research*, 42, W05416,doi:10.1029/2005WR004383.
- Rossman, L. A. (2000). “*EPANET 2-User manual.*” National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency, Cincinnati, OH.
- Savic, D. A., and Waters G.A. (1997). “Genetic algorithms for least-cost design of water distribution networks.” *J. Water Resource Planning and Management*, 123(2), 67-77.
- Simpson, A. R., Dandy, G. C., and Murphy, L. J. (1994). “Genetic algorithms compared to other techniques for pipe optimization.” *J. Water Resources Planning and Management*, 120(4), 423-443.
- Storn, R., and Price, K. (1995). “Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous space.” *Technical Report*, International Computer Science Institute, Berkeley, CA.
- Suribabu, C. R., and Neelakantan T. R. (2006). “Design of water distribution networks using particle swarm optimization.” *Urban Water J.*, 3(2), 111-120.
- Suribabu, C. R. (2010). “Differential evolution algorithm for optimal design of water distribution networks.” *Journal of Hydroinformatics*, 12 (1), 66-82.
- Tolson, B. A., et al. (2009). “Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization.” *Water Resources Research*, 45, W12416, doi:10.1029/2008WR007673.
- Vasan, A., and Simonovic, S. P. (2010). “Optimization of water distribution network design using differential evolution.” *J. Water Resource Planning and Management*, 136(2), 279-287.
- Vesterstroem, J., and Thomsen, R. (2004). “A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems.” In *Proc. IEEE Congr. Evolutionary Computation*, Portland, 1980-1987.
- Weickgenannt, M., Kapelan, Z., Blokker, M., and Savic, D. A. (2010). “Risk-based sensor placement for contaminant detection in water distribution systems.” *Journal of Water Resources Planning and Management*, 136(6), 629-636.
- Wu, Z. Y., and Simpson, A. R. (2002). “A self-adaptive boundary search genetic algorithm and its application to water distribution systems.” *Journal of Hydraulic Research*, 40 (2), 191-203.

- Wu, Z. Y., and Walski, T. (2005). “Self-adaptive penalty approach compared with other constraint-handling techniques for pipeline optimization.” *J. Water Resources Planning and Management*, 131(3), 181-191.
- Wu, Z. Y., and Sage, P. (2006). “Water loss detection via genetic algorithm optimization-based model calibration.” *Proc., ASCE 8th Annual International Symposium on Water Distribution System Analysis*, Cincinnati, Ohio.
- Zecchin, A. C., et al. (2005). “Parametric study for an ant algorithm applied to water distribution system optimization.” *IEEE Tans. Evol. Comput.*, 9(2), 175-191.
- Zecchin, A. C., et al. (2007). “Ant colony optimization applied to water distribution system design: comparative study for five algorithms.” *J. Water Resources Planning and Management*, 133(1), 87-92.
- Zheng, F., Simpson, A. R., and Zecchin, A (2011). “Performance study of differential evolution with various mutation strategies applied to water distribution system optimization.” *Proc., World Environmental and Water Resources Congress 2011*, ASCE, 166-176.

Chapter 6. Journal Paper 4-Combined NLP-DE Method for WDS Optimization

6.1 Synopsis

A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems

In Chapters 3, 4 and 5, improving the performance of the evolutionary algorithms (EAs) (genetic algorithms and differential evolution) in terms of optimizing the design for water distribution systems (WDSs) is the main focus, which is the first aim of the research presented in this thesis (see Section 1.1 of Chapter 1). Another promising approach to enhance the effectiveness of the EAs is combining them with traditional deterministic optimization methods, such as linear programming (LP) and nonlinear programming (NLP). This idea is not new and a few hybrid optimization approaches that couple EAs with deterministic optimization methods have been proposed to tackle the optimal design problems for WDSs. The review of these hybrid optimization methods was presented in Section 2.4 of Chapter 2. However, the majority of currently available hybrid optimization techniques remain in the research domain due to their limitations. These limitations include the inclusion of impractical pipe solutions (split pipe solutions or continuous diameter pipe solutions), unacceptable computational budgets and the high likelihood of premature convergence. These therefore lead to a limited application for these hybrid algorithms to deal with real-world sized WDS design problems (see the discussion in Section 2.4.3 of Chapter 2).

This research aims to develop more sophisticated hybrid optimization techniques compared to the existing hybrid optimization approaches for WDS design. The hybrid optimization methods developed in this research aim to optimally design real-world sized water networks, which is the second objective of this study (see Section 1.1 of Chapter 1).

During this research, two advanced hybrid optimization techniques have been developed to deal with the WDS optimization problems, which are presented in Chapters 6 and 7

respectively. In Chapter 6 (this Chapter), a combined NLP-differential evolution optimization method is introduced. Three distinct steps are involved in the proposed optimization approach. In the first step, the shortest-distance tree within the looped water network is identified using the Dijkstra graph theory algorithm. In the second step, an NLP solver is employed to optimize the pipe diameters for the shortest-distance tree (chords of the shortest-distance tree are allocated the minimum allowable pipe sizes). Finally, in the third step, the original looped water network is optimized using a differential evolution (DE) algorithm seeded with diameters in the proximity of the continuous pipe sizes obtained in step two. As such, the proposed optimization approach combines the traditional deterministic optimization technique of NLP and with the emerging evolutionary algorithm DE via the proposed network decomposition.

Traditionally, in hybrid optimization methods, EAs have been normally used to determine the regions of optimal solutions, whereas deterministic optimization methods (such as LP or NLP) have been used to further explore the interior of these regions identified by EAs. In contrast, the new proposed NLP-DE combination model here differs from the traditional combination models in that an NLP is used first to identify the approximate region of the optimal solution, while an EA is employed to further search the interior of the region.

A total of four WDS case studies with the number of decision variables ranging from 21 to 454 are used to verify the effectiveness of the proposed NLP-DE method, in which two of them are real-world sized water networks. This work has been published on *Water Resources Research* and the paper is provided here.

It should be noted that the standard differential evolution (SDE), rather than the self-adaptive differential evolution (SADE) algorithm described in Chapter 5, was used in Chapters 6 to 9. This is because that the experimental runs for Chapters 6 to 9 were completed before the publication of the SADE algorithm (Zheng et al. (2012a) in the *Journal of Computing in Civil Engineering*). It also should be highlighted that the optimization methods described in Chapters 6 to 9 are new optimization frameworks, by

which evolutionary algorithms, deterministic optimization approaches and graph decomposition techniques are combined. The SDE algorithm was used in this research in order to demonstrate the effectiveness of the proposed optimization frameworks, although other evolutionary algorithms also can be used in these developed optimization frameworks given in Chapters 6 to 9. The excellent performance of these optimization methods is predominately due to the decomposition the full optimization problem to sub-problems by using the graph decomposition techniques. Thus, it is believed that the performance of these developed methods will not be significantly affected by the underlying evolutionary algorithms that are adopted.

The parameter values for the SDE algorithm applied to case studies in Chapters 6 to 9 have been tuned based on trying a number of different combinations. The SADE algorithm developed in this thesis (Chapter 5) has been demonstrated to exhibit a similar performance, if not better, than the SDE with tuned parameter values. Thus, the authors recommend that the SDE algorithm used in the developed optimization frameworks can be replaced by the SADE algorithm for future applications. This will remove the need to tune the parameter values of the DE algorithms and can further improve the attractiveness of the optimization approaches proposed in this thesis.

It is noted that the results from Tolson et al. (2009) in the thesis were obtained from Dr Tolson based on personal communication, which are slightly different with what has been published in Tolson et al. (2009). However, this does not affect the conclusion of this work.

Citation of Paper

Zheng, F., A. R. Simpson, and A. C. Zecchin (2011a). “A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems.” *Water Resources Research*, 47(8), W08531.

STATEMENT OF AUTHORSHIP (PUBLISHED)

Journal paper title: **A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems.**

Authors: Feifei Zheng¹, Aaron Zecchin² and Angus Simpson³

Corresponding author:

¹Feifei Zheng (Candidate): PhD student, School of Civil, Environmental and Mining Engineering, University of Adelaide

Wrote the manuscript, performed all analysis, developed the model and theory and acted as corresponding author.

I hereby certify that the statement of contribution is accurate .

Signed _____ Date _____

²Aaron Zecchin: Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide

Supervised the development of the method and reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

³Angus Simpson: Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide

Supervised the development of the method and reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

Water Resources Research, 47(8), W08531, 2011.

6.2 Journal Paper 4: A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems (Published in *Water Resources Research*)

Feifei Zheng, Angus R. Simpson and Aaron C. Zecchin

Abstract

This paper proposes a novel optimization approach for the least cost design of looped water distribution systems (WDSs). Three distinct steps are involved in the proposed optimization approach. In the first step, the shortest-distance tree within the looped network is identified using the Dijkstra graph theory algorithm, for which an extension is proposed to find the shortest-distance tree for multi-source WDSs. In the second step, a non-linear programming (NLP) solver is employed to optimize the pipe diameters for the shortest-distance tree (chords of the shortest-distance tree are allocated the minimum allowable pipe sizes). Finally, in the third step, the original looped water network is optimized using a differential evolution (DE) algorithm seeded with diameters in the proximity of the continuous pipe sizes obtained in step two. As such, the proposed optimization approach combines the traditional deterministic optimization technique of NLP and with the emerging evolutionary algorithm DE via the proposed network decomposition. The proposed methodology has been tested on four looped WDSs with the number of decision variables ranging from 21 to 454. Results obtained show the proposed approach is able to find optimal solutions with significantly less computational effort than other optimization techniques.

1. Introduction

In most cases, the design and construction of water distribution systems (WDSs) is costly, often in the order of millions of dollars for larger capital works. Thus, the optimization of WDSs has historically been investigated by many researchers in order to potentially save significant costs. The nonlinear relationship between pipe head loss and discharge, plus the discrete nature of pipe sizes that can be used, bring about many complexities for optimally designing WDSs. This is increasingly difficult for looped

WDSs, in which pipe flows and nodal heads are unknown quantities. Two aspects contribute to the nonsmoothness properties of the WDS optimization problems. These include: (1) the pipe diameter choices being composed of discrete sizes rather than being continuous decision variables; and (2) the nonlinear value term involving the velocity within the head loss equations. Generally, there are two different types of WDS optimization problems. One is the completely new WDS design problem, while the other one is the expansion of the existing WDSs (such as the optimal rehabilitation of WDSs where there are some already existing pipes).

Historically, a number of traditional optimization techniques have previously been applied to water network optimal design, including linear programming (LP) [Alperovits and Shamir 1977; Fujiwara *et al.* 1987; Bhave and Sonak 1992; Sonak and Bhave 1993] and non-linear programming (NLP) [Lansey and Mays 1989; Fujiwara and Khang 1990]. These methods are deterministic and exhibit fast convergence. Often convergence to local optimal solutions occurs due to the nonsmoothness properties of the WDS optimization problem. In addition, the final solution is usually given in terms of continuous pipe sizes or split pipe sizes, which represents a significant practical limitation.

In the last two decades, considerable research has been undertaken into the optimization of WDSs using evolutionary algorithms (EAs). EAs are able to handle discrete search spaces directly and are less likely to be trapped at local optima. The search strategy of EAs differs compared with traditional optimization techniques (such as LP or NLP) in that they explore the search space broadly based on stochastic evolution rather than on gradient information. Genetic algorithms (GAs) were one of the first EAs applied to the optimal design of WDSs [Murphy and Simpson 1992; Simpson *et al.* 1994, Savic and Walters 1997; Montesinos *et al.* 1999]. Other applications have included: Cunha and Sousa [2001] who employed simulated annealing; Geem *et al.* [2002] who developed a harmony search model; Eusuff and Lansey [2003] who proposed a shuffled frog leaping algorithm (SFLA); Maier *et al.* [2003] who applied an Ant Colony Optimization

approach; and *Suribabu and Neelakantan* [2006] who introduced particle swarm optimization (PSO). These techniques have been successfully applied to a number of WDS optimization problems, and have been demonstrated to be more effective in finding optimal solutions compared with traditional optimization techniques.

More recently, *Tolson et al.* [2009] developed a hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for WDS optimization and concluded that HD-DDS was as good as, if not better, than other EAs in terms of search ability, while being significantly more computationally efficient. The differential evolution (DE) algorithm is a relatively new optimization technique that has received attention recently within WDS optimization research. *Vasan and Simonovic* [2010] and *Suribabu* [2010] applied DE to the optimization of WDSs and concluded that the search ability of DE was found to be better than other EAs, such as GAs and Ant Colony Optimization. Generally, EAs have been demonstrated to be robust in finding optimal design solutions for WDSs. However, they are computationally expensive, especially when dealing with large scale WDSs.

In order to overcome the limitations of each optimization method (the deterministic and the EA approaches), a new optimization approach that incorporates both types of optimization techniques has been previously proposed by researchers. *Reis et al.* [2006] proposed a GA-LP model to obtain the optimized operation of reservoir systems. *Afshar et al.* [2009] developed a hybrid two stage GA-LP algorithm to optimize the design and operation of a nonlinear, nonconvex and large-scale cycle storage system. In terms of WDS design optimization, *Krapivka and Ostfeld* [2009] proposed a coupled GA-LP scheme for the least-cost pipe sizing of water networks. In this method, the optimization problem was decomposed into an “inner” and an “outer” problem. The “inner” LP was formulated and solved for a fixed set of flows, while the flows were altered in the “outer” using a GA. In their proposed optimization approach, an enumeration approach was initially used to identify all possible spanning trees for a looped water network. Then a LP solver was employed to optimize the pipe diameter sizes for each spanning tree to

allow the least-cost spanning tree to be determined. Lastly, the spanning tree chords were locked into the minimum permissible pipe diameters and the least-cost spanning tree was further optimized using the proposed coupled GA-LP technique. The main advantage of this approach is that the search space handled by the GA-LP is reduced as the chords of the spanning tree are set to be the minimum allowable pipe sizes and removed as decision variables. However, this approach is computationally expensive for finding the least-cost spanning tree since all the possible spanning trees need to be evaluated. This method is therefore limited in practical applications by the fact that it is impossible to evaluate all the spanning trees for a relatively large water network, and the global optimal solution for the original water network could be missed as the spanning tree chords are fixed by the minimum allowable pipe sizes in this method. An additional criticism is that a split-pipe approach is used in their proposed optimization technique. The new coupled optimization approach proposed in this paper overcomes the problems associated with earlier approaches.

The research presented in this paper employs a graph theory decomposition method to effectively combine the EA (DE) and NLP. Typically, graph theory has been frequently used to analyze network connectivity properties and reliabilities [Yang *et al.* 1996; Shinstine *et al.* 2002; Davidson *et al.* 2005, Deuerlein 2006], whilst little effort has been made to use graph decomposition in the optimization of WDSs.

The objective of this paper is to introduce a novel approach for dealing with two different types of WDS optimization problems (either a completely new design or the expansion of the existing WDS). Features of this new methodology include the use of an efficient graph theory algorithm in determining the shortest-distance tree for a looped water network, and the combination of a deterministic optimization technique (NLP) and an evolutionary optimization algorithm (DE). It is observed that, in most of the traditional combinations of optimization models [Reis *et al.* 2006; Afshar *et al.* 2009; Krapivka and Ostfeld 2009], EAs have been used to determine the regions of optimal solutions, whereas deterministic methods (such as LP) have been used to further explore the

interior of these regions identified by EAs. The new proposed combination model here differs with the traditional combination models in that an NLP is used to identify the approximate region of the optimal solution, while an EA is employed to further search the interior of the region. In this proposed approach, an NLP solver is used to optimize the pipe diameters for the shortest-distance tree within a continuous pipe diameter search space (as opposed to a discrete diameter search space). This continuous solution, complemented by the chords of the shortest-distance tree with minimum allowable pipe sizes, forms an approximately optimal solution for the original water network. A DE is then seeded in the vicinity of this approximately optimal solution, thereby allowing the DE search to concentrate only on promising regions of the search space. As a result, better quality solutions are expected to be reached more efficiently, and with a higher likelihood. A total of four WDS case studies, including an expansion of an existing WDS and three new designs where they are no existing pipes, have been used to verify the effectiveness of the proposed optimization approach.

2. Methodology

The three steps involved within the proposed methodology are outlined below.

2.1. Step 1-Shortest-distance tree

A WDS can be described as a graph G , in which, vertices of the graph represent the nodes of the WDS, and edges of the graph represent links between nodes. In graph theory, a connected graph without any loops is referred as a tree [Deo 1974].

For a looped WDS, a demand node i may have a number of alternative paths to receive water from the source node s . Of these paths between s and node i , the path with the shortest total length of edges is denoted as the shortest path for node i . If we take the shortest path from the source node s to each of the other demand nodes, then the union of these paths will be a tree T rooted at source node s . Every path in T from s is the shortest path in the original graph G . Such a tree is called the shortest-distance tree [Deo 1974].

The remaining edges of G that are not traversed by any shortest paths are termed as chords.

For a looped WDS, when a demand node has two or more alternative paths receiving flow from a source node, the assumption is that an effective way of delivering demand (for the optimal design) is along the shortest path [Kadu *et al.* 2008]. Thus, the shortest-distance tree is considered to be an optimal tree of the looped WDS, in that each demand node has one and only one shortest path to the source node. An example of shortest-distance tree and chord for a looped water network is given in Fig. 1.

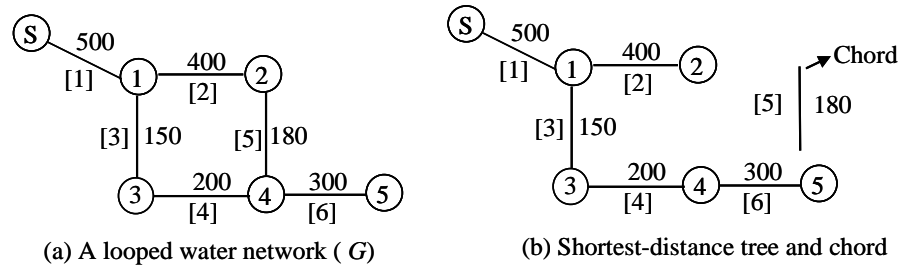


Figure 1 An example of shortest-distance tree and chord for a looped water network ((a) A looped water network (G), (b) Shortest-distance tree and chord)

The looped water network in Fig. 1(a) consists of five demand nodes labeled from 1 to 5, six links (with lengths in meters) and one source node s . Each node has one, or more than one, path to the source node s . Identification of the shortest-distance tree for such a simple looped water network can be carried out by visual inspection. All the paths from each demand node back to the source node s in this simple looped network are given in the third column of Table 1.

Table 1 Determination of shortest-distance tree (bold indicates shortest path)

Source node	Node number	Paths	Lengths (meters)	Shortest path
S	1	1-s	500	1-s
	2	2-1-s	900	2-1-s
		2-4-3-s	1030	
	3	3-1-s	650	3-1-s
		3-4-2-1-s	1280	
	4	4-2-1-s	1080	4-3-1-s
		4-3-1-s	850	
	5	5-4-2-1-s	1380	5-4-3-1-s
		5-4-3-1-s	1150	

As can be seen from Table 1, node 1 to s has only one path, and hence this is the shortest path from node 1 to s . For node 2, there are two alternative paths including path 2-1- s and 2-4-3- s to s . The path 2-1- s has the shorter length with value of 900 meters. Thus, the path of 2-1- s is the shortest path from node 2 to s . The shortest path for each node is identified as shown in the fifth column of Table 1. As a result, the shortest-distance tree is formed with these shortest paths as shown in Fig. 1(b). The remaining link 5 is the chord of the shortest-distance tree.

For this looped water network with limited alternative paths, a complete enumeration approach can be used to compute the sum of lengths of each alternative path for a node, thereby directly identifying the shortest path. However, the complete enumeration approach becomes intractable for larger water networks.

2.1.1. The Dijkstra algorithm

An efficient graph theory algorithm, called the Dijkstra algorithm [Deo 1974], is employed to identify the shortest-distance tree for complex water networks. The Dijkstra algorithm works by iteratively assigning and updating labels for each node indicating to the shortest path found so far for that particular node. For the source, a permanent label 0 is assigned. A permanent label is given to a vertex once the shortest path from this vertex to source vertex has been determined. The value of the permanent label is made equal to the sum of lengths of the shortest path. In contrast, a temporary label is given to a vertex for which the shortest path has not yet been identified. The value of this temporary label is set to be equal to the sum of lengths of the shortest path in the current iteration and this value is to be updated in later iterations. The Dijkstra algorithm is efficient in finding the shortest-distance tree for a looped network, especially for large and complex networks [Deo 1974]. The computational complexity (a proxy for execution time for the algorithm) for the Dijkstra algorithm implementation on a general graph with V vertices and E edges, is $O(V^2+E)$. The graph representing a WDS is sparse, thus the Dijkstra algorithm can be implemented more efficiently by storing the graph in the form of linked lists. In this case, the computational complexity time is $O((E+V)\log(V))$ [Deo 1974].

2.1.2. An extension of the Dijkstra algorithm

The Dijkstra algorithm is formulated for a single source node graph. In this paper, a supersource approach is used to extend the Dijkstra algorithm to handle systems with multi-source nodes. Variants of the supersource approach have been previously used to generate a treed network based on a looped network [Walters and Lohbeck 1993; Walters and Smith 1995]. The details on extension of the Dijkstra algorithm to deal with the multi-source WDS are given below.

For a multi-source WDS of k sources (reservoirs), an artificial supersource node is created to connect all the source nodes. Note that the lengths of the artificial links are set to be zero. The Dijkstra algorithm starts the search from the supersource node which is given a permanent label of zero (0). In the following step, each source of the WDS is also given a permanent label of zero. In the third step, all successors of the k sources are labeled as temporary with a value equal to the length between the successor and its corresponding source node. For each successor connected to more than one source, all the distances between this successor and its connecting sources are evaluated and the smallest value is given to this successor as the temporary label. Then, the Dijkstra algorithm is implemented to determine only one permanent label in the third step and the subsequent iterations. With this method, a complex WDS with k sources is decomposed into k different sub-networks connected via an artificial supersource node and k artificial links.

2.2. Step 2-Non-linear programming optimization

In Step 2, the objective is to find the lowest cost design for the shortest-distance tree network determined in Step 1, while satisfying the nodal head constraints. The objective function F is given by

$$F = a \sum_{i=1}^n D_i^b L_i \quad (1)$$

where D_i =diameter of pipe i , L_i =length of pipe i , a , b =specified coefficients and n =total number of pipes in the network.

Typically, the constraints for optimization of water networks include flow continuity at each node, energy conservation in each primary loop and the minimum allowable head requirement at each node. Since a tree network is optimized in this step, the discharges for each pipe of the tree network can be determined to satisfy continuity at each node. Two alternative methods are proposed in this study to determine the discharges of the shortest-distance tree for a WDS. The application of these two methods is dependent on the types of WDSs being optimized. The description of these two methods is given as follows:

Method 1: For the shortest-distance tree of the optimization problem of a completely new WDS, the flows in the chords of this WDS are assumed to be zero. Thus, the discharges for the pipes in the shortest-distance tree network are determined accordingly.

Method 2: For the shortest-distance tree of an expansion WDS optimization problem, an alternative method is proposed where flows in the chords are taken to be equal to that from the hydraulic analysis for the original WDS. The flows in the treed network pipes are then determined as the flows in the chords have assumed values.

In this study, for the expansion WDS optimization problem, the two methods mentioned above are tested to determine the most effective one. For the shortest-distance tree, energy conservation does not need to be considered in the formulation of the NLP as there are no loops involved in a treed network. Thus, the number of constraints for NLP in Step 2 is reduced significantly for the optimization of the shortest-distance tree produced in Step 1.

For the formulation of the NLP for optimizing the shortest-distance tree, the remaining constraints are the head constraint at each node and the diameter sizes that can be used. Since each node has a path to the source node, the head loss along this path should be less than a specified value that is equal to the head provided at source node minus the head required at this node. Two of the most frequently used formulas for head loss calculation in pipes are the Hazen-Williams (H-W) and Darcy-Weisbach (D-W)

equations [Walski 1984]. The constraint for each node (i) and these two formulas are given by

$$\sum_{k=1}^m h_{fk} \leq H_s - H_i^{\min} \quad (2)$$

$$\text{Hazen - Williams : } h_f = \omega \frac{L}{C^\alpha D^\beta} Q^\alpha \quad (3)$$

$$\text{Darcy-Weisbach : } h_f = f \frac{L}{D} \frac{V^2}{2g} \quad (4)$$

where, h_{fk} =head loss in pipe k, H_s =head at source node, H_i^{\min} =head requirement at node i , m =total pipes involved from node i to source node, ω =numerical conversion constant which depends on the units, α , β =coefficients, L = length of pipe (m), C =Hazen-Williams coefficient, D =diameter of pipe (m) and Q =pipe flow rates (m^3/s). In this study, $\alpha=1.852$ and $\beta=4.871$ are used. For SI units, *i.e.* the units of L , D , in meters, and Q in m^3/s , $\omega=10.667$ is used. In Equation (4), f =D-W friction factor for the pipe (dimensionless) and V =water velocity (m/s).

For the NLP formulated in this study, the diameters of pipes are treated as continuous variables, and the constraint for the diameters are given by

$$D_{\min} \leq D \leq D_{\max} \quad (5)$$

Where D_{\min} and D_{\max} are the minimum and maximum allowable pipe sizes respectively.

The continuous solution for the shortest-distance tree network, complemented by the chords of the shortest-distance tree set to the minimum allowable pipe diameters, is an approximately optimal solution for the original looped water network. For the final step, this approximately optimal solution needs to be replaced using commercially available discrete pipe sizes, and cannot be guaranteed to be the global optimal solution based on the assumption that was made in Step 1. To obtain the global optimal solution using commercially available pipe diameters based on the current solution achieved in Step 2, a DE algorithm is applied and the optimization is moved to Step 3.

2.3. Step 3-The differential evolution algorithm

The differential evolution (DE) algorithm, introduced by *Storn and Price* [1995], is found to be a relatively simple but powerful EA for global optimization. There are three important operators involved in the DE algorithm including the mutation operator, crossover operator and selection operator, which is quite similar to GAs. Several parameters that need to be determined in the use of the DE include population size (N), mutation weighting factor (F) and crossover rate (CR). A DE differs significantly compared to a GA in the mutation process such that the mutant solution is generated by adding the weighted difference (F) between two random population members to third member. The process of DE is given as follows.

2.3.1. Initialization

The DE is a population based stochastic search technique. Thus, a set of members of the initial population is required to initialize the DE search. Normally, each initial population $X_{i,0} = \{ x_{i,0}^1, x_{i,0}^2, \dots, x_{i,0}^D \}$ is generated by randomizing individuals from a uniform distribution within the search space, that is

$$x_{i,0}^j = x_{\min}^j + rand(0, 1)(x_{\max}^j - x_{\min}^j) \quad i=1, 2, \dots, N, j=1, 2, \dots, D \quad (6)$$

where $x_{i,0}^j$ =the initial value of the j^{th} parameter for the i^{th} individual in the initial population, x_{\min}^j and x_{\max}^j = the minimum and maximum bounds of the j^{th} parameter respectively, $rand(0, 1)$ represents a uniform distributed random variable in the range $[0, 1]$, while N and D =population size and dimension of the vector respectively. The population size is not changed during the DE evolution process.

2.3.2. Mutation

The DE is mainly driven by its mutation strategy compared with GAs. A mutant vector $V_{i,G}$ with respect to each individual $X_{i,G}$ is produced by adding the weighted difference (F) between two random population members to a third member from the current population. Each individual $X_{i,G}$ associated with a mutant vector is denoted as the target vector. A frequently used mutation strategy in DE is given as follows:

$$V_{i,G} = X_{r_1^i,G} + F(X_{r_2^i,G} - X_{r_3^i,G}) \quad (7)$$

where $V_{i,G}$ = the mutant vector with respect to the target vector of $X_{i,G}$ at generation G , $X_{r_1^i,G}$, $X_{r_2^i,G}$, $X_{r_3^i,G}$ are three vectors randomly selected from the current population ($r_1^i \neq r_2^i \neq r_3^i$). These three indexes are randomly generated for each mutant vector. F is the mutation weighting factor.

2.3.3. Crossover

After the mutation, a trial vector $U_{i,G}$ is generated through selecting solution component values of either from $X_{i,G}$ or $V_{i,G}$. In the basic DE version (Storn and Price 1995), uniform crossover is employed as:

$$u_{i,G}^j = \begin{cases} v_{i,G}^j, & \text{if } \text{rand}(0,1)_j \leq CR \\ x_{i,G}^j, & \text{otherwise} \end{cases} \quad (8)$$

where $u_{i,G}^j$, $v_{i,G}^j$, $x_{i,G}^j$ = the j^{th} parameter for the i^{th} trial vector, mutant vector and target vector respectively, CR is the crossover rate within the range of $[0, 1]$, $\text{rand}(0,1)_j$ is a random number between 0 and 1 generated for each parameter j . If $\text{rand}(0,1)_j$ is smaller than CR , the parameter $v_{i,G}^j$ in the mutant vector is copied to the trial vector, otherwise, the parameter $x_{i,G}^j$ in the target vector is copied to the trial vector.

2.3.4. Selection

After crossover, all the trial vectors are evaluated using the objective function $f(U_{i,G})$ and are compared with their corresponding trial vector objective function $f(X_{i,G})$. The vector with a lower objective function value (given a minimization problem) survives for the next generation. That is

$$X_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases} \quad (9)$$

Where $X_{i,G+1}$ is the i^{th} individual at the generation $G+1$.

Mutation, crossover and selection are repeatedly applied generation by generation until the stopping criterion is satisfied. It is observed that the basic DE is a continuous global optimization search algorithm. As a result, DE should be modified to solve discrete WDS

optimization problems. A new approach to deal with the truncation of the continuous variables to the available discrete pipe sizes is proposed. The continuous pipe sizes are rounded to the nearest commercially available pipe diameter after application of the mutation operator given in Equation (7). Each vector element is checked after application of the mutation operator. If its value is smaller or larger than the minimum or maximum allowable pipe size, then the minimum or maximum allowable pipe size is assigned. If its value is between two sequentially discrete pipe diameters, the discrete pipe diameter that is closest is assigned. In addition, constraint tournament selection is used in the DE to handle head constraints [Deb 2000].

The NLP continuous pipe diameter solution obtained in Step 2 is used to initialize or seed the population for DE optimization. In this study, the initial population of the DE was generated by randomly selecting pipe diameters for each decision variable from a set of limited options based on the NLP optimal solution instead of all available pipe diameters. The set of limited pipe diameter options is referred to as a seeding table for its corresponding pipe. Two different initial seeding tables are created for the continuous pipe size solution of the shortest-distance tree network. One seeding table consists of two adjacent pipe diameters, one having a discrete diameter that is immediately larger than the NLP continuous pipe size and the other having a discrete diameter that is immediately smaller. The other seeding table is composed of four adjacent pipe diameters, two having discrete diameters that are larger than the NLP continuous pipe size and the other two having discrete diameters that are smaller. The DE that is seeded with two pipe diameters is denoted as NLP-DE1, while the DE that is seeded with four pipe diameters is denoted as NLP-DE2. These two DEs that are seeded with different sizes from initial tables are applied to the four case studies. For the initial DE population, pipe diameters in the range of initial seeding tables are randomly selected. For each chord of the shortest-distance tree, the two and four adjacent minimum permissible pipe sizes are randomly selected for the NLP-DE1 and NLP-DE2 initial population respectively. It is noted that, with this approach, each decision variable has only two or four tailored optional pipe sizes to be randomly selected for starting the DE exploration.

Thus the initial solutions that need to be evolved are scattered in the region around the approximate-optimal solution produced in Step 2, rather than randomly distributed throughout the entire search space. It should be highlighted that the tailored seeding table obtained in Step 2 is used only to initialize the DE's search, and it does not necessarily specify a limited search space for the DE exploration. That is, in Step 3 the DE is not limited to only explore the interior region of the search space defined by the initial seeding table, but the search can expand to the region that is outside the initial seeding table. Hence the finally selected pipe diameters for some pipes may be outside those contained in the initial seeding table.

3. Case study results and discussion

The Dijkstra algorithm that is used in Step 1 and the DE that is used in Step 3 has been coded in C++. The NLP formulated in Step 2 is solved by software Lingo12 [*LINDO Systems Inc.*, 2009]. The DE application in Step 3 combines the EPANET2.0 solver [*Rossman* 2000]. Four case studies are used to verify the effectiveness of the proposed optimization approach including New York Tunnels Problem (NYTP), Hanoi Problem (HP), ZJ network (ZJ) and Balerna network (BN). The Hazen-Williams formula is used to calculate the head loss for the NYTP, HP and ZJ case studies and the Darcy-Weisbach formula is used for the BN case study. *Storn and Price* [1995] recommended the parameter ranges for the DE of $1D \leq N \leq 10D$, $0.3 \leq F \leq 0.9$, $0.5 \leq CR \leq 1.0$ as the DE with these parameter ranges showed generally favorable performance in terms of convergence properties. For each case study in this paper, a preliminary sensitivity analysis was performed to determine the effective N , F and CR values based on the range given by *Storn and Price* [1995] for each parameter.

3.1. Case study 1: New York Tunnels Problem (NYTP)

A schematic of the NYTP system is given as Fig. 2. The network has 21 existing tunnels and 20 nodes fed by a fixed-head reservoir. The details of this network, including the head constraints, pipe costs and water demands are given by *Dandy et al.* [1996]. The

objective is to determine which pipe should be installed in parallel with the existing pipes such that the cost is minimized while satisfying the minimum head requirement at all nodes. There are 15 pipe diameters that can be selected for the NYTP. In addition, a zero pipe size provides a total of 16 options (15 actual pipe diameters plus a zero pipe size) for each link. Thus the total search space is 16^{21} (approximately 1.934×10^{25}).

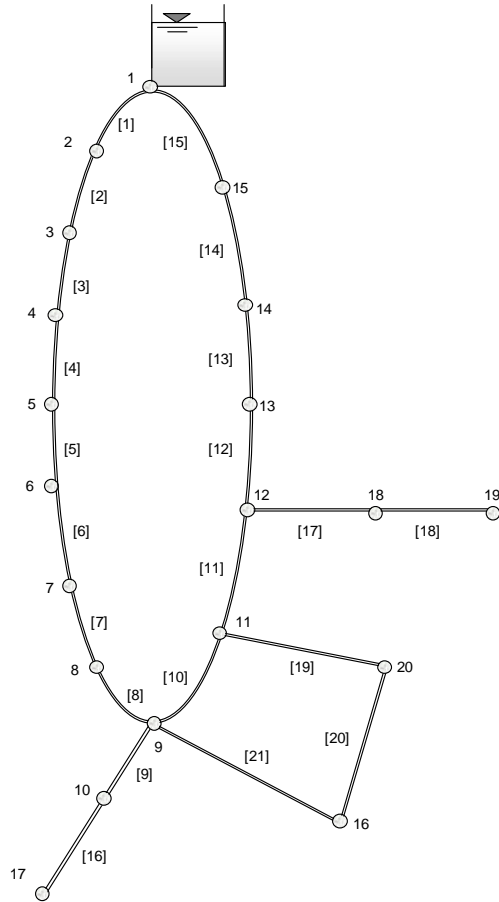


Figure 2 The layout of the New York Tunnels

In Step 1, the Dijkstra algorithm is applied for the NYTP network to identify the shortest-distance tree. The identified shortest-distance tree is given in Fig. 3. As shown in Fig. 3, pipes 10 and 20 are identified as the chords and all the other pipes form the shortest-distance tree. Since the NYTP is an existing water network and the diameters of chords (pipes 10 and 20) are known, the two proposed methods (see section 2.2) are used to determine the flow distribution for the shortest-distance tree. The flow results for the

shortest-distance tree determined by methods 1 and 2 described in section 2.2 are given in the second and third columns of Table 2.

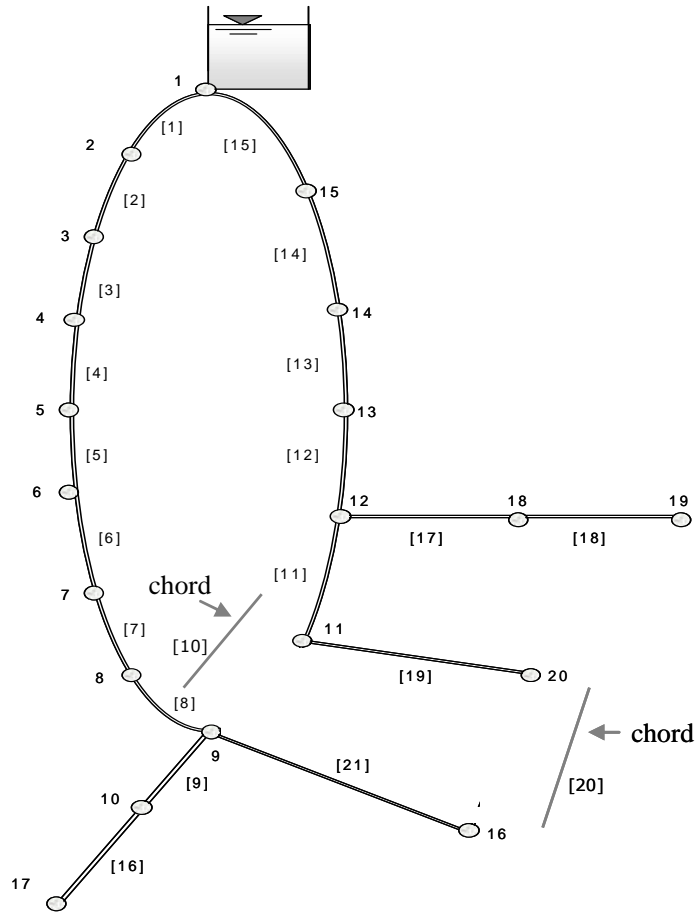


Figure 3 The layout of the shortest-distance tree of the NYTP network

In Step 2, two separate NLPs are formulated for the shortest-distance tree with two sets of different flow distributions and solved. The two NLP continuous solutions as shown in the fourth and fifth columns of Table 2 complemented by chords of the shortest-distance tree with minimum pipe sizes (0 inch for the NYTP case study) produced optimal solutions with a cost of \$55.12 million and \$34.78 million respectively. Thus, the optimal solution produced by the assumption that flows in the chords are taken to be equal to that from the hydraulic analysis for the original water network (Method 2 in section 2.2), is better than that produced based on the assumption that no flows exist in these chords

(Method 1 in section 2.2). This indicates that Method 2 is more effective for a WDS optimization that includes existing pipes. The NLP solution based on the Method 2 (mentioned in section 2.2) is adopted for further analysis in this study. The final best solution obtained by the combined NLP-DE approach for the NYTP case study is given in the last column of Table 2. It is observed that the design of the solution obtained in Step 2 (in the fifth column) is close to the final best solution as it has 15 pipes with the same diameter of zero.

Table 2 Initial seeding tables for the NYTP case study and the combined NLP-DE results

Links	Flows[1] in the shortest- distance tree (m3/s)	Flows[2] in the shortest- distance tree (m3/s)	Pipe diameters (inches)				Combined NLP-DE final solution (inches)
			The NLP solution ^[1] produced in Step 2	The NLP solution ^[2] produced in Step 2	Initial seeding table for NLP- DE1	Initial seeding table for NLP- DE2	
1	29.00	24.48	157.16	0.00	0, 36	0, 36, 48, 60	0
2	26.39	21.86	138.78	0.00	0, 36	0, 36, 48, 60	0
3	23.77	19.24	0.00	0.00	0, 36	0, 36, 48, 60	0
4	21.27	16.75	0.00	0.00	0, 36	0, 36, 48, 60	0
5	18.77	14.25	0.00	0.00	0, 36	0, 36, 48, 60	0
6	16.28	11.75	0.00	0.00	0, 36	0, 36, 48, 60	0
7	13.78	9.25	144.93	111.31	108, 120	96, 108, 120, 144	144
8	11.28	6.76	124.58	0.00	0, 36	0, 36, 48, 60	0
9	1.65	1.66	0.00	0.00	0, 36	0, 36, 48, 60	0
10 [#]	0.00	4.86	0.00	0.00	0, 36	0, 36, 48, 60	0
11	9.63	14.16	0.00	0.00	0, 36	0, 36, 48, 60	0
12	19.58	24.10	0.00	0.00	0, 36	0, 36, 48, 60	0
13	22.89	27.42	0.00	0.00	0, 36	0, 36, 48, 60	0
14	25.51	30.04	0.00	0.00	0, 36	0, 36, 48, 60	0
15	28.12	32.65	0.00	0.00	0, 36	0, 36, 48, 60	0
16	1.63	1.63	67.13	72.94	72, 84	60, 72, 84, 96	96
17	6.63	6.63	91.50	100.39	96, 108	84, 96, 108, 120	96
18	3.31	3.32	72.53	80.01	72, 84	60, 72, 84, 96	84
19	4.81	4.48	53.68	59.31	48, 60	36, 48, 60, 72	72
20 [#]	0.00	0.33	0.00	0.00	0, 36	0, 36, 48, 60	0
21	4.81	5.15	71.55	76.03	72, 84	60, 72, 84, 96	72
Cost (\$M)	-	-	55.12	34.78	-	-	38.64

[#]—chords of the NYTP network. ^[1] Flows and NLP solution are determined based on the assumption that there are no flows in chords (pipes 10 and 20). ^[2] Flows and NLP solution are determined based on the assumption that flows in chords (pipes 10 and 20) are the same with that of performing the hydraulic analysis for the original water network.

Based on the continuous pipe diameter solution obtained in Step 2, two different initial seeding tables are created including seeding tables for NLP-DE1 and seeding tables for NLP-DE2 as shown in Table 2 (column 6 and 7).

Both DE applications were assumed to have identical parameters including population size (N), maximum allowable number of evaluations (MAE), mutation weighting factor (F) and crossover rate (CR), while seeded with different initial pipe diameters. For the NYTP case study, $N=50$, $MAE=20,000$, $F=0.7$ and $CR=0.8$ were used. A total of 100 different DE runs using different starting random number seeds were performed for each of these two DE applications.

The statistics of the results for the NYTP case study are given in Table 3. These include the best solution found, percentage of trials for which the current best solution was found, the average cost solution, the worst solution found and the average number of evaluations to find the best cost solution based on the different runs. For comparison, Table 3 also lists the results of other optimization techniques that have previously been used to optimize the NYTP case study.

Table 3 Algorithm performance for the NYTP case study

Algorithm	Number of different runs	Best solution found (\$M)	Percentage of trials with best solution found (%)	Average cost solution (\$M)	Worst solution (\$M)	Maximum number of allowable evaluations	Average number of evaluations to find best solutions
NLP-DE1	100	38.64	99	38.64	38.80	20,000	8,277
NLP-DE2	100	38.64	99	38.64	38.80	20,000	10,631
HD-DDS ¹	50	38.64	86	38.65	38.77	50,000	13,000
DE-Dandy ²	30	38.64	70	40.33	51.16	100,000	-
MMAS-ACO ³	20	38.64	60	-	-	50,000	30,711
Standard GA ⁴	1000	38.64	45	39.00	-	100,000	49,950
PSO ²	30	38.64	33	38.93	-	100,000	-
PSO variant ¹	2000	38.64	30	38.83	-	80,000	-

¹Tolson et al. [2009]. ²Dandy et al. [2010]. ³Zecchin et al. [2006]. ⁴Zheng et al. [2010].

NLP-DE1: DE seeded with 2 tailored pipe diameters based on NLP solution obtained in Step 2.

NLP-DE2: DE seeded with 4 tailored pipe diameters based on NLP solution obtained in Step2.

The best known solution for the NYTP case study is \$38.64 million first found by Maier et al. (2003) with the Ant Colony Optimization technique. This best known solution was also found by the proposed combined NLP-DE optimization technique. As shown in Table 3, NLP-DE1 and NLP-DE2 exhibited similar performance in terms of percentage of the best known solutions found. NLP-DE1 was only slightly better than NLP-DE2 in terms of convergence speed. For the NYTP case study, the proposed optimization algorithm variants located the current best solution with a frequency of 99%, which is higher than that of other optimization techniques including HD-DDS [Tolson et al. 2009], DE [Dandy et al. 2010], MMAS-ACO [Zecchin et al. 2006], GA [Zheng et al. 2010], PSO [Dandy et al. 2010] and PSO variant [Tolson et al. 2009] as shown in Table 3.

In terms of efficiency, the proposed new algorithm exhibited the best performance on the NYTP case study as it was able to locate the best known solution faster than other algorithms as shown in Table 3. The maximum allowable evaluations for the NYTP case study was 20,000 and the average number of evaluations required to find the best solution for NYTP case study were 8,277 for NLP-DE1 and 10,631 for NLP-DE2. Both values are far less than those of other optimization techniques.

3.2. Case study 2: Hanoi Problem (HP)

The Hanoi Problem (HP) is a network design where all new pipes are to be selected. The layout of HP network is given in Fig. 4. The network is comprised of 34 pipes and 32 nodes which are fed by a single reservoir with a head of 100 meters. The minimum head requirement of the other nodes is 30 meters. A total of six pipe diameters of {12, 16, 20, 24, 30, 40} inches can be selected for each new pipe. The total search space is $6^{34} \approx 2.8651 \times 10^{26}$. The Hazen-Williams coefficient for each new pipe is 130. Details of this network and the formulation of the cost for pipes are given in *Fujiwara and Khang [1990]*.

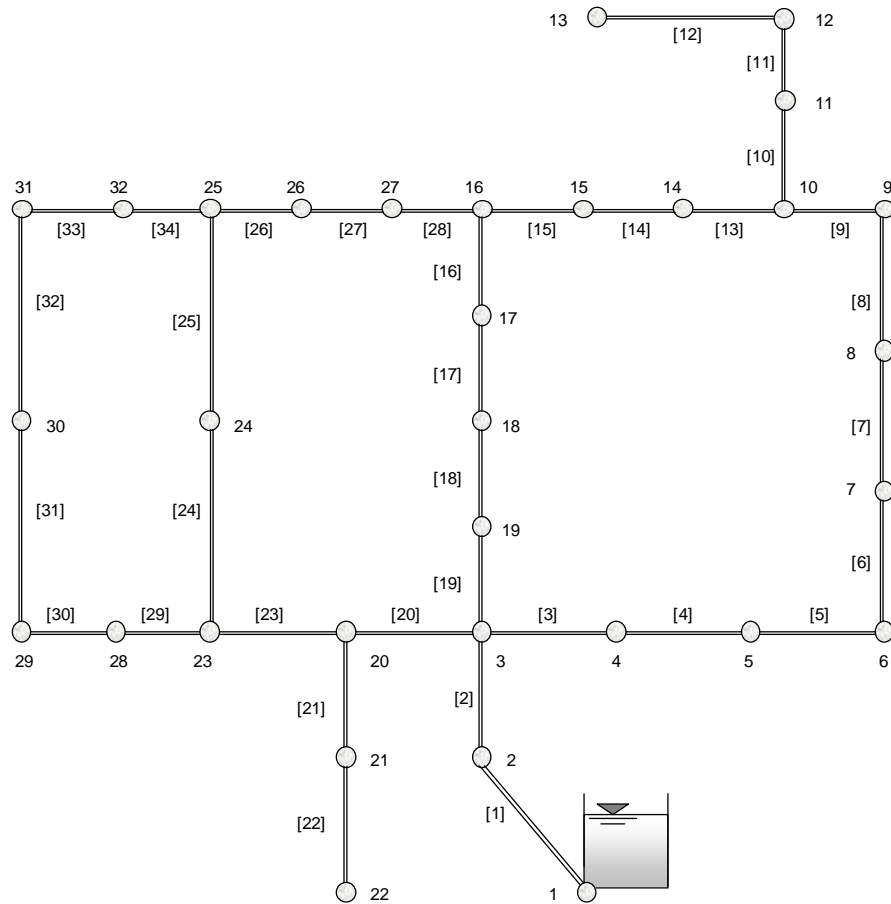


Figure 4 The layout of the Hanoi Problem network

In Step 1, the shortest-distance tree for the HP network is shown in Fig. 5 based on the Dijkstra algorithm. As can be seen from Fig. 5, pipes 13, 26 and 31 are identified to be the chords. All the discharges in links can be determined based on Method 2 described in section 2.2 for this shortest-distance tree as shown in the second column of Table 4.

An NLP is formulated for the shortest-distance tree of the HP network and solved in Step 2. The continuous pipe diameters solution is given in the third column of Table 4. This solution, complemented by chords with minimum pipe sizes (12 inches for the HP case study), produced an approximately optimal solution with a cost of \$5.924 million. . The final best solution produced by the combined NLP-DE approach, for the HP case study applied in Step 3, is given in the last column of Table 4.

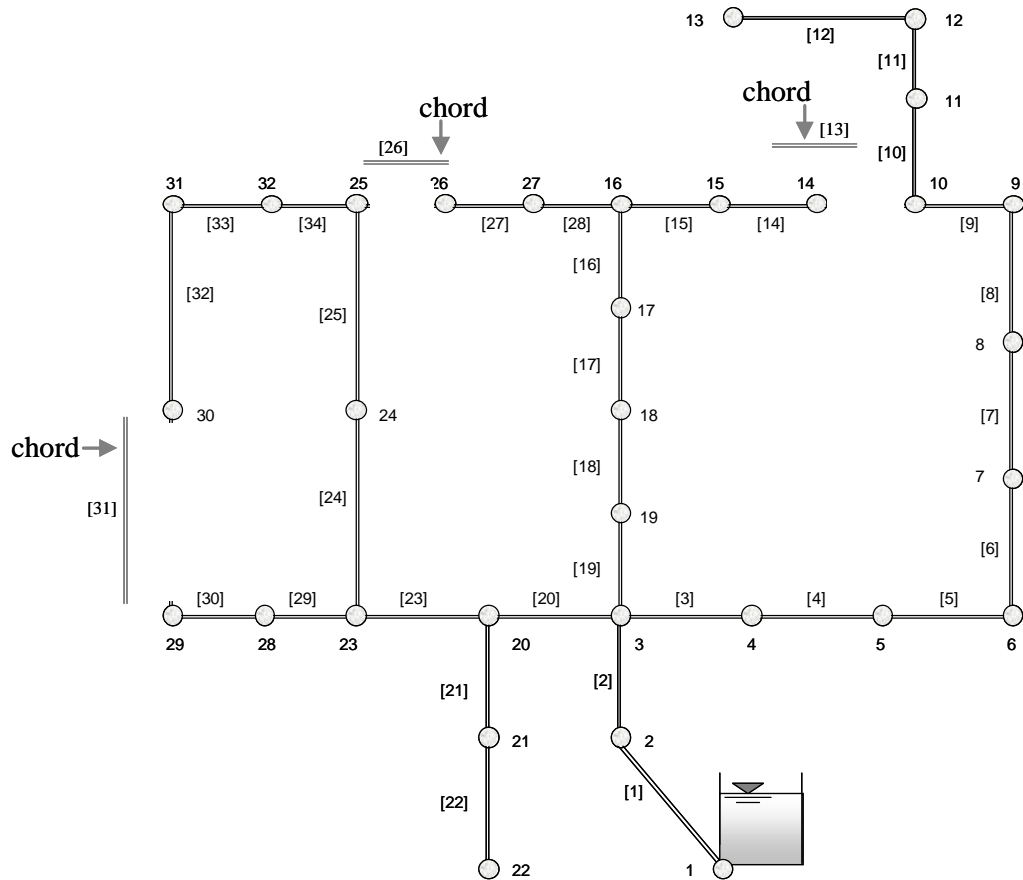


Figure 5 The layout of the shortest-distance tree of the HP network

It is observed from Table 4 that the NLP continuous pipe diameters solution is close to the final best solution design as many continuous pipes diameters fall in the proximity of pipe diameters of the final best solution. Based on the continuous pipe diameters solution achieved in Step 2, the tailored seeding tables that were created for each pipe for NLP-DE1 and NLP-DE2 are given in the fourth and fifth columns of Table 4 respectively.

For the HP case study, the parameters including $N=80$, $MAE=80,000$, $F=0.7$ and $CR=0.8$ were used for NLP-DE1 and NLP-DE2. A total of 100 DE runs with different starting random number seeds have been implemented for each DE application. Table 5 gives the results of the proposed method applied to the HP case study. Results obtained by other optimization techniques for the HP case study are also included in Table 5 to enable performance comparison.

Table 4 Initial seeding tables for the HP case study and the combined NLP-DE results

Links	Flows in the shortest-distance tree (m ³ /s)	Pipe diameters (inches)			Combined NLP-DE final solution (inches)
		The NLP solution produced in Step 2	Initial seeding table for NLP-DE1	Initial seeding table for NLP-DE2	
1	5.54	40.00	30, 40	20, 24, 30, 40	40
2	5.29	40.00	30, 40	20, 24, 30, 40	40
3	1.89	39.81	30, 40	20, 24, 30, 40	40
4	1.86	39.59	30, 40	20, 24, 30, 40	40
5	1.65	38.29	30, 40	20, 24, 30, 40	40
6	1.38	36.29	30, 40	20, 24, 30, 40	40
7	1.00	33.08	30, 40	20, 24, 30, 40	40
8	0.85	31.52	30, 40	20, 24, 30, 40	40
9	0.70	29.84	24, 30	20, 24, 30, 40	40
10	0.56	27.88	24, 30	20, 24, 30, 40	30
11	0.42	25.65	24, 30	20, 24, 30, 40	24
12	0.26	19.28	16, 20	12, 16, 20, 24	24
13 [#]	0.00	12.00	12, 16	12, 16, 20, 24	20
14	0.17	15.70	12, 16	12, 16, 20, 24	16
15	0.25	17.51	16, 20	12, 16, 20, 24	12
16	0.69	26.79	24, 30	20, 24, 30, 40	12
17	0.93	29.22	24, 30	20, 24, 30, 40	16
18	1.30	32.24	30, 40	20, 24, 30, 40	24
19	1.32	32.36	30, 40	20, 24, 30, 40	20
20	1.85	39.11	30, 40	20, 24, 30, 40	40
21	0.39	17.37	16, 20	12, 16, 20, 30	20
22	0.13	12.72	12, 16	12, 16, 20, 30	12
23	1.10	33.08	30, 40	20, 24, 30, 40	40
24	0.63	26.66	24, 30	20, 24, 30, 40	30
25	0.40	23.38	20, 24	16, 20, 24, 30	30
26 [#]	0.00	12.00	12, 16	12, 16, 20, 24	20
27	0.25	18.23	16, 20	12, 16, 20, 24	12
28	0.35	20.15	20, 24	16, 20, 24, 30	12
29	0.18	16.10	16, 20	12, 16, 20, 24	16
30	0.10	13.56	12, 16	12, 16, 20, 24	12
31 [#]	0.00	12.00	12, 16	12, 16, 20, 24	12
32	0.10	15.63	12, 16	12, 16, 20, 24	16
33	0.13	16.84	16, 20	12, 16, 20, 24	16
34	0.35	22.54	20, 24	16, 20, 24, 30	24
Cost (\$M)		5.924	-	-	6.081

[#]=chords of the HP network.

The current best known solution for the HP case study with value of \$6.081 million was first found by *Reca and Martínez* [2006] using a GA variant (GENOME). This solution has been also found by the proposed optimization approach.

Table 5 Algorithm performance for the HP case study

Algorithm	Number of different runs	Best solution found (\$M)	Percentage of trials with best solution found (%)	Average cost solution (\$M)	Worst solution (\$M)	Maximum number of allowable evaluations	Average number of evaluations to find best solutions
NLP-DE1	100	6.081	97	6.082	6.108	80,000	34,609
NLP-DE2	100	6.081	98	6.081	6.100	80,000	42,782
DE-Suribabu ¹	50	6.081	80	-	-	100,000	48,724
GENOME ²	10	6.081	10	6.248	6.450	150,000	-
HD-DDS ³	50	6.081	8	6.252	6.408	100,000	≤100,000
PSO variant ³	2000	6.081	5	6.310	6.550	80,000	-
Standard GA ⁴	30	6.126	0	6.214	6.368	500,000	-
MMAS-ACO ⁵	20	6.134	0	6.394	6.635	100,000	85,571
PSO ⁴	30	6.373	0	6.483	6.801	500,000	-

¹Suribabu [2010]. ²Reca and Martínez [2006]. ³Tolson et al. [2009]. ⁴Dandy et al. [2010]. ⁵Zecchin et al. [2006].

As can be seen from Table 5, NLP-DE1 and NLP-DE2 show a similar performance in finding the best known solution, while NLP-DE1 was found to show slightly better performance than NLP-DE2 in terms of convergence speed. The proposed new optimization models achieved the best performance in terms of percentage of trials with which the best solution was found amongst all the algorithms mentioned in Table 5. As shown in Table 5, NLP-DE1 and NLP-DE2 located the best known solution for the HP case study in 97% and 98% of the optimization trials compared to 80% of DE used in *Suribabu* [2010], 10% of GENOME GA proposed by *Reca and Martínez* [2006], 8% of HD-DDS proposed by *Tolson et al.* [2009] and 5% of PSO variant used in *Tolson et al.* [2009]. The worst solutions produced NLP-DE1 and NLP-DE2 in the 100 different optimization trials were \$6.108 million and \$6.100 million respectively, which deviates only 0.444% and 0.312% from the current best known solution. The standard GA

[Dandy *et al.* 2010], MMAS-ACO [Zecchin *et al.* 2006] and PSO [Dandy *et al.* 2010] were unable to locate the current best solution for the HP case study. The average number of evaluations required by NLP-DE1 and NLP-DE2 were 34,609 and 42,782 respectively, which are less than those reported for any other algorithm.

3.3. Case study 3: ZJ network

The ZJ network, taken from eastern province of China, is an actual water network with a single reservoir. The reservoir has a fixed head of 45 meters. There are 164 pipes, 113 demand nodes and 50 primary loops (as shown in Fig. 6). At each demand node, a minimum pressure of 22 meters is required for the design of this water network. All the pipes are assumed to have an identical Hazen-Williams coefficient of 130. The objective is to determine the least-cost design of this water network, while satisfying the pressure constraints. A total of 14 commercial available pipe diameters ranging from 150 mm up to 1000 mm can be selected for each pipe. Thus, the total search space is $14^{164} \approx 9.2257 \times 10^{187}$.

The shortest-distance tree of the ZJ network determined in Step 1 is shown in Fig. 7. The NLP continuous pipe diameters solution obtained in Step 2, plus the chords with minimum allowable pipe sizes (150 mm for the ZJ network case study) provide an approximately optimal solution with a cost of \$6.970 million. Since this is a new case study that has not been investigated previously, a DE algorithm is applied to optimize this water network directly in order to enable comparison of results.

A total of three DE applications have been performed for the ZJ network optimization including a DE seeded with two tailored pipe diameters for each pipe (NLP-DE1), a DE seeded with four tailored pipe diameters for each pipe (NLP-DE2) and a DE seeded with all 14 available pipe diameters. For each DE application, the parameters including, $N=500$, $MAE=2,000,000$, $F=0.3$ and $CR=0.8$ were used based on a few trials. A total of 10 DE runs with different starting random number seeds have been implemented for each DE application.

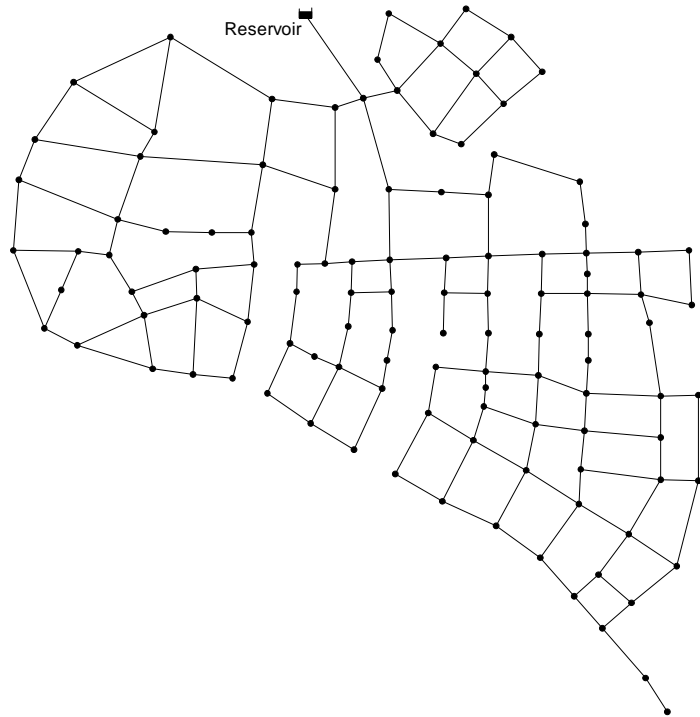


Figure 6 The layout of the ZJ network

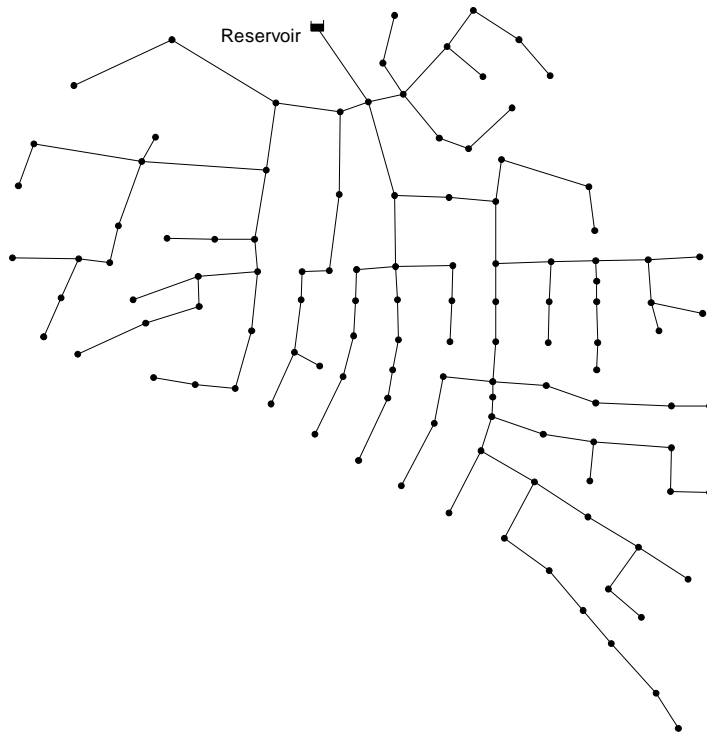


Figure 7 The layout of the shortest-distance tree of the ZJ network

The solutions obtained by the three DE variants and statistics of the results are given in Fig. 8 and Table 6 respectively. It is clearly seen from Fig. 8 that NLP-DE1 converged the fastest and the DE3 converged the slowest. Although NLP-DE2 converged slower than NLP-DE1, NLP-DE2 was able to produce lower cost solutions. It is noted that the solutions obtained by NLP-DE1 and NLP-DE2 are less scattered by those found by DE3. This shows that the solutions of NLP-DE1 and NLP-DE2 are less dependent on the starting random number seeds.

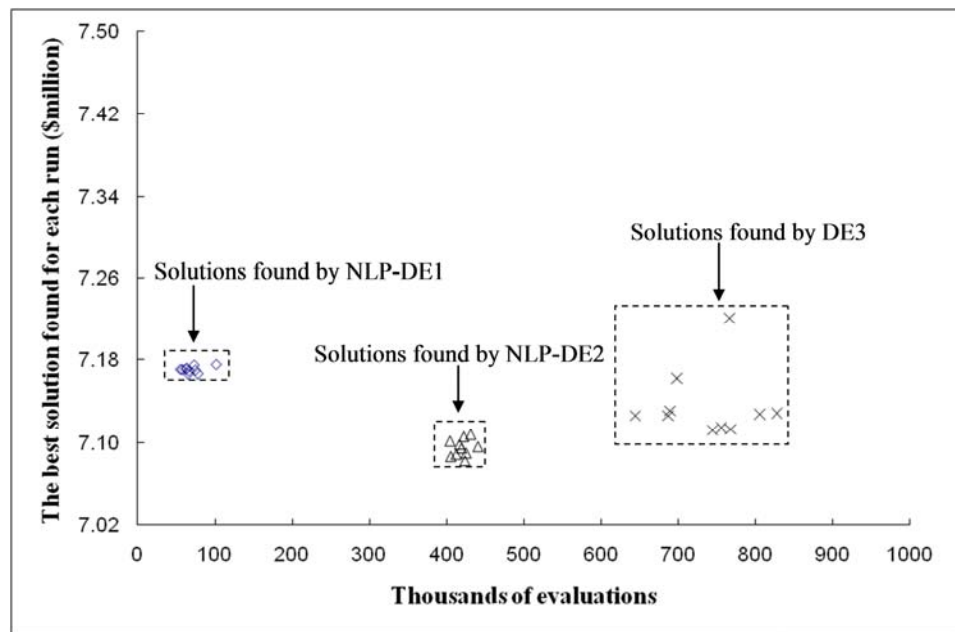


Figure 8 Solution distributions for the ZJ case study

Table 6 Algorithm performance for the ZJ case study

Algorithm	Number of different runs	Best solution found (\$M)	Percentage of trials with best solution found (%)	Average cost solution (\$M)	Worst solution found (\$M)	Maximum number of allowable evaluations	Average number of evaluations to find best solutions
NLP-DE1	10	7.167	0	7.170	7.175	2,000,000	69,300
NLPDE2	10	7.082 ¹	10	7.093	7.105	2,000,000	400,853
DE3	10	7.112	0	7.136	7.220	2,000,000	820,657

¹The current best solution for the ZJ case study.

As observed from Table 6, different NLP-DE1 runs are significantly more computationally efficient than DE3. This is evidenced by the fact that NLP-DE1 only

required a total of 8.44% of the computation overhead required by that of DE3. This shows that the DE seeded with 2 tailored pipe diameters derived from the approximately optimal solution obtained from NLP in Step 2 is able to find optimal solutions with significantly enhanced computational efficiency.

As can be seen from Table 6, the NLP-DE2 found the current best solution for this case study with a cost of \$7.082 million, 0.42% cheaper than the best solution found by DE3. Additionally, the worst solution found by the 10 NLP-DE2 runs was lower than the best solution found by DE3. It is noted that NLP-DE2 converged quicker than DE3 as the average number of evaluations required to converge by 10 different NLP-DE2 runs is only 48.84% of that required by DE3.

3.4. Case study 4: Balerma network (BN)

The Balerma network (BN), an irrigation water distribution network located in the province of Almeria (Spain), was first investigated by *Reca and Martínez* [2006]. It consists of 4 reservoirs, 8 loops, 454 pipes and 443 demand nodes as shown in Fig. 9. A total of 10 PVC commercial pipes with nominal diameters from 125 mm to 600 mm are to be selected for this network. Thus, the search space is 10^{454} , which is significantly larger than the previous three case studies in this paper. All the pipes are assumed to have an absolute roughness height $k=0.0025$ mm and the minimum required pressure at each node is 20 meters. Pipe costs are given in *Reca and Martínez* [2006].

Since there are four reservoirs involved in the Balerma network, the proposed extension to the Dijkstra algorithm described in section 2.1.2 is employed to find the shortest-distance tree for this multi-source WDS. The shortest-distance tree for the Balerma network identified, based on the proposed extension of the Dijkstra algorithm, is given in Fig. 10. It is seen from Fig. 10, the original Balerma network has been decomposed into four sub-networks connected via an artificial node and four artificial links. An NLP is formulated for this tree network and solved in Step 2, producing an approximately optimal solution with a cost of € 2.114 million (all the chords are assumed to be the

smallest pipe size). Note that the artificial node and artificial links are not included in the NLP.

For the BN case study, like the ZJ case study, a total of three DE applications are carried out. These include a DE seeded with two tailored pipe diameters for each pipe (NLP-DE1), a DE seeded with four tailored pipe diameters for each pipe (NLP-DE2) and a DE seeded with all 10 available pipe diameters. For each DE application, the parameters used were $N=500$, $MAE=10,000,000$, $F=0.3$ and $CR=0.8$. A total of 10 DE runs with different starting random number seeds have been implemented for each DE application. The solution distribution and a summary of results are given in Fig. 11 and Table 7 respectively.



Figure 9 The layout of the Balerma network



Figure 10 The layout of the shortest-distance tree of the Balerna network

As can be seen from Fig. 11, the NLP-DE1 and NLP-DE2 runs located overall lower cost solutions for the BN case study compared to the DE3 runs with significantly less computational effort. NLP-DE2 converged slightly slower than NLP-DE1, while being able to find better quality solutions as shown in Fig. 11. It is seen from Table 7, the average number of evaluations required to find the better quality solutions for NLP-DE1 and NLP-DE2 are only 4.47% and 15.50% of that required by DE3. This shows that a DE with initial estimates provided by an NLP run in the proposed optimization approach is able to locate better quality solutions with a significantly faster convergence speed than a DE without initial estimates. In addition, the solutions produced by NLP-DE1 and NLP-DE2 are less scattered than those of DE3 for the BN case study. The NLP-DE2

produced a new currently lowest cost solution with a value of € 1.923 million for the BN case study.

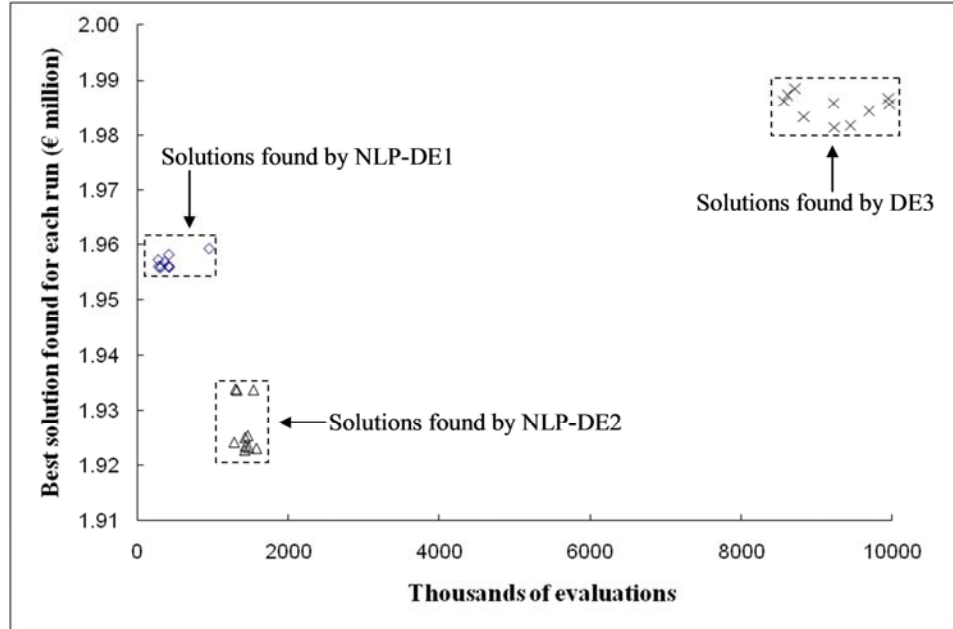


Figure 11 Solution distributions for the BN case study

Table 7 Algorithm performance for the BN case study

Algorithm	Number of different runs	Best solution found (\$M)	Percentage of trials with best solution found (%)	Average cost solution (\$M)	Worst solution found (\$M)	Maximum number of allowable evaluations	Average number of evaluations to find best solutions
NLP-DE1	10	1.956	0	1.957	1.959	1,000,000	412,000
NLP-DE2	10	1.923 ⁴	10	1.927	1.934	2,000,000	1,427,850
DE3	10	1.982	0	1.986	1.989	10,000,000	9,210,143
HD-DDS-1 ¹	1	1.941	0	-	-	30,000,000	-
HD-DDS-2 ¹	10	1.956	0	-	-	10,000,000	-
GHEST ²	-	2.002	-	2.055	-	10,000,000	254,400
GENOME GA ³	10	2.302	0	2.334	2.350	10,000,000	-

¹Tolson et al. [2009]. ²Reca and Martínez [2006]. ³Bolognesi et al. [2010]. ⁴A new current best solution for the BN case study. HD-DDS-1 and HD-DDS-2 are HD-DDS approach with maximum number of allowable evaluations of 30,000,000 and 10,000,000 respectively.

On comparing the algorithmic performance with other optimization techniques, the proposed new optimization approach (NLP-DE1 and NLP-DE2) performed the best in terms of quality of the best solution found and efficiency as shown in Table 7. NLP-DE1

found the same best cost solution with a value of € 1.956 as that found by HD-DDS [Tolson *et al.* 2009] in a total of 10 different runs. However, it is noted that the computational budget for the NLP-DE1 was only 10% of that for HD-DDS-2 [Tolson *et al.* 2009]. The HD-DDS-1 [Tolson *et al.* 2009] found the previous best solution with a value of €1.941 million using 30 million evaluations, while the NLP-DE2 located a new lower cost solution with a cost of €1.923 million using only 2 million evaluations (6.67% of the computational budget required by HD-DDS-1). In addition, the worst solution produced by 10 different NLP-DE2 runs was € 1.934 million, which is still lower than the best solution found by HD-DDS [Tolson *et al.* 2009], GHEST [Bolognesi *et al.* 2010] and GENOME GA [Reca and Martínez 2006]. This implies that the proposed optimization approach is able to locate better quality solutions with significantly improved computational efficiency when dealing with such large scale water networks.

3.5. Summary of results

It has been shown that the new proposed NLP-DE algorithm has outperformed all the other optimization algorithms in terms of efficiently finding optimal solutions for the four case studies. The dominance of the proposed method is more clearly shown for the larger networks including the ZJ and BN case studies. In terms of solution quality, NLP-DE1 yielded a similar performance to NLP-DE2 for relatively small water networks (such as the NYTP and HP case studies). However, for relatively larger water networks (such as the ZJ and BN case studies), NLP-DE1 was able to converge faster than NLP-DE2 while NLP-DE2 found lower cost solutions than NLP-DE1. This is explained by the fact that NLP-DE2 was seeded with an initial seeding table with four different pipe diameters, while NLP-DE1 was initialized with a seeding table consisting of only two different pipe diameters. Consequently, NLP-DE2 explored a relatively larger search space than NLP-DE1 and hence resulted in a greater search time but with better quality solutions being found. Based on the observation of this study, a DE seeded with two tailored pipe diameters based on the NLP solution (NLP-DE1) is recommended for relatively small water network optimization. For relatively large WDS case studies, DE seeded with four tailored pipe diameters based on the NLP solution (NLP-DE2) is

recommended. For a WDS case study with a larger number of pipes and loops, the continuous diameter solution obtained in Step 2 with the NLP may be more of an approximation to the actually global optimum as more pipes are removed as chords. In such a case, the initial seeding table based on this continuous diameter solution can be further increased in size (for example-to say six successive pipes diameters to be included in the seeding table for each pipe). As a result, this should lead to a more effective seeding of the DE exploration.

It has also been found from this study that, for the optimization problem of designing a completely new WDS, the flows in the shortest-distance tree determined by the assumption that there is no discharge in the chords (Method 1 in section 2.2) is effective. For the expansion of an existing WDS optimization problem (such as the NYTP case study), the flows in the shortest-distance tree determined by the assumption that the discharges in the chords are set to be equal to that from the hydraulic calculation for the original existing WDS (Method 2 in section 2.2) is the most effective.

In the proposed method, the shortest path is used as a surrogate indicator of the main flow paths within the network (the network tree). It is considered that the accuracy of this assumption will be reduced in situations where there are significant differences in nodal elevations. However, the NLP solution based on the assumed tree is simply used to identify an initial seeding table for seeding the DE optimization. Minor changes in the NLP solution may not necessarily vary the initial seeding table components as the two or four adjacent discrete pipe sizes based on the continuous diameter pipe solution from the NLP are included in the seeding table for each pipe. In addition, our experiments have shown that a moderate change to the initial seeding table components does not influence the performance of DE significantly as the DE is able to progress the search outside of the bounds of the seeding table. The BN case study involved in this study is a network having significant nodal elevation differences, however, the proposed method was observed to exhibit satisfactory performance on this case study in terms of solution quality and efficiency (See Table 7).

An analysis of the computational effort required in Step 1 and Step 2 in the proposed optimization approach has been undertaken. The computational time required to find the shortest-distance tree and to run the NLP solver for each case study is converted to an equivalent number of case study evaluations respectively. Note all these tests were performed in the same computer (Pentium PC at 3.0 GHz). The results are given in Table 8. It can be seen from Table 8 that the computational effort required to find the shortest-distance tree in Step 1 and to run the NLP solver in Step 2 is negligible compared to that required in Step 3. Thus, the computational effort in running the Dijkstra algorithm and NLP for each case study has not been included in the total computational overhead. For example, the computational overhead of running the Dijkstra algorithm and NLP for the ZJ case study is only 0.19%, 0.033% and 0.014% of that required by the NLP-DE1, NLP-DE2 and DE3 respectively. This implies that it is computationally efficient to find the shortest-distance tree and solve the NLP for the shortest-distance tree for a given WDS. This further improves the attractiveness of the proposed approach for optimization of WDSs.

Table 8 Computational effort analysis for finding shortest-distance tree and running the NLP solver for each case study

Case study	Number of decision variables	Computational effort required to find the shortest-distance tree (Step 1)	Computational effort required to solve the NLP for the shortest-distance tree (Step 2)	Average number of evaluations required by NLP-DE1 (Step 3)	Average number of evaluations required by NLP-DE2 (Step 3)
NYTP	21	11	10	8,277	10,631
HP	34	10	26	34,609	42,782
ZJ	164	6	125	69,300	400,853
BN	454	8	2,133	412,000	1,427,850

Note: The computational effort in Steps 1 and 2 has been converted to an equivalent number of evaluations for its corresponding case study. One simulation for the NYTP, HP, ZJ and BN case study on Pentium PC at 3.0GHz was 0.001, 0.001, 0.016 and 0.015 seconds respectively.

4. Conclusions

A new optimization approach aimed at optimizing the design of WDSs has been presented in this paper. This new approach divides the optimization process into three steps. These include:

1. Find the shortest-distance tree for the looped WDS that is being optimized.
2. Carry out an NLP optimization of the shortest-distance tree.
3. Optimize the original water network using the DE seeded based on pipe sizes in the proximity of those found in Step 2.

The shortest-distance tree is identified in Step 1 using the Dijkstra algorithm. The shortest-distance tree is viewed as an optimal tree based on the assumption that delivering demand along the shortest path for each node is the most effective mode. A NLP is then formulated for optimizing the design of the shortest-distance tree in Step 2 and has been solved by an NLP solver in this study. For each case study, a range of different initial starting points have been used for solving the NLP applied to the optimization of the shortest-distance tree. It was found that the final solution is identical for all the different initial starting points. The continuous pipe diameter solution produced in Step 2 complemented by the shortest-distance tree chords with the minimum allowable pipe sizes are used to create the initial seeding tables for the differential evolution (DE) optimization process. The DE optimization for finding the optimal discrete pipe size solution in Step 3 is seeded with the tailored pipe diameters seeding tables created in Step 2.

Results for four cases studies show that the proposed new combined NLP-DE optimization approach has superior convergence properties. For the NYTP and HP case studies, the proposed optimization technique reached the current best known solution for each network more frequently and more efficiently compared with other optimization techniques. For the ZJ and BN case studies, the proposed new optimization approach found the new lowest cost solutions with a cost of \$7.082 million and € 1.923 million respectively. In addition, the new method produced optimal solutions with an extremely fast convergence speed. The consistent superior performance of the proposed optimization approach on four case studies illustrates that the proposed methodology is well suited for the least-cost design of WDSs.

The utility of the proposed method is that it provides an efficient and effective approach for seeding the optimization of the full combinatorial problem using near optimal solutions (achieved by solving an approximated continuous problem with NLP). A natural extension of this method to find an approximate Pareto front for multi-objective problems (to seed a full multi-objective combinatorial search) could be achieved by incorporating one of the many approaches to map multi-objective problems to a series of single objective problems [Konak *et al.* 2006]. This approximate front would then be used to seed a multi-objective combinatorial optimizer (i.e. NSGA2: Deb *et al.* [2002]) to determine the actual front. This extension should be the focus of future research. Another issue that needs to be addressed is that the decision variables in this paper are only pipe diameters for the case studies, whereas the real WDS design problems may be more complex. Since the proposed methodology has shown to be effective for the pipes-only WDS design problems, future work should be focus on applying the proposed methodology to deal with the real-world WDS problems that may include pumps, valves, storage facilities and pipes.

References

- Alperovits, E., and U. Shamir (1977), Design of water distribution systems, *Water Resour. Res.*, 13(6), 885-900, doi:10.1029/WR013i006p00885.
- Afshar, A., A. Zahraei, and M. A. Mariño (2009), Large-scale nonlinear conjunctive use optimization problem: Decomposition algorithm, *J. Water Resources Planning and Management*, 136(1), 59-71, doi:10.1061/(ASCE)0733-9496(2010)136:1(59).
- Bolognesi, A., C. Bragalli, A. Marchi, and S. Artina (2010), Genetic heritage evolution by stochastic transmission in the optimal design of water distribution networks, *Advances in Engineering Software*, 41(5), 792-801, doi:10.1016/j.advengsoft.2009.12.020.
- Bhave, P. R., and V. V. Sonak (1992), A critical study of the linear programming gradient method of optimal design of water supply networks, *Water Resour. Res.*, 28(6), 1577-1584, doi:10.1029/92wr00555.
- Cunha, M. C., and J. Sousa (2001), Hydraulic Infrastructures Design Using Simulated Annealing, *Journal of Infrastructure Systems*, 7(1), 32-38, doi:10.1061/(ASCE)1076-0342(2001)7:1(32).

- Dandy, G. C., A. R. Simpson, and L. J. Murphy (1996), An improved genetic algorithm for pipe network optimization, *Water Resour. Res.*, 32(2), 449-457, doi:10.1029/95wr02917.
- Dandy, G. C., A. Wilkins, and H. Rohrlach (2010), A methodology for comparing evolutionary algorithms for optimizing water distribution systems, *Proceedings of the 12th Water Distribution System Analysis Symposium*, Tucson, Arizona.
- Davidson, J., B. Francois, C. Stephen, and J. Paul (2005), Real-time connectivity modeling of water distribution networks to predict contamination spread, *J. Comput. Civ. Eng.*, 19(4), 377-386, doi:10.1061/(ASCE)0887-3801(2005)19:4(377).
- Deb, K. (2000), An efficient constraint handling method for genetic algorithms, *J. Comput. Methods Appl. Mech. Engineering*, 186,311-338, doi:10.1016/s0045-7825(99)00389-8.
- Deb, K., A. Pratap, S. Agawal and T. Meyarivan (2002), A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Trans Evol. Comput.*, 6(2), 182-197.
- Deo, N (1974). Graph theory with applications to engineering and computer science. Prentice-Hall, Englewood Cliffs, N. J.
- Deuerlein, J. W. (2008), Decomposition model of a general water supply network graph, *Journal of Hydraulic Engineering*, 134(6), 822-832.
- Eusuff, M. M., and K. E. Lansey (2003), Optimization of water distribution network design using shuffled frog leaping algorithm, *J. Water Resources Planning and Management*, 129(3), 210-225.
- Fujiwara, O., B. Jenchaimahakoon, and N. C. P. Edirisinghe (1987), A modified linear programming gradient method for optimal design of looped water distribution networks, *Water Resour. Res.*, 23(6), 977-982.
- Fujiwara, O., and D. B. Khang (1990), A two-phase decomposition method for optimal design of looped water distribution networks, *Water Resour. Res.*, 26(4), 539-549, doi: 10.1029/WR026i004p00539.
- Geem, Z. W., J. H. Kim, and G. V. Loganathan (2002), Harmony search optimization: application to pipe network design, *Int. J. Model. Simulat.*, 22(2), 125-133.
- Kim, J. H., T. G. Kim, J. H. Kim, and Y. N. Yoon (1994), A study on the pipe network system design using non-linear programming, *Journal of Korean Water Resources Association*, 27(4), 59-67.

- Kadu, M. S., G. Rajesh, and P. R. Bhawe (2008), Optimal design of water networks using a modified genetic algorithm with reduction in the search space, *J. Water Resources Planning and Management*, 134(2), 147-160, doi:10.1061/(ASCE)0733-9496(2008)134:2(147).
- Konak, A., D. W. Coit, and A. E. Smith (2006), Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, 91, 992-1007, doi:10.1016/j.res.2005.11.018.
- Krapivka, A., and A. Ostfeld (2009), Coupled genetic algorithm-linear programming scheme for least-cost pipe sizing of water-distribution systems, *J. Water Resources Planning and Management*, 135(4), 298-302, doi:10.1061/(ASCE)0733-9496(2009)135:4(298).
- Lansley K. E., and L. W. Mays (1989), Optimization model for water distribution system design, *J. Hydraul. Eng.*, 115(10), 1401-1417, doi:10.1061/(ASCE)0733-9429(1989)115:10(1401).
- LINDO Systems Inc., (2009), LINGO12 User's Guide, *LINDO Systems Inc.*, Chicago.
- Maier, H. R., A. R. Simpson, A. C. Zecchin, W. F. Foong, K. Y. Phang, H. Y. Seah, and C. L. Tan (2003), Ant colony optimization for the design of water distribution systems, *J. Water Resources Planning and Management*, 129(3), 200-209, doi:10.1061/(ASCE)0733-9496(2003)129:3(200).
- Montesinos, P., A. García, and J. L. Ayuso (1999), Water distribution network optimization using modified genetic algorithm, *Water Resour. Res.*, 35(11), 3467-3473, doi: 10.1029/1999wr900167.
- Murphy, L. J., and A. R. Simpson (1992). Pipe optimization using genetic algorithms, *Research Report No. R93*, Department of Civil Engineering, University of Adelaide.
- Reca, J., and J. Martínez (2006), Genetic algorithms for the design of looped irrigation water distribution networks, *Water Resour. Res.*, 42, W05416, doi:10.1029/2005WR004383.
- Reis, L. F. R., F. T. Bessler, G. A. Waters, and D. Savic (2006), Water supply reservoir operation by combined genetic algorithm-linear programming (GA-LP) approach, *Water Resources Management*, 20: 227-255, doi:10.1007/s11269-006-8049-z.
- Rossman, L. A. (2000), *EPANET 2-User manual*, National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency, Cincinnati, OH.

- Savic, D. A., and G. A. Waters (1997), Genetic algorithms for least-cost design of water distribution networks, *J. Water Resource Planning and Management*, 123(2), 67-77, doi: 10.1061/(ASCE)0733-9496(1997)123:2(67).
- Shinstine, D. S., I. Ahmed, and K. E. Lansey (2002), Reliability/availability analysis of municipal water distribution networks: Case studies, *J. Water Resources Planning and Management*, 128(2), 140-151, doi: 10.1061/(ASCE)0733-9496(2002)128:2(140).
- Simpson, A. R., G. C. Dandy, and L. J. Murphy (1994), Genetic algorithms compared to other techniques for pipe optimization, *J. Water Resources Planning and Management*, 120(4), 423-443, doi: 10.1061/(ASCE)0733-9496(1994)120:4(423).
- Sonak, V. V., and P. R. Bhave (1993), Global optimum tree solution for single-source looped water distribution networks subjected to a single loading pattern, *Water Resour. Res.*, 29(7), 2437-2443, doi: 10.1029/93WR00289.
- Storn, R., and K. Price (1995), Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous space, *Technical Report*, International Computer Science Institute, Berkeley, CA.
- Suribabu, C. R., and T. R. Neelakantan (2006), Design of water distribution networks using particle swarm optimization, *Urban Water J.*, 3(2), 111-120, doi:10.1080/15730620600855928.
- Suribabu, C. R. (2010), Differential evolution algorithm for optimal design of water distribution networks, *Journal of Hydroinformatics*, 12 (1), 66-82.
- Tolson, B. A., M. Asadzadeh, H. R. Maier, and A. C. Zecchin (2009), Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization, *Water Resour. Res.*, 45, W12416, doi:10.1029/2008WR007673.
- Vasan, A., and S. P. Simonovic (2010), Optimization of water distribution network design using differential evolution, *J. Water Resource Planning and Management*, 136(2), 279-287..
- Walters, G. A., and T. K. Lohbeck (1993), Optimal layout of tree networks using genetic algorithms, *Engineering Optimization*, 22, 27-48.
- Waters, G. A., and D. K. Smith (1995), Evolutionary design algorithm for optimal layout of tree networks, *Engineering Optimization*, 24, 261-281.
- Yang, S., P. W. F. Louie, and W. W.-G. Yeh (1996), Water distribution network reliability: Connectivity analysis, *J. Infrastruct. Syst.*, 2(2), 54-64.

Zecchin, A. C., A. R. Simpson, and H. R. Maier (2006), Application of two ant colony optimization algorithms to water distribution system optimization.” *J. Mathematical and Computer Modeling*, 44,451-668.

Zheng, F., A. R. Simpson, and A. C. Zecchin (2010), A method of assessing the performance of genetic algorithm optimization water distribution design, *Proceedings of the 12th Water Distribution System Analysis Symposium*, Tucson, Arizona.

Chapter 7. Journal Paper 5-Coupled BLP-DE Method for WDS Optimization

7.1 Synopsis

A coupled binary linear programming-differential evolution algorithm approach for optimizing water distribution system optimization

In Chapter 6, a combined NLP-DE method is presented, in which NLP is employed to optimize the entire water network in order to obtain an approximate optimal solution. Then this approximate optimal solution is used to initialize the DE search. As such, the DE focuses on exploring only the promising regions specified by the approximate optimal solution obtained by NLP optimization rather than the original whole search space. As a result, better quality solutions for the water network are reached more efficiently, and with a higher likelihood.

In the NLP-DE method presented in Chapter 6, the deterministic optimization technique NLP is used to provide a good estimate for the DE exploration. Another novel hybrid optimization technique is developed in this Chapter based on a binary linear programming coupled with a DE (BLP-DE) for the water network optimization. This proposed optimization technique is presented in this Chapter (Chapter 7).

Three stages are involved in the proposed BLP-DE optimization method. In the first stage, the WDS that is being optimized is decomposed into trees and the core using a graph algorithm. BLP is then used to optimize the design of the trees during the second stage. In the third stage, a DE algorithm is utilized to deal with the core design while incorporating the optimal solutions for the trees obtained in the second stage, thereby yielding optimal solutions for the original whole WDS. The proposed method takes advantage of both BLP and DE algorithms: BLP is capable of providing a global optimal solution for the trees (no loops involved) with great efficiency, while a DE is able to efficiently generate good quality solutions for the core (where loops are involved) with a reduced search space compared to the original full network.

Three main differences exist when comparing the BLP-DE method outlined in this Chapter with the NLP-DE approach presented in Chapter 6, which are (i) for the BLP-DE method, the deterministic BLP method is used to optimize the design for the trees of the water network rather than providing the estimates for the evolutionary algorithms as for the NLP-DE method; (ii) BLP is able to provide discrete diameter solutions while the NLP only generated continuous diameter solutions and (iii) the DE in the BLP-DE is only utilized to deal with the core of the original whole network, in contrast, the DE in the NLP-DE is used to tackle the whole network.

Another novelty of the proposed BLP-DE method is that a solution choice table method has been proposed to incorporate the optimal solutions for the trees when the core of the water network is being optimized. As such, the final optimal solutions obtained by the DE applied to the core are actually the optimal solutions for the original entire network.

In the proposed BLP-DE method, different components of the whole water network are optimized by different optimization techniques, in which the trees are optimized by BLP and the core is handled by the DE. This proposed approach makes good use of the advantages of both the deterministic optimization techniques and the evolutionary algorithms, i.e., deterministic optimization techniques are suitable for the tree optimization as no loops are involved and the evolutionary algorithms are able to efficiently explore the relatively small search space effectively (only the core rather than the whole network is optimized by the DE in the proposed BLP-DE method).

This work has been submitted to *Journal of Water Resources Planning and Management* and the paper is provided here.

Citation of Paper

Zheng, F., A. R. Simpson, and A. C. Zecchin (2012c). “A combined binary linear programming and differential evolution algorithm approach for water distribution system optimization.” *Journal of Water Resources Planning and Management*, submitted June 2012.

STATEMENT OF AUTHORSHIP (SUBMITTED)

Journal paper title: **A coupled binary linear programming-differential evolution algorithm approach for water distribution system optimization.**

Authors: Feifei Zheng¹, Angus Simpson² and Aaron Zecchin³

Corresponding author:

¹Feifei Zheng (Candidate): PhD student, School of Civil, Environmental and Mining Engineering, University of Adelaide

Wrote the manuscript, performed all analysis, developed the model and theory and acted as corresponding author.

I hereby certify that the statement of contribution is accurate.

Signed _____ Date _____

²Angus Simpson: Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide

Reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

³Aaron Zecchin: Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide

Reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

Submitted to Journal of Water Resources Planning and Management, 1 June 2012

7.2 Journal Paper 5: A coupled binary linear programming and differential evolution approach for water distribution system optimization (Submitted to the *Journal of Water Resources Planning and Management*)

Feifei Zheng, Angus R. Simpson and Aaron C. Zecchin

ABSTRACT

A coupled binary linear programming-differential evolution (BLP-DE) approach is proposed in this paper to optimize the design of water distribution systems (WDSs). Three stages are involved in the proposed BLP-DE optimization method. In the first stage, the WDS that is being optimized is decomposed into trees and the core using a graph algorithm. Binary linear programming (BLP) is then used to optimize the design of the trees during the second stage. In the third stage, a differential evolution (DE) algorithm is utilized to deal with the core design while incorporating the optimal solutions for the trees obtained in the second stage, thereby yielding optimal solutions for the original whole WDS. The proposed method takes advantage of both BLP and DE algorithms: BLP is capable of providing global optimal solution for the trees (no loops involved) with great efficiency, while a DE is able to efficiently generate good quality solutions for the core (loops involved) with a reduced search space compared to the original full network. Two benchmark WDS case studies and one real-world case study (with multiple demand loading cases) with a number of decision variables ranging from 21 to 96 are used to verify the effectiveness of the proposed BLP-DE optimization approach. Results show that the proposed BLP-DE algorithm significantly outperforms other optimization algorithms in terms of both solution quality and efficiency

INTRODUCTION

A number of deterministic optimization techniques have previously been applied to the optimization design problem of water distribution systems (WDSs). These include a complete enumeration approach (Gessler 1985); linear programming (LP) (Alperovits and Shamir 1977; Sonak and Bhave 1993; Guercio and Xu 1997); and

non-linear programming (NLP) (Lansey and Mays 1989; Fujiwara and Khang 1990). Each deterministic method offers advantages, but also has disadvantages in terms of optimizing WDS design. The complete enumeration approach guarantees that the global optimal solution will be found, for example, but the computational overhead is extremely intensive as this method evaluates every possible combination of discrete pipe diameters for a WDS. In most cases, this is an impossible task. LP and NLP converge quickly, on the other hand, but only a local optimum is located. In addition, split pipe sizes are usually allowed by a LP solution and continuous pipe diameters are normally included in a NLP solution, neither of which is practical from an engineering perspective.

Samani and Mottaghi (2006) proposed a binary linear programming (BLP) approach for WDS design optimization, in which the objective function and constraints were linearized using zero-one variables. Four steps are involved for their BLP method, which are step 1: each pipe in the water network to be optimized is initially assigned a commercially available pipe diameter; step 2: a hydraulic solver is performed for the known network configuration to obtain water flows for each pipe; step 3: a BLP model is formulated and solved for the water network based on the known flows at each pipe and solved while satisfying the head constraints at each node and step 4: the resulting pipe sizes obtained in step 3 are compared with the assumed pipe diameters in step 1. If they are the same, the optimization process has converged and the resulting pipe sizes are the final solution, otherwise, the resulting pipe sizes are assigned to the water network and steps 2, 3 and 4 are repeatedly performed until the convergence (where resulting pipe sizes in step 3 are the same as the those used in step 2) is achieved.

Samani and Mottaghi (2006) used two relatively small WDS case studies to verify the effectiveness of their proposed BLP method, and reported that the performance of the BLP method was satisfactory in terms of accuracy and convergence based on results of two WDS case studies.

The advantage of the BLP developed by Samani and Mottaghi (2006) over LP and NLP is that it is able to handle the discrete search space, thereby providing discrete pipe diameter solutions. However, the BLP approach is compromised by extreme inefficiency when dealing with relatively large WDS case studies (Savic and Cunha 2008). In addition, the global optimum for a looped WDS cannot be guaranteed as the final solution reached by this BLP approach is dependent on the initially assumed pipe diameters (Martínez 2008).

In addition to deterministic optimization techniques (LP, NLP and BLP), evolutionary algorithms (EAs), as stochastic approaches, have also been employed to optimize the design for WDSs. Simpson et al. (1994) first applied a genetic algorithm (GA) to tackle the water network optimization problem. Afterwards, a number of other evolutionary algorithms were developed and applied to WDS design. These include simulated annealing (Cunha and Sousa 2001); harmony search (Geem et al. 2002); the shuffled frog leaping algorithm (Eusuff and Lansey 2003); Ant Colony Optimization (Maier et al. 2003); the modified GA (Vairavamoorthy and Ali 2005); particle swarm optimization (Suribabu and Neelakantan 2006); cross entropy (Perelman and Ostfeld, 2007); scatter search (Lin et al. 2007); HD-DDS (Tolson et al 2009) and differential evolution (Suribabu 2010). These EAs have been applied to a number of WDS case studies and exhibit good performance in terms of finding optimal solutions.

The advantages of EAs over deterministic optimization methods are (i) EAs are able to deal with the discrete search space directly and (ii) EAs explore the search space broadly and are therefore more likely to provide good quality solutions. However, a major issue pertaining to the application of EAs for WDS design is the computational intensity. This is a severe limitation for EAs dealing with real-world WDS optimization, for which, a large number of pipes are normally involved. Zheng et al. (2011a) reported that EAs perform well on relatively small case studies in terms of solution quality, whereas solution quality deteriorates for EAs when dealing with

larger networks. Thus, it is desirable to develop advanced optimization techniques to overcome these limitations to enable a more generic application of optimization techniques for WDS design. The development of hybrid optimization methods is a way of overcoming the problems outlined above.

Recently, EAs have been combined with deterministic optimization methods in attempts to overcome the disadvantages of both optimization techniques when optimizing the design of WDSs. Krapivka and Ostfeld (2009), for example, proposed a coupled GA-LP method for the WDS optimization. In their technique, all possible spanning trees for a looped water network are first evaluated to identify the least-cost spanning tree. The chords of the tree are assigned with the minimum allowable pipe diameters. The coupled GA-LP technique is then used to further polish the optimal solution of the least-cost spanning tree, in which a GA is used to update the flow distribution, while LP is employed to optimize the tree for a given flow distribution. However, this GA-LP method is limited by the fact that split pipe sizes are allowed in the final solution and it is computationally expensive to evaluate all possible spanning trees for a large WDS.

Tolson et al. (2009) developed a hybrid discrete dynamically dimensioned search algorithm (HD-DDS) for WDS design optimization. In the HD-DDS, a stochastic algorithm is combined with two local search methods (one-pipe search and two pipes search algorithms). These two local search methods are carried out using complete enumeration. Efficiency improvements were reported by Tolson et al. (2009) when this method was compared to other optimization algorithms in terms of optimizing WDSs.

Zheng et al. (2011b) developed a combined NLP-DE approach to deal with WDS optimization problems. In the NLP-DE method, graph decomposition is first employed to identify the shortest-distance tree for a looped WDS. NLP is then used to optimize the shortest-distance tree and an approximately optimal solution is obtained for the original full network. Finally, a DE is seeded in the vicinity of the

approximately optimal solution rather than the whole search space in order to optimize the original full network. It was reported by Zheng et al. (2011b) that the combined NLP-DE method was able to find good quality solutions for the WDSs with great efficiency based on four case studies.

Haghighi et al. (2011) combined a simple GA with BLP for WDS optimization design. In this GA-BLP method, a water network is first converted to a tree by removing one pipe from each primary loop and hence a total of NL pipes are removed, where NL is the number of loops in the water network. Then a set of N diameter combinations for the NL pipes are randomly generated using commercially available pipe diameters to form the initial population of the GA, where N is the population size of the GA. For each GA individual with different diameter combinations for the NL pipes, an iterative procedure using BLP combined with a hydraulic solver (EAPNET) is utilized to optimize the remaining tree (the NL pipes are not included in the BLP optimization).

The optimum pipe diameters obtained from the iterative BLP optimization for the tree are returned to the GA along with corresponding cost. This cost in combination with the cost of the NL pipes handled by the GA provides the total cost of the original water network. This total cost is used to calculate the fitness of the GA individual. Subsequently, the GA operators (selection, crossover and mutation) are performed to evolve the initial solutions to achieve the final optimal solutions.

In the GA-BLP method (Haghighi et al. 2011), the GA was only used to deal with the NL pipes, while BLP was employed to tackle the optimization of the tree that was obtained by removing NL pipes. Thus, efficiency of the GA optimization is expected to be improved as the GA only handles the NL pipes rather than the total number pipes in the original whole network (NL is normally significantly smaller than the total number pipes). However, the computational effort for iterative BLP optimization in this GA-BLP approach is massive when dealing with large water

networks since BLP has been found to be extremely inefficient when tackling large optimization problems (Savic and Cunha 2008; Martínez 2008).

In the current paper, a novel hybrid optimization approach that combines BLP and DE is proposed for optimizing the design of WDSs. Three stages are proposed in the BLP-DE method. In the first stage, the full water network that is being optimized is decomposed into trees and the core using a graph decomposition algorithm. In the second stage, the trees of the original full network are individually optimized by BLP. In the third stage, the core of the original full network is optimized by a DE algorithm and the optimal solutions for the trees obtained in the second stage are incorporated during the DE optimization. The proposed BLP-DE method has been verified by two benchmark case studies each with a single demand loading case and a larger real-world network with multiple loading cases. The details of the proposed BLP-DE method are given in the next section.

THE PROPOSED BLP-DE METHOD

The first stage: water network decomposition

Deuerlein (2008) introduced the novel idea of decomposing a water network based on its connectivity properties, using terms and concepts drawn from graph theory; and describe how a full WDS could be decomposed into forests, blocks and bridges. This decomposition allowed various types of systems analysis to be conducted on water supply networks. In the first stage of the proposed method, the full water network is decomposed into two main components, rather than the forest, blocks and bridges. These two components are trees and the core, where trees are the outer component of the network, while the core is the inner component of the network (Deuerlein 2008).

Figure 1 represents the network layout of the New York tunnels problem (NYTP), a case study often used to test methods of WDS optimization. The NYTP is a pipe duplication optimization problem, the details of which are given by Dandy et al. (1996).

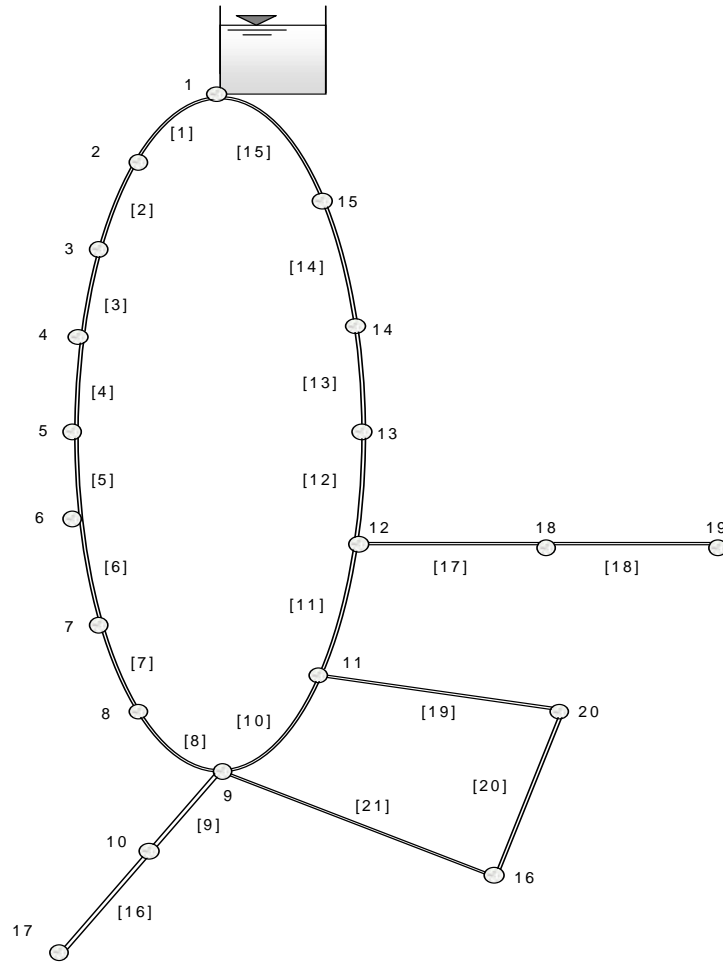


Figure 1 The network layout of the New York Tunnels problem

Normally, a WDS can be described as a graph $G(V,E)$, in which, vertices (V) of the graph represent the nodes of the WDS, and edges (E) of the graph represent links between nodes. For the NYTP network $G(V,E)$ given in Figure 1, $V=\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$ and $E=\{[1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]\}$.

In graph theory, a connected graph without any loops is referred as a tree (T) (Deo 1974). Based on the decomposition method proposed by Deuerlein (2008), the trees (T_s) and the core (C) of the NYTP network $G(V,E)$ (see Figure 1) are obtained and shown in Figure 2, where $G(V,E)= T_s \cup C$.

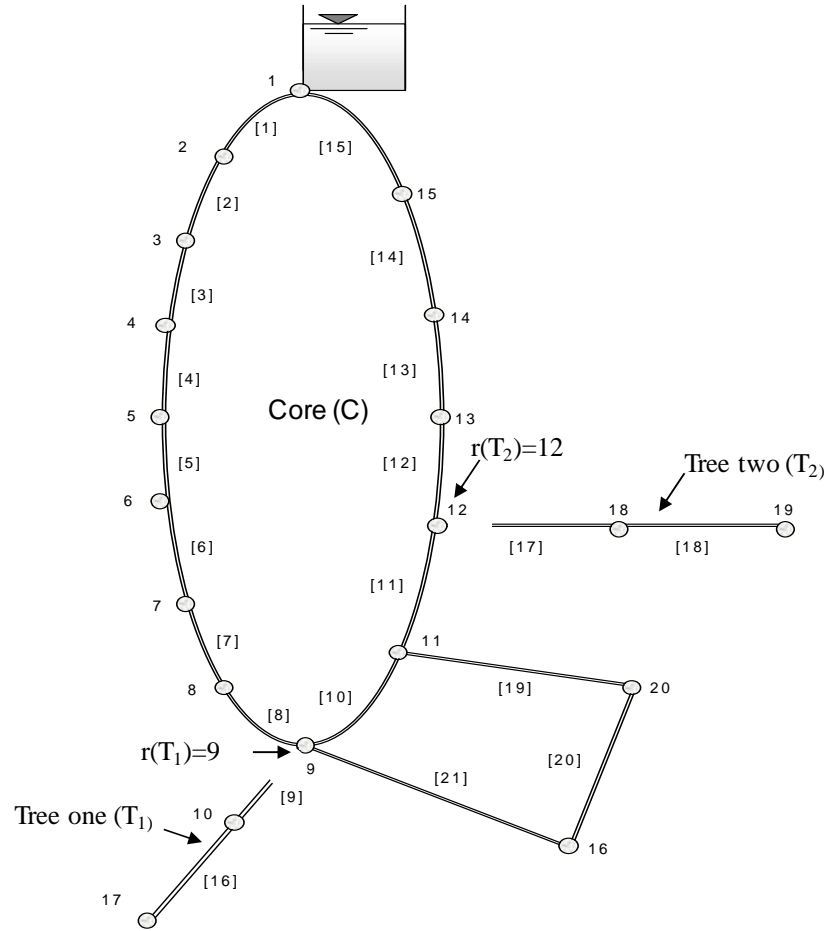


Figure 2 The decomposition results of the New York Tunnels problem

As shown in Figure 2, two trees are identified after decomposition, including $T_1 = \{10, 17, [9], [16]\}$ and $T_2 = \{18, 19, [17], [18]\}$, where 10, 17, 18 and 19 are nodes, and [9], [16], [17] and [18] are links in Figure 1. The remaining nodes and pipes form the core (C) of the NYTP network, where $C = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 20, [1], [2], [3], [4], [5], [6], [7], [8], [10], [11], [12], [13], [14], [15], [19], [20], [21]\}$. As can be seen from Figure 2, the trees and the core (C) overlap at the nodes 9 and 12, i.e., $T_1 \cap C = 9$ and $T_2 \cap C = 12$. The nodes that connect the core and the trees in the original water networks are defined as root nodes r (Deuerlein 2008). Thus, for the NYTP network given in Figure 1, nodes 9 and 12 are root nodes, i.e., $r(T_1) = 9$ and $r(T_2) = 12$, as shown in Figure 2.

The second stage: BLP optimization for the trees

In the proposed method, binary linear programming (BLP) is employed in the second stage to optimize the design of the trees obtained at the end of the first stage. Since the WDS optimization problem is mathematically nonlinear due to the nonlinearity of the head loss equation, during BLP, zero-one variables are used as decision variables in order to convert the optimization problem from a nonlinear to a binary linear system. The trees of the original full network are individually optimized by BLP in the proposed method. The BLP formulation for tree optimization is given as follows.

Objective function of BLP

The objective function involved for the least-cost WDS design is normally the sum of the construction cost of each pipe in the WDS. The objective function F for a tree in BLP is given by:

$$F = \sum_{i=1}^N \sum_{j=1}^P X_{ij} L_i C(D_j) \quad (1)$$

where N is the total number of pipes that needs to be optimized; P is the total number of commercially discrete pipe diameters that can be used; L_i is the length of pipe i ; $C(D_j)$ is the unit length cost of the pipe diameter D_j and X_{ij} is the zero-one variable.

In Equation (1), $X_{ij}=1$ indicates that the diameter D_j is selected for pipe i while $X_{ij}=0$ indicates that the diameter D_j is not selected for pipe i . No nonlinear terms are involved in the objective function F .

Constraints of BLP

Normally, when designing a WDS the hydraulic balance for the water network (including a continuity equation at each node and the energy conservation for each primary loop and required path) and the head requirement for each node are usually constraints that need to be satisfied. In the proposed method, however, BLP is only employed to deal with the trees of the original full network. Therefore, the hydraulic balance does not need to be considered as a constraint since no loops are involved in

the trees and the flows at each pipe in the trees can be determined for each demand loading case before BLP optimization is carried out.

The head requirement for each node still needs to be considered as a constraint for the BLP optimization in the second stage of the proposed method. The Hazen-Williams or Darcy-Weisbach formula may be used during the BLP to determine the head loss for each pipe. For the Hazen-Williams formulation,

$$h_{fi}^n = \sum_j^P \omega \frac{L_i X_{ij}}{C_j^\alpha D_j^\beta} (q_i^n)^\alpha \quad n \in TN, i \in T \quad (2)$$

where h_{fi}^n is the frictional head loss for pipe i for demand loading case n in the tree (T) that is being optimized; q_i^n =flows in pipe i for demand loading case n ; TN = total number of demand loading cases; P =total number of available pipe diameters; ω =numerical conversion constant which depends on the units of flows and diameters; α, β =coefficients; C_j =Hazen-Williams coefficient of pipe diameter D_j .

As can be seen from Equation (2), for pipe i , each pipe diameter D_j is considered as its potential option. The final diameter for pipe i is selected by using X_{ij} (the zero-one variables), where $X_{ij}=1$ implies that diameter D_j is used for pipe i and then the h_{fi}^n is based on the selected diameter D_j ; a value of $X_{ij}=0$ means that diameter D_j is not selected for pipe i and no head loss is involved for the diameter D_j .

The only unknown in Equation (2) is X_{ij} (the zero-one variables) since q_i^n is already determined for each link in the tree and each commercially available pipe diameter is known. Consequently, by utilizing the zero-one decision variables X_{ij} , the nonlinear Hazen-Williams formula is converted to a linear formula.

In the proposed BLP, the constraint for each node k is that the total head loss used by the pipes involved in the water supply path from source node R to node k should be less than the value of the head at the source node minus the head requirement at node k for each demand loading case, which is given by:

$$\sum_m^{W_{k-R}} h_{jm}^n \leq H_R - H_{k,n}^{\min} \quad k \in T, m \in W_{k-R} \quad (3)$$

where H_R is the available head provided by the source node (R) of the tree that is being optimized by BLP; $H_{k,n}^{\min}$ is the minimum head requirement for node k in the tree (T) for demand loading case n ; W_{k-R} is the water supply path from source node R to node k (only one path is available for each node k to receive water demand from the source node R in a tree). $\sum_m^{W_{k-R}} h_{jm}^n$ is total head loss involved in water supply path W_{k-R} for demand loading case $n \in TN$.

An additional constraint in BLP is that the sum of X_{ij} for each link i must be equal 1 as only one pipe diameter is selected for each link, which is given by:

$$\sum_{j=1}^P X_{ij} = 1 \quad (4)$$

BLP optimization for the trees

It is noted that no supply sources (R) are available for trees that are obtained by decomposing the original whole network and hence H_R is unknown in Equation (3). For the trees T_1 and T_2 given in Figure 2, no supply source is available for these two trees obtained by decomposition. For the purposes of the proposed optimization method, however, the root nodes r are assumed as the supply source nodes for the trees. This is because the root nodes are the connection of the trees and the core and all the water demands required by a particular tree are delivered via its corresponding root node. As such, the supply source nodes for T_1 and T_2 in Figure (2) are $r(T_1)=9$ and $r(T_2)=12$ respectively.

The water demands at the root nodes are not considered when conducting the tree optimization using BLP. A series of assumed H_R values are used for each root node to enable BLP optimization of the corresponding tree. In the proposed method, H_R values are selected from a pre-specified head range with a particular interval (of say 1 foot or 0.1 meters). The lower boundary of the head range is the maximum value of

the head requirement across the whole tree ($H_{min} = \max(H_k^{\min})$), while the upper boundary of the head range is the head provided by the supply source node of the original full network ($H_{max} = H_s$). T_1 of the NYTP network given in Figure 2 is used to illustrate the proposed BLP optimization algorithm. For the NYTP case study, a single demand loading case was specified as per the original paper (Schaaake and Lai 1969).

For T_1 , $H_{min} = 272.8$ ft and $H_{max} = 300$ ft, where 272.8 ft is the maximum value of the minimum allowable head requirement of all nodes contained within T_1 (nodes 10 and 17 as shown in Figure 2) and 300 ft is the allowable head provided by the reservoir given in Figure 1 (the head information is given by Dandy et al. 1996).

A series of values of H_R ($H_R \in [272.8, 300]$) with an increment of 1 ft is used for root node $r(T_1)$. BLP is formulated (see Equations (1), (2) and (3)) for each H_R value and solved. The final solutions for T_1 with different H_R assigned for $r(T_1)$ are presented in Table 1.

Table 1 Optimal solutions for T_1 of the NYTP

H_R at $R(T_1)$ (ft)	Cost of Optimal solutions from BLP (\$)	Duplicate pipe diameters ¹ (inches)	Minimum pressure head excess (H_e) (ft) ¹	$H_R^* = H_R - H_e$ (ft)
272.8	Infeasible solution	-	-	-
273.8	8,337,060	(0, 96)	0.09	273.71
274.8	7,0648,50	(0, 84)	0.64	274.16
275.8	5,835,646	(0, 72)	0.96	274.84
276.8	4,654,834	(0, 60)	1.00	275.80
277.8	3,529,683	(0, 48)	0.77	277.03
278.8	2,470,653	(0, 36)	0.47	278.33
279.8	2,470,653	(0, 36)	1.47	278.33
280.8 ²	0	(0, 0)	0.71	280.09

¹The first diameter is for link 9 and the second diameter is for link 16 in Figure 1. ²The solution is zero in cost as no pipe needs to be duplicated when the H_R is greater than 280.8 ft and hence these solutions are not given in Table 1.

As can be seen from Table 1, a lower cost solution was found by BLP when a higher head was assigned for $r(T_1)$. When H_R is equal to or larger than 280.8 ft, no pipes need to be duplicated and hence the solution is zero in cost since NYTP is a pipe duplication optimization problem. The fourth column of Table 1 displays the minimum pressure head excess H_e across the tree for each optimal solution obtained

by BLP. For each optimal solution, the corresponding H_R can be further reduced by its corresponding H_e while still maintaining its feasibility. For example, there is a minimum pressure excess of 0.09 ft ($H_e = 0.09$ ft) for the optimal solution with $H_R = 273.8$ ft at $r(T_1)$ (as shown in the third row of Table 1). This H_R value can be reduced to 273.71 ft while still guaranteeing the feasibility of the optimal solution. The reduced H_R value is denoted as H_R^* , which is the minimum head required at $r(T_1)$ to maintain the feasibility of its corresponding optimal solution ($H_R^* = H_R - H_e$). H_R^* values for all the optimal solutions for T_1 are provided in the fifth column of Table 1. It should be noted that for each optimal solution, the H_e value varies for different demand loading cases and the H_R^* value is therefore different for each loading case.

A solution choice table is developed for T_1 (denoted as $S(T_1)$) including the H_R^* values, the optimal solution costs and the pipe diameters of the optimal solutions. In $S(T_1)$, each unique H_R^* is associated with a unique optimal solution (including the cost and the pipe diameters for each link of T_1). In addition, H_R^* in the solution choice table is sorted from the smallest to the largest, while the optimal solution cost is sorted from the largest to the smallest. For each tree of the original full network, a solution choice table is constituted during the second stage of the proposed optimization method. For a water network having a total of TN demand loading cases, the solution choice table is composed of the optimal costs, pipe diameters for each optimal solution and $H_{R,n}^*$ ($n \in TN$) values for each demand loading case. In the solution choice table, each demand loading case is associated with a different set of $H_{R,n}^*$ values but the same optimal costs and the pipe diameters for the tree.

The third stage: DE optimization for the core

During the third stage of the proposed optimization method, a differential evolution (DE) algorithm is employed to optimize the design for the core of the original full network. The water demands at the root nodes in the core have to be increased by the total water demands of their corresponding trees before DE optimization. For the

example given in Figure 2, the water demands of T_1 and T_2 are added to the demands at node $r(T_1)=9$ and $r(T_2)=12$ in the cores respectively. Furthermore, during the DE optimization of the core, the optimal solutions obtained for the trees during the second stage are incorporated. The proposed DE optimization algorithm for the core is given as follows.

- (1) A total of N solutions (pipe diameter combinations for the core) are randomly generated for the core to initialize the DE search, where N is the population size of the DE algorithm. It should be noted here that only the pipes in the core are handled by the DE algorithm in the third stage of the proposed method.
- (2) For each individual solution, a hydraulic solver (EPANET2.0 [Rossman 2000]) was used in this study) is used to obtain the head at each node for each demand loading case. The head at each root node in the core for demand loading case n is tracked (denote as $H_{R,n}^\#$).
- (3) The total pipe cost of the core (PC) is computed for each individual solution of the DE algorithm. In addition, a penalty cost (PE) is computed for the solution has head deficits at the nodes in the core.
- (4) The optimal solutions for the trees during in the second stage are now incorporated into the DE process. The optimal solutions of trees are selected from their corresponding solution choice tables based on the head at the root nodes. The selection of the optimal solution for each tree from its corresponding solution choice table is guided by one of two possible sets of circumstances:
 - (i) If any head value at a root node for loading case $n \in TN (H_{R,n}^\#)$ in the core obtained by a hydraulic solver is smaller than the minimum $H_{R,n}^*$ value associated with its corresponding demand loading case n in the solution choice table of its corresponding tree, the optimal solution cost associated with the minimum $H_{R,n}^*$ is added to the PC . Additionally, a penalty cost is added to the PE for this individual solution as no feasible solution is found for this tree to satisfy the head constraints for all demand loading cases.

- (ii) If all $H_{R,n}^{\#}$ values in the core are greater than the minimum $H_{R,n}^*$ values of their corresponding demand loading cases in the solution choice table, the minimum optimal solution in terms of cost in the choice table that has all $H_{R,n}^*$ values smaller than their corresponding $H_{R,n}^{\#}$ values is selected and added to PC .

The total pipe cost is obtained, therefore, by combining the selected optimal solution cost for each tree and the cost for the core. The total penalty cost is achieved by adding the penalty cost for each tree (if applicable) and the penalty cost of the core (if applicable).

- (5) The objective function value is obtained for each individual solution of the DE by adding the total pipe cost and the total penalty cost. Then the mutation, crossover and selection operators of the DE are carried out to generate the offspring.
- (6) Steps (2) to (5) are performed iteratively until the convergence criterion is satisfied.

During the DE optimization, real continuous values for the pipe diameter are created in the mutation process although discrete pipe diameters are used to initialize the search. In the proposed method, the real diameter values are rounded to the nearest commercially discrete pipe diameters after the mutation operator of the DE is performed. Since the optimal solutions for the trees are included when the DE is optimizing the core, the final solution obtained is actually the optimal solution for the whole original network. However, the decision variables handled by the DE are only the pipes in the core as the solutions for the trees are selected from their existing solution choice tables. The DE, therefore, has a significantly reduced search space to explore, as defined by the core, while optimal solutions are provided for the whole network. This is the great benefit of the proposed optimization method.

CASE STUDIES

Two benchmark WDS case studies each with a single demand loading case are used to demonstrate the effectiveness of the proposed coupled BLP-DE method. These studies are the New York Tunnels problem (NYTP) and the Hanoi Problem (HP). A real-world WDS case study with multiple demand loading cases is then used to further verify the effectiveness of the BLP-DE method in terms of dealing with a relatively large and more complex case study. The DE algorithm and the BLP formulation were coded using Matlab 7.5 and the BLP was solved by the ‘bintprog’ function in the Matlab 7.5. It is noted that the EPANET2.0 was used in this paper to enable the hydraulic simulation. The Hazen-Williams equation (Equation (2)) was used. The coefficients of Hazen-Williams equation used in this paper according to those used in EPANET 2.0 were $\omega=10.670$ (SI units used in this study); $\alpha =1.852$ and $\beta=4.871$. For the NYTP and HP benchmark case studies, all the previously published results presented in this paper have used EPANET2.0 as the hydraulic simulation model and hence utilized the same coefficients of Hazen-Williams equation as those used in the proposed method. This therefore enables a fair comparison between the proposed BLP-DE method and other previously published algorithms.

Case study 1: New York Tunnels Problem (21 decision variables)

The layout of the NYTP and the decomposition results of the NYTP were given in Figure 1 and 2 respectively. Two trees were identified for the NYTP network and a series of values of H_R ($H_R \in [272.8, 300]$) with an increment of 1 foot was used for $r(T_1)=9$ to enable the BLP optimization for T_1 . For T_2 , a series of values of H_R ($H_R \in [255, 300]$) with an increment of 1 foot was used for $r(T_2)=12$ to conduct BLP optimization. A solution choice table was generated for each tree of the NYTP case study in the second stage of the proposed BLP-DE approach. Figure 3 gives the H_R^* value versus the optimal solution cost in the solution choice table for each tree.

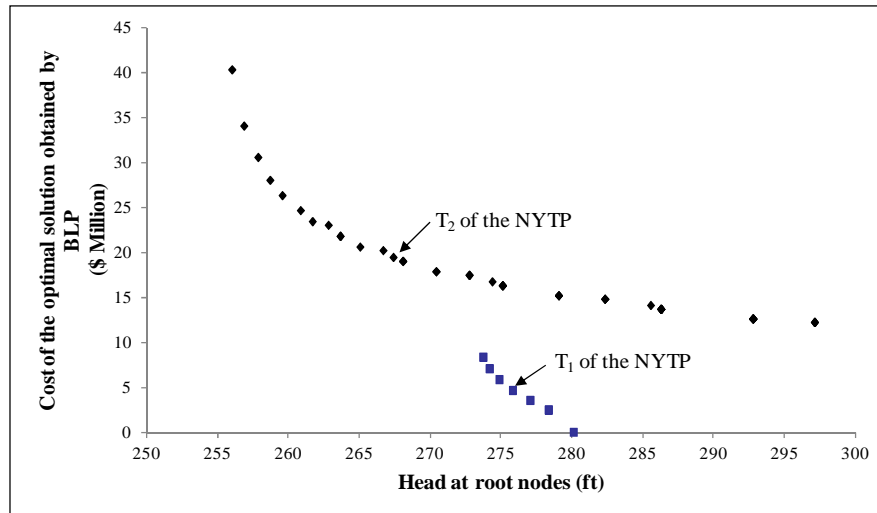


Figure 3 The H_R^* versus the optimal costs in the solution choice tables of two trees for the NYTP case study.

As shown in Figure 3, only seven different optimal solutions were found for T_1 with a large range of assumed heads at root node $r(T_1)$, while 23 different optimal solutions can be located for T_2 . For each tree, the optimal solution cost decreases as the head value at the root node increases. These optimal solutions are used to generate solution choice tables that are used during the third stage of the proposed method

The number of decision variables in the core is 17, compared to 21 decision variables in the original full NYTP, as four pipes are located in the trees. A population size (N) of 50, a weighing factor (F) of 0.5 and a crossover rate (Cr) of 0.5 are used for the DE applied to the core optimization in the third stage of the proposed method. The number of maximum allowable evaluations ($NMAE$) is 7,500 and 100 runs with different starting random number seeds are performed for the DE applied to the core. The results of the proposed BLP-DE method and other optimization algorithms that have previously been applied to the NYTP case study are presented in Table 2.

The current best known solution for the NYTP case study is \$38.64 million (Maier et al. 2003). In the current study, this best solution was found with a 100% success rate by the proposed BLP-DE method over 100 different runs. The rate at which the best

known solution is found by the new BLP-DE method is higher than all the other optimization algorithms in Table 2.

The most noticeable advantage of the proposed BLP-DE method over other optimization algorithms is the efficiency improvement. The proposed BLP-DE approach required only an average of 3,486 evaluations over 100 different runs to find the optimal solutions, which is significantly less than those required by all the other algorithms given in Table 2. It can be concluded that, for the NYTP case study, the proposed BLP-DE outperformed all the other optimization algorithms given in Table 2 in terms of percentage with the current best solution found and efficiency.

Table 2 Summary of the results of the proposed method and other EAs applied to the NYTP case study

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Algorithm ¹⁰	No. of runs	Best solution (\$M)	Percent of trials with best solution found	Average cost (\$M)	Average evaluations to find first occurrence of the best solution	Maximum allowable evaluations or evaluations for convergence
BLP-DE ¹	100	38.64	100%	38.64	3,486 ⁹	7,500
NLP-DE ²	100	38.64	99%	38.64	8,277	20,000
GHEST ³	60	38.64	92%	38.64	11,464	-
HD-DDS ⁴	50	38.64	86%	38.64	13,000	50,000
Suribabu DE ⁵	300	38.64	71%	NA	5,492	10,000
Scatter Search ⁶	100	38.64	65%	NA	57,583	-
MMAS ⁷	20	38.64	60%	38.84	30,700	50,000
PSO variant ⁸	2000	38.64	30%	38.83	-	80,000

¹Results from this study. ²Zheng et al. (2011). ³Bolognesi et al. (2010). ⁴Tolson et al. (2009). ⁵Suribabu (2010). ⁶Lin et al. (2007). ⁷Zecchin et al. (2007). ⁸Montalvo et al. (2008). ⁹The total computational overhead required by proposed BLP-DE method has been converted to the equivalent number of full NYTP evaluations. ¹⁰Results are ranked based on column (4).

It should be noted that all the computational overhead of the proposed BLP-DE method (including the BLP optimization for the trees and the DE optimization for the core) was converted to the equivalent number of full NYTP network evaluations using the same computer configuration. In particular, the full network was run 1000 times with randomly selected pipe configurations using the Matlab code developed for this proposed method. The average computational time for each full network

simulation was obtained. Then the total computational time used by the core optimization and the BLP optimization applied to the trees was converted to the equivalent time for full network simulations. This allows a fair comparison between the proposed BLP-DE method and other optimization algorithms in terms of efficiency. This computational analysis approach has been used for each case study investigated in this paper.

Case study 2: Hanoi Problem (34 decision variables)

The Hanoi Problem (HP) (Fujiwara and Khang 1990) has frequently been used as a benchmark WDS case study to test the performance of various optimization algorithms. The layout of the HP is given in Figure 4. The details of the HP, the available pipe diameters and the cost of these diameters are given by Fujiwara and Khang (1990).

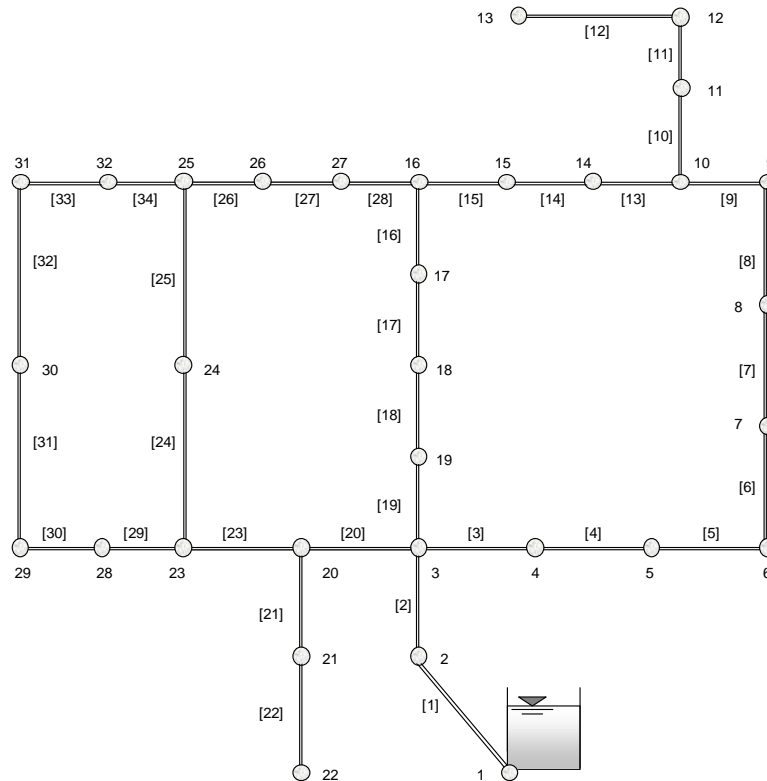


Figure 4 The network layout of the HP case study

The decomposition results for the HP network in the first stage of the proposed BLP-DE method are given in Figure 5. As shown in Figure 5, two trees were identified for the HP network including $T_1=\{11, 12, 13, [10], [11], [12]\}$ and $T_2=\{21, 22, [21], [22]\}$, where 11, 12, 13, 21 and 22 are nodes, and [10], [11], [12], [21] and [22] are links. $r(T_1)=10$ and $r(T_2)=20$ are root nodes of the T_1 and T_2 respectively as $T_1 \cap C=10$ and $T_2 \cap C =20$ (C is the core of the HP network as shown in Figure 5).

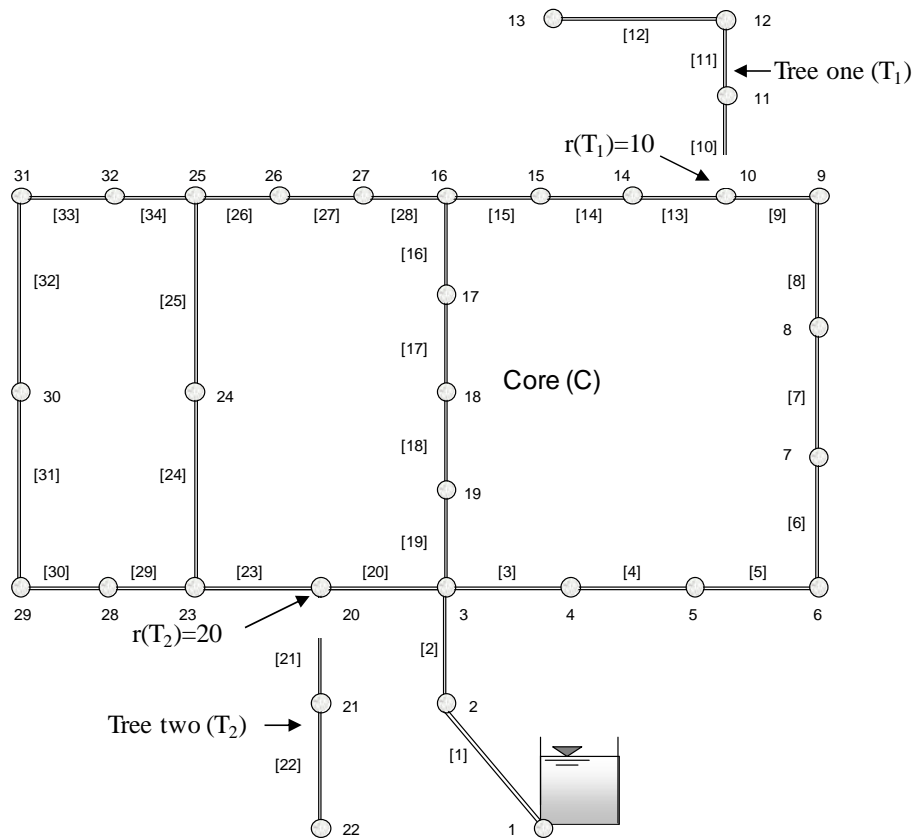


Figure 5 The decomposition results of the HP case study

For the HP case study, the head provided by the reservoir is 100 meters and the minimum head requirement for each node is 30 meters (Fujiwara and Khang 1990). In this study, a series of H_R ($H_R \in [30,100]$) with an increment of 0.1 meter was used for $r(T_1)$ and $r(T_2)$ to enable the BLP optimization of T_1 and T_2 in the second stage, thereby generating a solution choice table for each of the trees. The values of

H_R^* and the optimal solution costs from the solution choice table for each tree are presented in Figure 6.

As shown in Figure 6, 18 different optimal solutions were found for each of both T_1 and T_2 of the HP network, although 700 BLP runs with a range of heads between 30 and 100 meters at the root nodes (0.1 meter interval) were performed for each tree. This indicates that a larger interval (of say 0.5 meter or 1 meter) may be enough to obtain these 18 optimal solutions for each tree. However, due to the extreme efficiency for the BLP applied to the tree optimization, a relatively small interval (0.1 meter) was used in this study to improve the likelihood of including all possible optimal solutions. These optimal solutions are used to form solution choice tables for the trees, which are used for the DE optimization in the third stage of the proposed method.

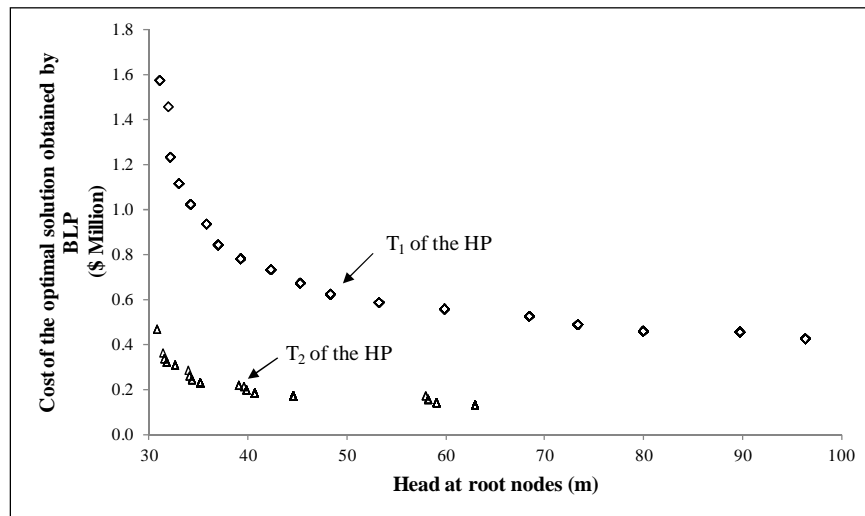


Figure 6 The H_R^* versus the optimal costs in the solution choice tables of two trees for the HP case study.

In the third stage, a DE with $N=80$, $F=0.7$, $Cr=0.8$, and $NMAE=40,000$ was applied to the core of the HP network. The number of decision variables to be considered is 29 since five pipes of the 34 pipes in the HP network are consigned to the two trees. Based on the heads at the root nodes, the DE algorithm chooses the optimal solutions for the trees from their corresponding solution choice tables while also exploring the

search space of the core. As such, the optimal solutions for the tree are used in conjunction with solutions from the core to yield optimal solutions for the whole HP network.

Table 3 provides the final results of the proposed BLP-DE approach applied to the HP case study. The other previously published results for the HP case study are also included in Table 3 to enable the performance comparison. The current best known solution for the HP case study, with a cost of \$6.081 million, was first found by Reca and Martínez (2006). As shown in Table 3, the proposed BLP-DE performs the best in terms of the percentage of the current best solution found for the HP case study. This is reflected by the fact that the proposed BLP-DE found the current best solution for the HP case study 98% of the time over 100 runs using different starting random number seeds, which is higher than all the other algorithms given in Table 3.

Table 3 Summary of the results of the proposed BLP-DE method and other EAs applied to the HP case study

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Algorithm ¹¹	No. of runs	Best solution (\$M)	Percent of trials with best solution found	Average cost (\$M)	Average evaluations to find first occurrence of the best solution	Maximum allowable evaluations or evaluations for convergence
BLP-DE ¹	100	6.081	98%	6.085	33,148 ¹⁰	40,000
NLP-DE ²	100	6.081	97%	6.082	34,609	80,000
Suribabu DE ³	300	6.081	80%	NA	48,724	100,000
Scatter Search ⁴	100	6.081	64%	NA	43,149	-
GHEST ⁵	60	6.081	38%	6.175	50,134	-
GENOME ⁶	10	6.081	10%	6.248	NA	150,000
HD-DDS ⁷	50	6.081	8%	6.252	100,000	100,000
PSO variant ⁸	2000	6.081	5%	6.310	NA	500,000
MMAS ⁹	20	6.134	0%	6.386	85,600	100,000

¹Results from this study. ²Zheng et al. (2011). ³Suribabu (2010). ⁴Lin et al. (2007). ⁵Bolognesi et al. (2010). ⁶Reca and Martínez (2006). ⁷Tolson et al. (2009). ⁸Montalvo et al. (2008). ⁹Zecchin et al. (2007). ¹⁰The total computational overhead required by proposed BLP-DE method has been converted to the equivalent number of full HP evaluations. ¹¹Results are ranked based on column (4).

The total computational effort required by the proposed BLP-DE method, including the BLP optimization for the trees and the DE optimization for the core, has been converted to the equivalent number of full HP network evaluations. As displayed in

Table 3, the proposed BLP-DE used an average number of 33,148 evaluations to find the optimal solutions, which is fewer than all the other algorithms shown in Table 3. This indicates that the proposed BLP-DE is able to find optimal solutions more quickly than other algorithms.

Case study 3: Real-world network case study (96 decision variables)

The real-world network case study (denoted as RN case study) was taken from a small town in the south of China. This is a completely new case study and has not been previously studied. The RN network has 96 pipes, 85 demand nodes and one reservoir with a fixed head of 50 meters. Three demands loading cases have been considered for this network including a peak hour demand loading case and two fire loading cases. The layout of RN case study and the two fire loading positions are shown in Figure 7.

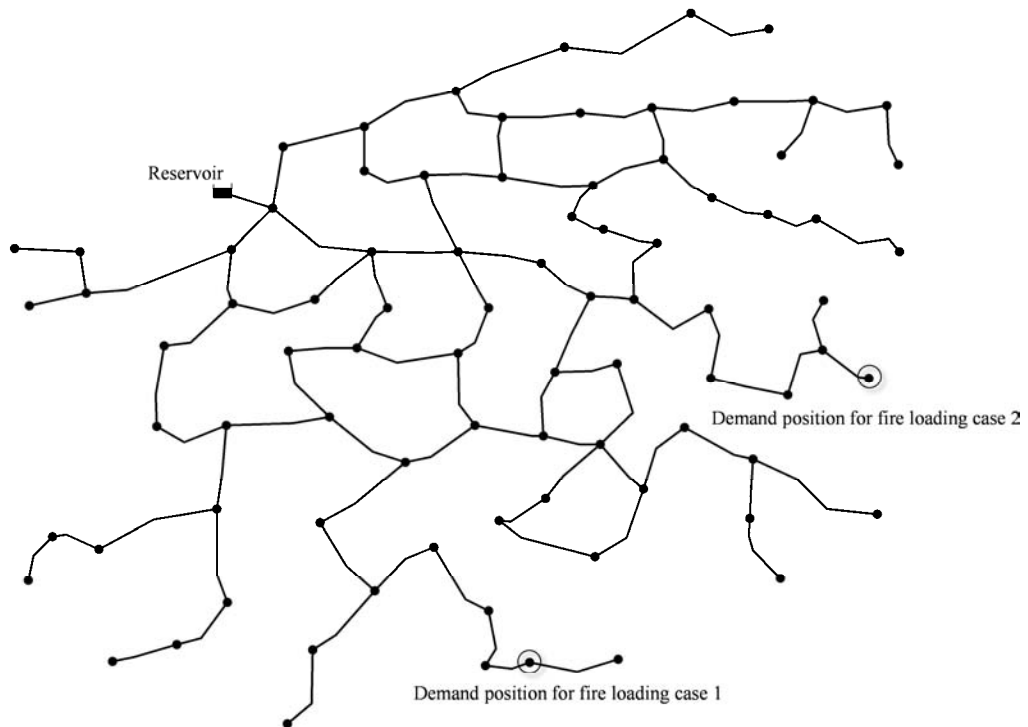


Figure 7 The network layout of the RN case study

The objective of this case study is to determine the least-cost design of this water network, while satisfying a minimum pressure of 15 meters at each demand node for all demand loading cases and 10 meters only at the fire demand loading nodes during the two separate fire loading cases. All the pipes have an identical Hazen-Williams coefficient of 130. A total of 14 commercially available pipe diameters ranging from 150 mm up to 1000 mm are available for selection for each pipe and the cost of each pipe is given by Kadu et al. (2008).

The graph decomposition algorithm was applied to RN case study in the first stage of the proposed method in order to identify the trees and the core. The decomposition results are given in Figure 8. As shown in Figure 8, a total of eight trees are determined and 43 pipes are assigned to these trees.

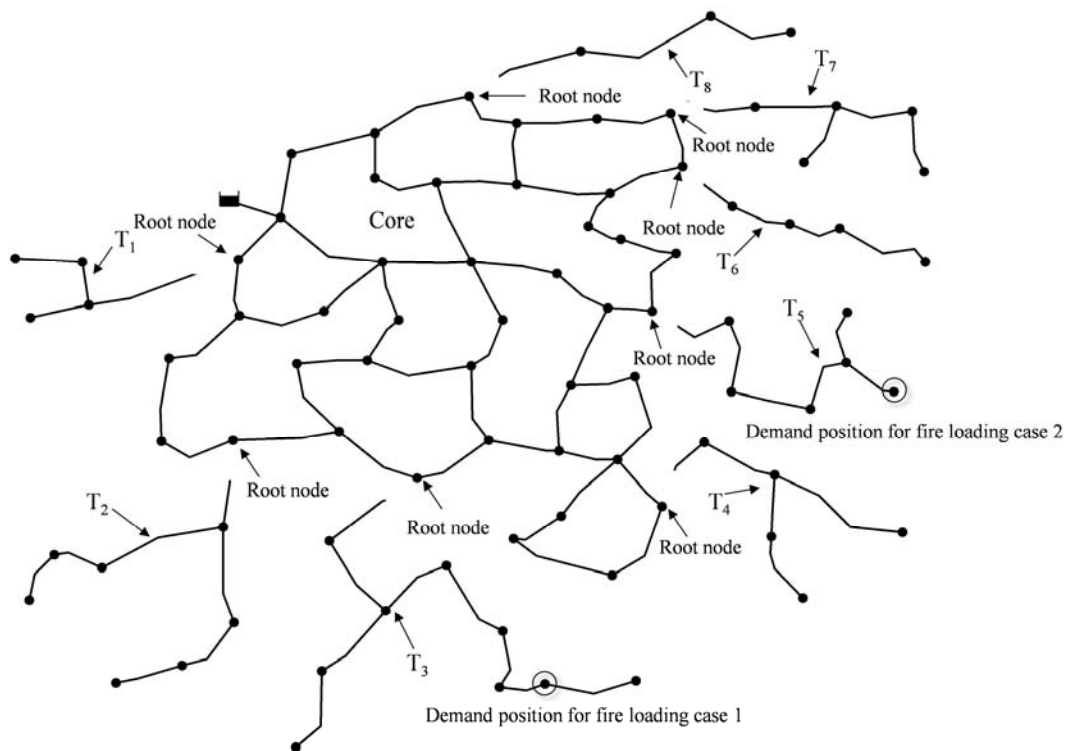


Figure 8 The decomposition results of the RN case study

In the second stage of the proposed method, eight solutions choice tables were generated for the trees. The number of pipes involved in the core of RN case study is

53 since 43 pipes are assigned to trees (96 pipes exist in the full RN case study). Thus, only the 53 pipes rather than the 96 pipes are handled by the DE in the third stage of the proposed method. For the DE applied to the core optimization of the RN case study, $N=150$, $F=0.4$ and $Cr=0.8$ were selected based on a few parameter trials. The maximum number of allowable evaluations was set 75,000 ($NMAE=75,000$).

In order to enable a performance comparison, a standard DE (SDE) was also applied to the original full RN case study (96 pipes). For the SDE, $N=300$, $F=0.3$, $Cr=0.8$ and $NMAE=600,000$ were selected based on a detailed preliminary analysis. Ten runs with different starting random number seeds were performed for the DE applied to the core in the third stage of the proposed method and the SDE applied to the original full network (the proposed method and the SDE used the same random number seeds). It is not necessary to perform multiple runs for the tree optimization in the second stage of the proposed method. This is because that the same solutions are found for the trees using the deterministic BLP method. The solutions of the proposed BLP-DE method and the SDE algorithm applied to the BN case study are provided in Figure 9.

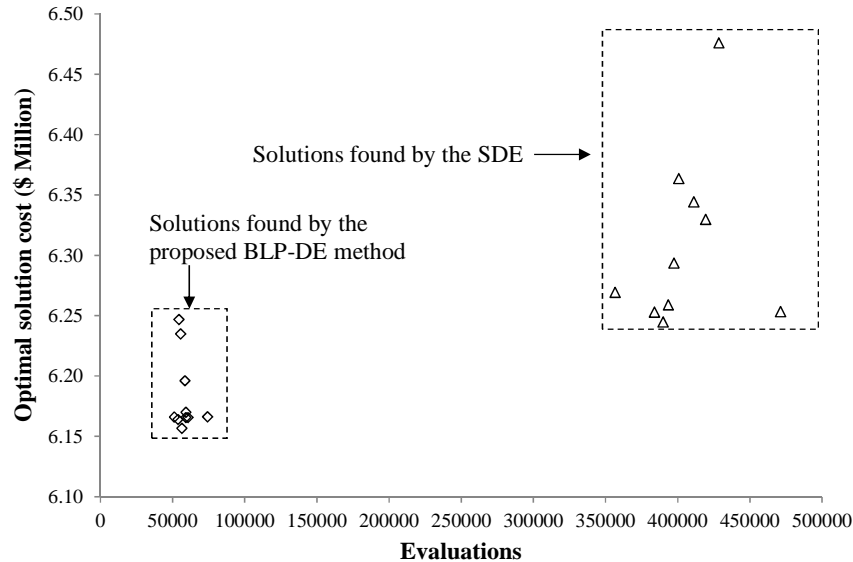


Figure 9 Solution distributions of two algorithms applied to the RN case study (10 runs for each)

It is clearly shown in Figure 9 that the proposed BLP-DE method significantly outperformed the standard different evolution (SDE) with calibrated parameter values for the RN case study in terms of solution quality and efficiency. In addition, it is observed that the solutions produced by the proposed BLP-DE are less scattered than those generated by the SDE based on ten different runs. This indicates that the performance the proposed BLP-DE approach is less affected by the random number seeds than that of the SDE. The proposed method is therefore able to yield good quality optimal solutions with a higher confidence level.

The results of the proposed BLP-DE and the SDE applied to the RN case study are provided in Table 4. As shown in Table 4, the proposed BLP-DE found a best solution for the RN case study with a cost of \$6.16 million. The best solution found by the SDE was \$6.24 million, which is 1.3% higher than the best solution produced by the proposed BLP-DE method. The average cost solution over 10 runs found by the BLP-DE method was \$6.18 million, which is only 0.3% higher than the best solution (\$6.16 million found by the proposed BLP-DE method in this study) while 2.1% lower than that generated by the SDE.

Table 4 Summary of the results of the proposed BLP-DE method and the SDE applied to the RN case study

Algorithm	No. of runs	Best solution (\$M)	Percent of trials with best solution found	Average cost (\$M)	Average evaluations to find first occurrence of the best solution	Maximum allowable evaluations	Total execution time (hours) for 10 runs to find optimal solutions on Matlab 7.5 ³
BLP-DE ¹	10	6.16	10	6.18	73,092 ²	75,000	2.2
SDE ¹	10	6.24	0	6.31	405,330	600,000	12.4

¹Results from this study. ²The total computational overhead required by proposed BLP-DE method has been converted to the equivalent number of full RN evaluations. ³The computer configuration is a 3.0 GHz Pentium PC (Inter R).

In terms of efficiency, the proposed BLP-DE was approximately 4.5 times more efficiently in terms of average numbers of evaluations than the SDE in finding optimal solutions for this case study. This is evidenced by the fact that the average number of equivalent full network evaluations required by the proposed BLP-DE

method for convergence was 73,092, which is only 18% of the number of evaluations required by the SDE. The average number of 73,092 was obtained by converting the total computational overhead of the proposed method to the equivalent number of full RN network evaluations. In terms of comparing the execution time in Matlab 7.5, the proposed method required a total of 2.2 hours to find optimal solutions for ten optimization runs, while the SDE needed a total of 12.4 hours to arrive at optimal solutions for ten runs. For the proposed method, the total execution time for the BLP optimization was 0.48 hours, which is 21% of that required by the total execution time of the proposed BLP-DE method. Note that the graph decomposition process is extremely efficient and hence its computational overhead is not included.

CONCLUSIONS

In this paper, a novel coupled binary linear programming and differential evolution (BLP-DE) optimization approach based on network decomposition is proposed for dealing with WDS optimization problems. Three stages are involved in the proposed BLP-DE optimization method. These are (i) the full water network is decomposed into trees and the core using a graph decomposition algorithm; (ii) the trees are individually optimized by BLP and a solution choice table is constituted for each tree and (iii) a DE is employed to optimize the core of the original full network while incorporating the optimal solutions for any tree. Different components of the original full network are, therefore, optimized by different optimization algorithms.

The proposed BLP-DE method has been applied to three case studies and the results compared with those of other algorithms. For the NYTP and HP case studies, the proposed BLP-DE method found the current best known solutions for both of them with a higher success rate and a significantly improved efficiency compared to other algorithms given in Tables 2 and 3. For the relatively larger and more complex case study (RN case study with three demand loading cases: 96 decision variables), the proposed BLP-DE was able to find better quality optimal solutions than a standard differential evolution (SDE) with approximately 4.5 times faster convergence speed.

Thus, the proposed method shows substantial promise as a tool for finding good quality solutions for relatively large water networks.

It should be noted that the proposed BLP-DE method is not appropriate to optimize the types of water networks having only loops or only trees. However, it is very common for a water network to have loops and multiple trees in practice and the proposed BLP-DE method has advantages in efficiently finding good quality optimal solutions for this common type of network over other optimization methods as demonstrated in this paper. It has been also found in this study that the computational overhead increases significantly when the number of decision variables in the tree handled by the BLP increases. Thus, the proposed method would need to be further developed when dealing with the water network having very large trees to be optimized by the BLP. Other future studies on this research line may include (i) the application of the proposed BLP-DE method to deal with the optimization of complex water networks, which may have pumps, valves and tanks involved, and (ii) the extension of the proposed BLP-DE method to tackle the multi-objective optimization problem for WDSs.

ACKNOWLEDGEMENTS

The authors are grateful to School Editor Barbara Brougham for improving the quality of this manuscript. The first author gratefully acknowledges the scholarship provided by Australian government and the University of Adelaide.

REFERENCES

- Alperovits, E., and Shamir, U. (1977). "Design of optimal water distribution systems." *Water Resour. Res.*, 13(6), 885-900.
- Bolognesi, A., Bragalli, C., Marchi, A., and Artina, S. (2010). "Genetic heritage evolution by stochastic transmission in the optimal design of water distribution networks." *Advances in Engineering Software*, 41(5), 792-801.
- Cunha, M. d. C., and Sousa, J. (2001). "Hydraulic infrastructures design using simulated annealing." *Journal of Infrastructure Systems*, 7(1), 32-39.

- Dandy, G. C., Simpson, A. R., and Murphy, L. J. (1996). "An improved genetic algorithm for pipe network optimization." *Water Resour. Res.*, 32(2), 449-458.
- Deo, N (1974). *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall, Englewood Cliffs, N. J.
- Deuerlein, J. W. (2008). "Decomposition model of a general water supply network graph." *Journal of Hydraulic Engineering*, 134(6), 822-832.
- Eusuff, M. M., and Lansey, K. E. (2003). "Optimization of water distribution network design using the shuffled frog leaping algorithm." *Journal of Water Resources Planning and Management*, 129(3), 210-225.
- Fujiwara, O., and Khang, D. B. (1990). "A two-phase decomposition method for optimal design of looped water distribution networks." *Water Resour. Res.*, 26(4), 539-549.
- Geem, Z. W., Kim, J. H., and Loganathan G. V. (2002). "Harmony search optimization: application to pipe network design." *Int. J. Model. Simulat.*, 22(2), 125-133.
- Gessler, J. (1985). "Pipe network optimization by enumeration." *Proc., Computer Application for Water Resources*, ASCE, New York, N.Y., 572-581.
- Cisty, M. (2010). "Hybrid genetic algorithm and linear programming method for least-cost design of water distribution systems." *Water Resources Management*, 24(1), 1-24.
- Guercio, R., and Xu, Z. (1997). "Linearized optimization model for reliability-based design of water systems." *Journal of Hydraulic Engineering*, 123(11), 1020-1026.
- Haghighi, A., Samani, H., and Samani, Z. (2011). "GA-ILP method for optimization of water distribution networks." *Water Resources Management*, 25(7), 1791-1808.
- Kadu, M. S., Gupta, R., and Bhave, P. R. (2008). "Optimal design of water networks using a modified genetic algorithm with reduction in search space." *Journal of Water Resources Planning and Management*, 134(2), 147-160.
- Krapivka, A., and Ostfeld, A. (2009). "Coupled genetic algorithm---linear programming scheme for least-cost pipe sizing of water-distribution systems." *Journal of Water Resources Planning and Management*, 135(4), 298-302.
- Lansey, K. E., and Mays, L. W. (1989). "Optimization model for water distribution system design." *Journal of Hydraulic Engineering*, 115(10), 1401-1418.

- Lin, M.-D., Liu, Y.-H., Liu, G.-F., and Chu, C.-W. (2007). "Scatter search heuristic for least-cost design of water distribution networks." *Engineering Optimization*, 39(7), 857 - 876.
- Maier, H. R., Simpson, A. R., Zecchin, A. C., Foong, W. K., Phang, K. Y., Seah, H. Y., and Tan, C. L. (2003). "Ant colony optimization for design of water distribution systems." *Journal of Water Resources Planning and Management*, 129(3), 200-209.
- Martínez, J. B. (2008). "Discussion of "Optimization of water distribution networks using Integer linear programming" by Hossein M. V. Samani and Alireza Mottaghi." *Journal of Hydraulic Engineering*, 134(7), 1023-1024.
- Montalvo, I., Izquierdo, J., Pérez, R., and Tung, M. M. (2008). "Particle swarm optimization applied to the design of water supply systems." *Computers & Mathematics with Applications*, 56(3), 769-776.
- Perelman, L., and Ostfeld, A. (2007). "An adaptive heuristic cross-entropy algorithm for optimal design of water distribution systems." *Engineering Optimization*, 39(4), 413-428.
- Rossman, L. A. (2000). "EPANET 2-User manual." National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency, Cincinnati, OH.
- Samani, H. M. V., and Mottaghi, A. (2006). "Optimization of water distribution networks using integer linear programming." *Journal of Hydraulic Engineering*, 132(5), 501-509.
- Savic, D., and Cunha, M. d. C. (2008). "Discussion of "Optimization of water distribution networks using integer linear programming" by Hossein M. V. Samani and Alireza Mottaghi." *Journal of Hydraulic Engineering*, 134(7), 1024-1025.
- Schaake, J., and Lai, D. (1969). "Linear programming and dynamic programming applications to water distribution network design." Rep.116, Dept. of Civ. Engrg., Massachusetts Inst. of Techno., Cambridge, Mass.
- Simpson, A. R., Dandy, G. C., and Murphy, L. J. (1994). "Genetic algorithms compared to other techniques for pipe optimization." *Journal of Water Resources Planning and Management*, 120(4), 423-443.
- Sonak, V. V., and Bhawe, P. R. (1993). "Global optimum tree solution for single-source looped water distribution networks subjected to a single loading pattern." *Water Resour. Res.*, 29(7), 2437-2443.

- Suribabu, C. R., and Neelakantan, T. R. (2006). "Design of water distribution networks using particle swarm optimization." *Urban Water J.*, 3(2), 111-120.
- Suribabu, C. R. (2010). "Differential evolution algorithm for optimal design of water distribution networks." *Journal of Hydroinformatics*, 12(1), 66-82.
- Tolson, B. A., Asadzadeh, M., Maier, H. R., and Zecchin, A. (2009). "Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization." *Water Resour. Res.*, 45(12), W12416.
- Vairavamoorthy, K., and Ali, M. (2005). "Pipe index vector: a method to improve genetic-algorithm-based pipe optimization." *Journal of Hydraulic Engineering*, 131(12), 1117-1125.
- Zheng, F., Simpson, A. R., and Zecchin, A. C. (2011). "A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems." *Water Resour. Res.*, 47(8), W08531.
- Zecchin, A. C., Maier, H. R., Simpson, A. R., Leonard, M., and Nixon, J. B. (2007). "Ant colony optimization applied to water distribution system design: comparative study of five algorithms." *Journal of Water Resources Planning and Management*, 133(1), 87-92.

Chapter 8. Journal Paper 6-Decomposition Optimization of WDS with Multiple Sources

8.1 Synopsis

A decomposition and multi-stage optimization approach applied to the optimization of water distribution systems with multiple supply sources

Chapters 3, 4 and 5 have outlined the research outcomes based on the first objective of this study (see Section 1.1 in Chapter 1). Chapters 6 and 7 have introduced two new hybrid optimization approaches for optimizing water distribution systems (WDSs) developed in this research, which is the second objective of this study. These two methods have been demonstrated to be effective in terms of efficiently finding good quality optimal solutions based on testing real-world sized water distribution systems.

For the third and fourth objectives presented in Section 1.1 of Chapter 1, two advanced optimization methods that incorporate graph decomposition techniques during the optimization process have been developed in this research, which are outlined in Chapters 8 and 9. Two new water network decomposition concepts have been developed to facilitate network optimization, which are optimal source partitioning technique and the sub-network identification approach. These two network decomposition methods are presented in Chapters 8 and 9 respectively.

In this Chapter, a decomposition and multi-stage optimization method developed in this research is introduced. For a WDS, a multiple source of supply strategy is normally adopted in addition to the presence of loops in the WDS in order to improve supply reliability. For such a complex WDS with multiple supply sources (WDS-MSS), existing algorithms normally tackle the system as a whole in order to find optimal design solutions. In contrast, a decomposition and multi-stage optimization approach is developed in this research to deal with the optimization for the WDS-MSS design.

In the proposed decomposition and multi-stage optimization method, an algorithm is developed to identify the optimal source partitioning cut-set for a WDS with K supply

sources. As such, the whole original WDS is decomposed to K sub-networks by removing the optimal source partitioning cut-set. For each sub-network, one and only one supply source is assigned. Each sub-network is then optimized by a DE algorithm independently, which is the first stage optimization phase. It is expected that the optimal solution for each sub-network will be achieved with great efficiency as a significantly reduced search space (compared with the original search space of the entire network) is explored by the DE algorithm.

The optimal solutions for all sub-networks are then combined to provide an approximate optimal solution for the whole original network. However, this approximate optimal solution needs to be further improved as the pipes within the optimal source partitioning cut-set are not included during the sub-network optimization (first stage optimization). Thus, a second phase DE is used to explore the search space in the region around the obtained approximate optimal solution and better quality solutions for the whole WDS are expected to be found with significant reduced computational effort. This is the second stage of the optimization process.

The concept of multi-stage optimization is based on the decomposition of large-scale and complex systems into independent subsystems (although these sub-systems are actually interconnected and are not truly independent of one another). Each subsystem is optimized independently, and the optimal solutions for each subsystem are then combined together to derive the optimal solution for the whole system. This is the first known work to undertake the multi-stage optimization technique for designing WDS.

This work has been published in *Water Resources Research* and the paper is presented here.

Citation of Paper

Zheng, F., A. R. Simpson, and A. C. Zecchin (2012d). “A decomposition and multi-stage optimization approach applied to optimization of water distribution systems with multiple sources.” *Water Resources Research*, doi:10.1029/2012WR013160.

STATEMENT OF AUTHORSHIP (PUBLISHED)

Journal paper title: **A decomposition and multi-stage optimization approach applied to the optimization of water distribution systems with multiple supply sources.**

Authors: Feifei Zheng¹, Aaron Zecchin² and Angus Simpson³

Corresponding author:

¹Feifei Zheng (Candidate): PhD student, School of Civil, Environmental and Mining Engineering, University of Adelaide

Wrote the manuscript, performed all analysis, developed the model and theory and acted as corresponding author.

I hereby certify that the statement of contribution is accurate.

Signed _____ Date _____

²Aaron Zecchin: Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide

Extensive discussion of concepts and reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

³Angus Simpson: Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide

Extensive discussion of concepts and reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

Water Resources Research, doi:10.1029/2012WR013160.

8.2 Journal Paper 6: A decomposition and multi-stage optimization approach applied to the optimization of water distribution systems with multiple supply sources (Published in *Water Resources Research*)

Feifei Zheng, Aaron C. Zecchin and Angus R. Simpson

Abstract

The aim of this paper is to present a decomposition and multi-stage approach for optimizing the design of water distribution systems with multiple supply sources (WDS-MSS). An algorithm is first proposed to identify the optimal source partitioning cut-set for a WDS-MSS. A WDS with K supply sources is therefore decomposed to K disconnected sub-networks by the removal of the determined cut-set. Then a total of K separate differential evolution (DE) algorithms are used to optimize the designs for the K sub-networks respectively. This is the first optimization stage. The optimal solutions for the K sub-networks plus the optimal cut-set being the minimum allowable pipe sizes are used to create a tailored seeding table. This table is used to initialize a second stage DE algorithm to optimize the whole of the original WDS, which is the second stage of the optimization process. Four WDS-MSS case studies are used to demonstrate the effectiveness of the proposed method. A standard DE algorithm (SDE) seeded by the total choice table rather than the tailored seeding table is applied to the entire network for each case study and the results are compared with those of the proposed method in terms of efficiency and solution quality. The comparison demonstrates that the proposed method (i.e., decomposition followed by multi-stage optimization) shows better performance than results from a whole of network optimization. Additionally, the proposed method also exhibits significantly improved performance compared with the optimization techniques that have been previously used to optimize these case studies.

1. Introduction

Over the last four decades, significant research has been undertaken to develop techniques to optimize the design of water distribution systems (WDSs). Various

optimization techniques including traditional optimization methods and evolutionary algorithms (EAs) have been applied to WDS optimization and these are summarized in Table 1 (it should be noted that only the first significant paper for each optimization technique applied to WDS optimization is provided in Table 1). Traditional optimization techniques such as linear programming (LP) and nonlinear programming (NLP) often converge at local optimal solutions due to the nonsmoothness properties of the WDS optimization problem [Eiger *et al.* 1994]. EAs given in Table 1 have been demonstrated to be able to find better quality solutions than traditional optimization methods based on testing on a number of WDS case studies. One major drawback with using EAs, however, is that they require a large number of network evaluations to find optimal solutions, resulting in an expensive computational overhead, especially for relatively large case studies. Thus, it is difficult for these EAs to find good quality optimal solutions for real-world sized WDSs, as these systems are generally complex, with large numbers of decision variables.

Table 1 Types of previously used optimization techniques applied to WDS optimization

Algorithm¹	First reference
Linear programming (LP)	<i>Alperovits and Shamir</i> [1977]
Nonlinear programming (NLP)	<i>Fujiwara and Khang</i> [1990]
Standard genetic algorithm (SGA)	<i>Simpson et al.</i> [1994]
Modified genetic algorithm (MGA)	<i>Dandy et al.</i> [1996]
Simulated annealing (SA)	<i>Loganathan et al.</i> [1995]
Tabu search (TS)	<i>Lippai et al.</i> [1999]
Harmony search (HS)	<i>Geem et al.</i> [2002]
Shuffled frog leaping algorithm (SFLA)	<i>Eusuff and Lansey</i> [2003]
Ant colony optimization (ACO)	<i>Maier et al.</i> [2003]
ANN metamodels	<i>Broad et al.</i> [2005]
Particle swarm optimization (PSO)	<i>Suribabu and Neelakantan</i> [2006]
Scatter search (SS)	<i>Lin et al.</i> [2007]
Cross-entropy algorithm (CE)	<i>Perelman and Ostfeld</i> [2007]
Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm	<i>Tolson et al.</i> [2009]
Differential evolution (DE)	<i>Suribabu</i> [2010]
Honey-Bee Mating Optimization (HB)	<i>Mohan and Babu</i> [2010]
Genetic Heritage Evolution by Stochastic Transmission (GHEST)	<i>Bolognesi et al.</i> [2010]

¹Only the first significant paper for each optimization technique applied to WDS optimization is provided.

Much research has been done in an attempt to improve the efficiency of EAs applied to large WDS optimization problems [Bolognesi *et al.* 2010]. Decomposing the original WDS using graph theory to facilitate the optimization process is one of these research lines.

2. Decomposition of WDSs

Normally, decomposition of a water network is used to carry out an analysis of network connectivity, reliability and management strategies. Ostfeld [2005] employed graph theory to undertake a connectivity analysis for WDSs. Deuerlein [2008] decomposed complex water networks into forests, blocks and bridges using graph theory. Based on the decomposition algorithm proposed by Deuerlein [2008], the original whole network can be simplified to several parts that are able to improve the understanding of the interaction among different network components, thereby enabling a network vulnerability analysis and improved management of the network. Yazdani and Jeffrey [2010] used graph theory and complex network principles to conduct a robustness analysis for WDSs while Di Nardo and Di Natle [2010] proposed a design support methodology for district metering of WDSs using graph decomposition.

Few attempts have been made to utilize graph decomposition to facilitate the WDS design optimization. Krapivka and Ostfeld [2009] proposed a network decomposition based GA-LP scheme for the least-cost pipe sizing of WDSs. In their work, the looped water network was first decomposed into a number of spanning trees and chords. Then an LP was utilized to optimize each spanning tree, allowing the identification of the least-cost spanning tree. Finally a GA was used to alter the flows for the least-cost spanning tree (referred to the “outer” problem) and the LP was employed to optimize the tree network with the updated flows (the “inner” problem).

Cisty [2010] proposed another network decomposition based GA-LP model for solving WDS design problems. In this proposed GA-LP method, a GA was used to

generate various trees for a complex looped network, and LP was used to optimize each tree network. *Haghighi et al.* [2011] developed a hybrid model incorporating a GA and integer linear programming (GA-ILP) to optimize the design of WDSs. As for the GA-LP method proposed by *Cisty* [2010], the GA in the GA-ILP model proposed by *Haghighi et al.* [2011] randomly generated tree networks for the original looped WDS and the ILP was utilized to optimize each tree network.

Zheng et al. [2011a] proposed a combined NLP-DE method for optimizing WDS design. In the proposed NLP-DE approach, the original WDS was decomposed into a shortest-distance tree and chords. Then an NLP was employed to arrive at an approximate optimal solution for the decomposed WDS. The approximate optimal solution obtained from the NLP was then used to seed a DE in order to generate improved quality solutions for the original full WDS.

3. The proposed decomposition and multi-stage optimization method

The above analysis indicates that graph theory is normally used to find various trees for the looped WDS in previously proposed decomposition based optimization methods. This is motivated by the fact that optimal solutions for trees can be obtained by deterministic optimization methods such as LP, NLP or ILP with great efficiency. In contrast in this paper, a novel decomposition method is proposed to alternatively decompose the original complex WDS into sub-networks rather than into trees in order to facilitate network design optimization.

For a real-world WDS, multiple sources of supply (that is - multiple tanks) are normally incorporated into the system in addition to having loops in order to improve the reliability of supply. For such a complex WDS with multiple supply sources (WDS-MSS), existing optimization algorithms normally tackle the system as a whole in order to find optimal design solutions. Normally, design of a large scale water network with multiple sources is computationally very rigorous. This is due to the size of the search space as well as the time for hydraulic simulation of the network.

The method proposed here has (i) developed a graph decomposition method to partition the larger optimization problems into smaller ones that in turn reduces the computational overhead for optimizing the design of the WDS-MSS, and has (ii) developed a multi-stage DE method to optimize the design of the sub-networks obtained by decomposing the WDS-MSS and then the original whole network. The outcome is a significantly more efficient and effective method for the optimization of the design of water networks with multiple sources.

In the proposed decomposition and multi-stage optimization method, an algorithm is developed to identify the optimal source partitioning cut-set for a WDS with K supply sources. By removing the optimal source partitioning cut-set, the whole original WDS is decomposed to K sub-networks. For each sub-network, one and only one supply source is assigned. Each sub-network is then optimized by a DE algorithm independently, which is the first stage of optimization.

The optimal solutions for all sub-networks are then combined to provide an approximate optimal solution for the whole original network. However, this approximate optimal solution needs to be further improved because the pipes within the optimal source partitioning cut-set were not included during the first stage of the sub-network optimization. A second phase DE is therefore used to explore the search space around the obtained approximate optimal solution and better quality solutions for the whole WDS are expected to be found with significantly reduced computational effort. This is the second stage of the optimization process.

The concept of multi-stage optimization is based on the decomposition of large-scale and complex systems into independent subsystems (although these sub-systems are actually interconnected and are not truly independent of one another). Each subsystem is optimized independently, and the optimal solutions for each subsystem are then combined together to derive the optimal solution for the whole system. Although a multi-stage optimization approach has been used to control the pollution of water resource systems [*Hass* 1970, *Haines* 1971], optimize urban water

management [Zhu *et al.* 2005] and deal with the reservoir operation problem [Canon *et al.* 2009], the method proposed here is the first time that multi-stage optimization method has been used to optimize the design of a WDS.

Although the DE algorithm is used in this study, other EAs such as a GA could also be implemented in the proposed optimization framework. However, the performance comparison of the DE algorithm with other optimization algorithms has not been carried out in this study. The methodology of the proposed decomposition and multi-stage method are given later.

4. Formulation of the WDS-MSS optimization problem

Typically, single-objective optimization of a WDS is the minimization of system total life cycle costs (pipes, tanks and other components) while satisfying head constraints at each node. In this paper, the proposed decomposition and multi-stage optimization method is verified using WDS-MSS case studies with pipes only for a single demand load case. Thus, the formulation of the WDS-MSS optimization problem can be given by:

$$\text{Minimize} \quad F = a \sum_{i=1}^{np} D_i^b L_i \quad (1)$$

Subject to:

$$H_k^{\min} \leq H_k \leq H_k^{\max} \quad k = 1, 2, \dots, nj \quad (2)$$

$$G(H_k, D) = 0 \quad (3)$$

$$D_i \in \{A\} \quad (4)$$

where F =network cost that is to be minimized [Simpson *et al.* 1994]; D_i =diameter of the pipe i ; L_i =length of the pipe i ; a , b =specified coefficients for the cost function; np =total number of pipes in the network; nj =total number of nodes in the network; $G(H_k, D)$ =nodal mass balance and loop (path) energy balance equations for the whole network,

which is solved by a hydraulic simulation package (EPANET2.0 in this study); H_k =head at the node $k=1,2,\dots,nj$; H_k^{\min} and H_k^{\max} are the minimum and maximum allowable head limits at the nodes; and A = a set of commercially available pipe diameters.

5. Methodology of the proposed method

The flowchart in Figure 1 outlines the features of each step of the proposed decomposition and multi-stage optimization approach.

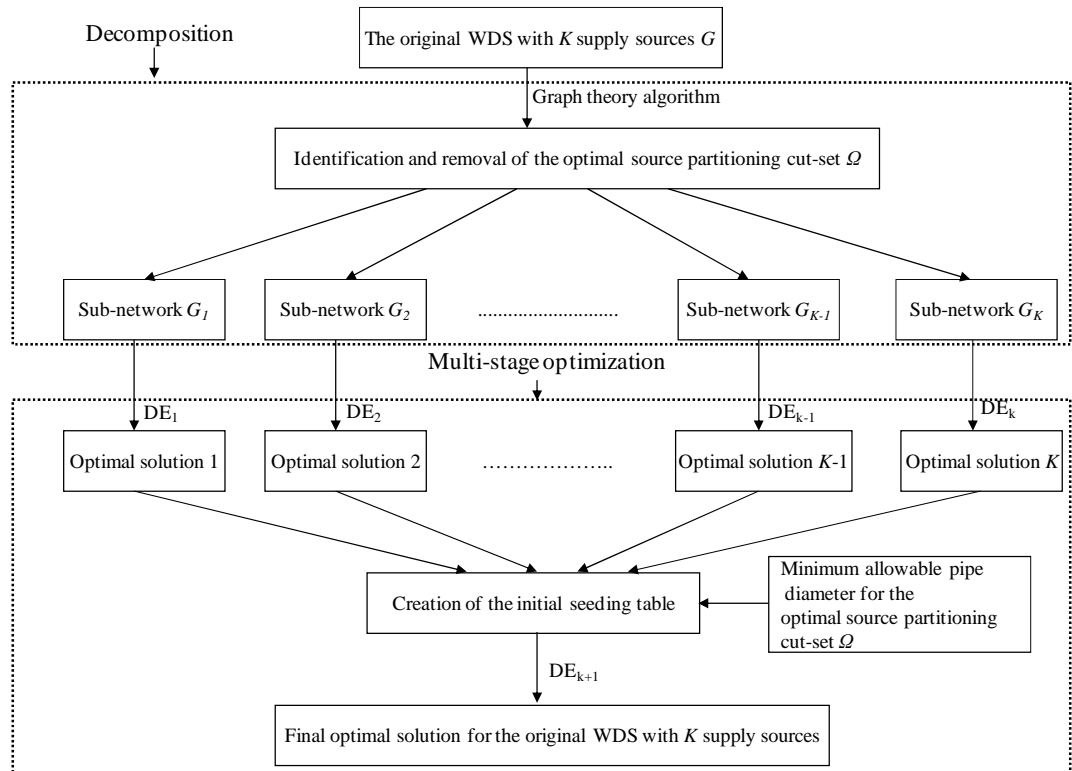


Figure 1 Flowchart of the proposed optimization approach

5.1. Decomposition of the WDS-MSS

5.1.1. Source partitioning cut-set of the WDS-MSS

In a connected graph $G(V,E)$, a cut-set is a set of edges whose removal from G results in G being disconnected [Deo 1974], where V is a set of vertices and E is a set of edges. In this paper, a source partitioning cut-set (C) for a WDS-MSS is a set of pipes whose removal from the system results in the WDS-MSS being separated in such a

way that each sub-network is attached to one and only one unique supply source. That is, the original WDS with K supply sources is decomposed into K disconnected sub-networks after removal of the source partitioning cut-set. For a WDS-MSS with two supply sources (reservoirs) (see Figure 2(a)), all source partitioning cut-sets (C) and their corresponding two sub-networks after removal of the cut-set are given in Figure 2 (b, c, d).

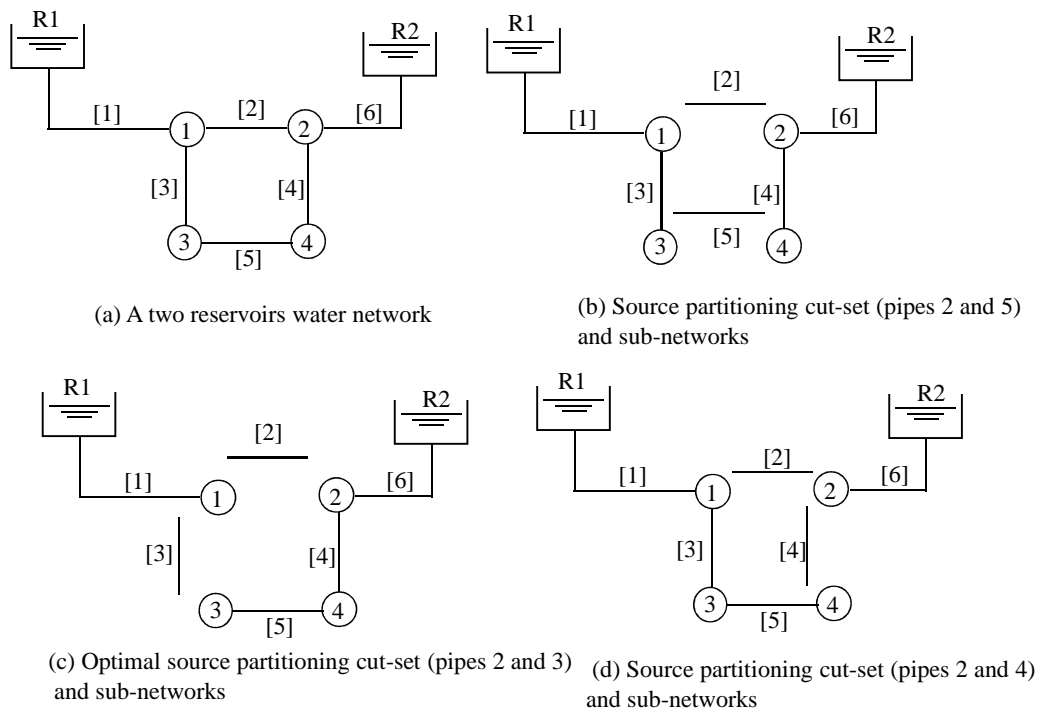


Figure 2 An example of cut-sets and the sub-networks for a two reservoir WDS (a: the two reservoirs water network; (b): Source partitioning cut-set (pipes 2 and 5) and sub-networks, (c): Optimal source partitioning cut-set (pipes 2 and 3) and sub-networks and (d): Source partitioning cut-set (pipes (2 and 4) and sub-networks))

As shown in Figure 2 (a), the original WDS $G(V,E)$, where $V=\{R1, R2, 1, 2, 3, 4\}$ and $E=\{1, 2, 3, 4, 5, 6\}$, has two reservoirs (R1 and R2), 6 links and 4 nodes. An arbitrarily selected source partitioning cut-set $C=\{2, 5\}$ is shown in Figure 2 (b). The original two-reservoir WDS is decomposed to two sub-networks $G_1(V_1,E_1)$, $G_2(V_2,E_2)$ after removal of the cut-set $C=\{2, 5\}$, where $V_1=\{R1, 1, 3\}$, $E_1=\{1, 3\}$, $V_2=\{R2, 2, 4\}$, $E_2=\{4, 6\}$. It can be observed that a total of three cut-sets exist in this two-reservoir

WDS, which enable the network disconnection. In the proposed decomposition and multi-stage optimization method, an optimal source partitioning cut-set Ω is proposed to decompose the WDS-MSS. The definition of the Ω and the algorithm that has been developed in this study to identify the Ω for a WDS-MSS are outlined in the next section.

5.1.2. Identification of the optimal source partitioning cut-set Ω of the WDS-MSS

For a WDS-MSS with K supply sources, each node i in the water network has K different potential water supply sources and a number of potential supply paths from each supply source. For a given supply source k and the demand i , there exists a finite set of independent paths joining these two nodes, symbolized here as P_{ki} . For each supply path $\lambda \in P_{ki}$, the available friction slope is calculated as:

$$S_{ki}(\lambda) = \frac{H_k - H_i^{\min}}{\sum_{l \in \lambda} L_l} \quad (5)$$

where $S_{ki}(\lambda)$ is the available friction slope from source k to node i based on the supply path $\lambda \in P_{ki}$, H_k is the head of the source k and H_i^{\min} is the minimum allowable head requirement at node i ; L_l is the length of link l ($l \in \lambda$). Amongst the different paths $\lambda \in P_{ki}$, the path that has the *largest* available friction slope (λ_{ki}^*) is considered to be the most economic supply path for this node i from source k [Zheng et al. 2011a], which is given as:

$$\lambda_{ki}^* = \arg \max_{\lambda \in P_{ki}} S_{ki}(\lambda) \quad (6)$$

Then for a given node i , the available friction slope for the economic paths from each source can be constructed to form the set $\xi_i = \{S_{1i}(\lambda_{1i}^*), S_{2i}(\lambda_{2i}^*) \cdots, S_{Ki}(\lambda_{Ki}^*)\}$. Given this, the source k with the greatest available friction slope λ_{ki}^* for node i is taken to be the supply source for node i . This is based on heuristic reasoning that it is economical overall for a demand node to receive flows from a supply source having a relatively high available head and/or a relatively short distance to this demand node. As such,

each node i is assumed to receive flows from one and only one supply source in the proposed method according to this heuristic approximation.

By assigning demand nodes to different supply source nodes, a demand node set N_k can be constructed for each supply source node k , which consists of all nodes for which k is the supply source. Then links that have different supply sources for two nodes on each side are obtained, which is defined as:

$$\Omega = \{(i, j) : i \in N_k, j \in N_m, k \neq m, k, m = 1, \dots, K\} \quad (7)$$

where (i, j) is the link having node i and j on each side. This set of links is defined as the optimal source partitioning cut-set Ω for the WDS-MSS and the removal of the optimal cut-set leaves the original WDS-MSS decomposed into several sub-networks. Each sub-network is composed of one and only one supply source and a particular number of nodes and pipes. Each supply source only provides water to specific nodes established when the optimal source partitioning cut-set is removed. Thus, the optimal source partitioning cut-set is actually the estimated optimal supply boundary of different supply sources in a WDS.

The two-reservoir WDS presented in Figure 2(a) is used to explain the proposed Ω to decompose the network. The data, including the length of each link, the elevation of each node and reservoir, and the minimum head requirement for each node is given in Table 2.

Table 2 Network data of the WDS with two reservoirs

Nodes	Elevation (m)	Head requirement (m)	Links	Length (m)
R1	54	-	5	550
R2	56	-	6	400
1	27	20	1	800
2	29	20	2	800
3	31	20	3	650
4	33	20	4	700

Each supply path for each node (λ) and the available friction slope for each path ($S(\lambda)$) are provided in Table 3. The path having the largest available friction slope has been highlighted for each node in Table 3.

Table 3 Supplying paths and the available friction slope for each node

Nodes (i)	Path (λ)	Length (m)	Available head ($H_k - H_i^{\min}$) (m)	Available friction slope ($S(\lambda)$)
1	R1-1	800	7	0.0088
	R2-6-2	1200	9	0.0075
	R2-6-4-5-3	2300	9	0.0039
2	R1-1-2	1600	5	0.0031
	R1-1-3-5-4	2700	5	0.0019
	R2-6	400	7	0.0175
3	R1-1-3	1450	3	0.0021
	R2-6-2-3	1850	5	0.0027
	R2-6-4-5	1650	5	0.0030
4	R1-1-2-4	2000	1	0.0005
	R1-1-3-5	2300	1	0.0004
	R2-6-4	1100	3	0.0027

As shown in Table 3, $N_{R1}=\{1\}$ as λ_{R1-1} is the most economical path that has the largest available friction slope for node 1. $N_{R2}=\{2, 3, 4\}$ as these nodes have the largest available friction slopes from R2. Thus the optimal source partitioning cut set is given as $\Omega=\{2, 3\}$ as the nodes on each side of these two links are assigned to different reservoirs. The optimal partitioning cut-set Ω and the sub-networks after removal of the Ω are given in Figure 2 (c).

For a relatively small WDS-MSS, the Ω can be determined using complete enumeration. However, it is impossible to enumerate all the paths for a relatively large WDS-MSS. An algorithm that is used to efficiently identify the optimal source partitioning cut-set Ω for a large WDS-MSS has been developed in this research. The proposed approach is motivated by the fact that the shortest-distance path P_{ki}^* of all the available paths from the same supply source to a particular node always has the largest available friction slope λ_{ki}^* . This is reflected in Equation (5), which shows that the available head for a particular node to a particular supply source is constant. Therefore, the shortest path between a node and a particular supply source has the largest available friction slope, *i.e.*, $P_{ki}^* = \lambda_{ki}^*$. The Dijkstra algorithm [Deo 1974] is employed in this study to find the shortest-distance path for each node to different supply sources. The details of Dijkstra algorithm [Deo 1974] are given as follows.

In the Dijkstra algorithm, either a permanent label or temporary label is assigned to each node. A permanent label is given to a node once the shortest path from this node to source node has been determined. The value of the permanent label is made equal to the sum of lengths of the shortest path. In contrast, a temporary label is given to a node for which the shortest path has not yet been identified. The value of this temporary label is set to be equal to the sum of lengths of the shortest path in the current iteration and this value will be updated in later iterations.

The Dijkstra algorithm begins by assigning a permanent label 0 to the starting node (supply source node) and a temporary label ∞ (this is replaced by a large number in the computer algorithm) to the remaining nodes (demand nodes in a WDS-MSS). In the search procedure, at each iteration, another node gets a permanent label according to the following rules [Deo 1974]:

Rule 1. Every node j that has not yet permanently been labeled is updated with a new temporary label whose value is given by

$$\min [\text{old label } j, \text{old label } i + d_{ij}]$$

where i is the latest node permanently labeled in the previous iteration. d_{ij} is the direct length from node i to node j . If nodes i and j are not directly connected, then $d_{ij}=\infty$.

Rule 2. At each iteration, the smallest value amongst all temporary labels is found and the corresponding node is permanently labeled with this value. Thus a new permanently labeled node is produced in this iteration. If more than one temporary label has the same value, then any one of the candidates for permanent labelling is selected.

Rules 1 and 2 are repeated until all the nodes are permanently labeled. An example illustration of the Dijkstra algorithm performed for $R1$ to other demand nodes in the looped water network of Figure 2(a) is given in Table 4. The shortest-distance path for source node $R1$ to other demand nodes is presented in the last column of Table 4.

Table 4 The Dijkstra algorithm for identifying the shortest-distance tree

iteration	Length to Node*					Description	Shortest path P_{ki}^*
	R1	1	2	3	4		
1	0	∞	∞	∞	∞	Starting at the source node $R1$. It is labeled 0 and all the other nodes are labeled ∞ .	$R1-R1$
2	0	800	∞	∞	∞	All successors of $R1$ are labeled using Rule 1. The smallest label (node 1) is permanently labeled (Rule 2).	$1-R1$
3	0	800	1600	1450	∞	All successors of 1 are labeled using Rule 1. The smallest label (node 3) is permanently labeled (Rule 2).	$3-1-R1$
4	0	800	1600	1450	2000	All successors of 3 are labeled using Rule 1. The smallest label (node 2) is permanently labeled (Rule 2).	$2-1-R1$
5	0	800	1600	1450	2000	All successors of 2 are labeled using Rule 1. The smallest label (node 4) is permanently labeled (Rule 2).	$4-3-1-R1$

*The bold values are the succession of assignment of permanent labels. The value of ∞ would be designated as a large number in a computer implementation.

The details of the proposed algorithm to identify the optimal source partitioning cut-set Ω for a WDS with K supply sources are given in Figure 3. As can be seen from Figure 3, three steps are involved in this proposed algorithm to identify the optimal source partitioning cut-set Ω . In Step 1, the Dijkstra algorithm is performed to identify the shortest-distance path $P_{ki}^* = \lambda_{ki}^*$ for each supply source node k to each node i within the WDS. Then the available friction slope for the shortest distance path $S_{ki}(\lambda_{ki}^*)$ is computed using Equation (5). As such, a total of K different $S_{ki}(\lambda_{ki}^*)$ values are obtained for each node i . In Step 2, node $i=1, \dots, n$ is assigned to the set N_k if $S_{ki}(\lambda_{ki}^*)$ is the largest value from the K total available friction slope values, indicating that k is the supply source node for node i . In Step 3, all the links (i, j) that have the nodes on each side assigned to different supply source nodes are identified and form the optimal source partitioning cut-set Ω .

It is observed from Figure 3 that the Dijkstra algorithm is performed K times to determine the optimal source partitioning cut-set for a WDS with K supply sources. The computational time required to identify the optimal source partitioning cut-set for each WDS-MSS case study is analyzed in later discussion. The sub-networks are

obtained after removal of the optimal source partitioning cut-set. These sub-networks are independent and can be optimized separately.

For a WDS with graph $G(V, E)$, with K supply sources, n nodes and nl links.

Step 1:

FOR $k=1, \dots, K$

Perform the Dijkstra algorithm for the supply source k to identify the shortest-distance path $P_{ki}^* = \lambda_{ki}^*$ to each node $i=1, \dots, n$ as illustrated in Table 4.

Compute the $S_{ki}(\lambda_{ki}^*)$ for each node i using Equation (5).

END FOR

Step 2:

FOR $i=1, \dots, n$

Select k such that $S_{ki}(\lambda_{ki}^*) > S_{ji}(\lambda_{ji}^*)$ for all $j=1, \dots, K, j \neq k$ using Equation (6)

Node i is assigned to set N_k for which k is the supply source for node i .

END FOR

Step 3:

FOR all $(i, j) \in E$

IF $i \in N_k, j \in N_m, k \neq m, k, m = 1, \dots, K$, i.e., the link with nodes at either end assigned to different sources.

Link (i, j) is assigned to the source partitioning cut-set Ω (Equation (7))

END IF

END FOR

Figure 3 Optimal source partitioning cut-set identification algorithm.

5.1.3. Summary of the proposed decomposed method for WDS-MSS

The proposed decomposition method partitions the whole water distribution system with K supply sources into K sub-networks. This differs significantly to the majority of the previously used decomposition approaches. These previous approaches identified a tree network as an approximation for the original full network [Krapivka and Ostfeld 2009; Kadu et al. 2008; Zheng et al. 2011a]. In the proposed decomposition method, the shortest-distance path only is used to assign the nodes to different supply sources and each node may receive flows via various paths from the

assigned supply source (not only the shortest-distance path). This is due to the fact that loops are retained within each sub-network obtained by the proposed decomposition method. However, in *Krapivka and Ostfeld* [2009], *Kadu et al.* [2008] and *Zheng et al.* [2011a], each node has one and only one path to receive flows to meet the demands from the source node.

The available friction slope for each node is used in the proposed decomposition method to determine the optimal source partitioning cut-set Ω for a WDS-MSS and the magnitude of the demands at each node are not considered during the decomposition. It is assumed to be cost effective overall for a demand node to receive the flows to meet the demands from a source having a relatively large available head and/or the shortest distance to this node. Thus, an approximate supply boundary is produced using the proposed decomposition method since each demand node receives the flows from one and only one supply source. However, it should be acknowledged that the supply boundary obtained by the proposed decomposition is an approximation to that of the real supply system as some nodes (especially nodes at the supply boundary) in the real WDS may receive the flows to supply demands from multiple supply sources.

The available friction slope concept has also been used by *Kadu et al.* [2008] to identify a tree for a looped WDS. Thus, it is necessary to clarify the differences between the method used by *Kadu et al.* [2008] and the approach proposed here in terms of decomposing the WDS. The proposed decomposition method aims to specify a particular supply source for each demand node, for which this supply source has the *largest* available friction slope to this demand node, while *Kadu et al.* [2008] used the *smallest* available friction slope to identify the critical path for the original WDS. In addition, disconnected sub-networks are obtained using the proposed decomposition method, within which loops are involved, while a tree network is finally obtained using the method proposed by *Kadu et al.* [2008].

It is also useful to highlight the difference between the proposed decomposition method and the network aggregation method proposed by *Pereman and Ostfeld* [2008]. The main differences include: (i) in the new method presented here, the whole network is decomposed into several disconnected sub-networks, while the aggregation method keeps the general topology of the original system and only removes some nodes and links from the original system; (ii) in the proposed decomposed method, the decomposition results for a WDS are based on the number of different supply sources, while the aggregation result is dependent on the connectivity properties of the original system (such as the location of the monitor stations); and (iii) the demand distribution and link properties (such as link length and conductance) are not varied in the proposed decomposition approach, while they are changed in the aggregation network of *Pereman and Ostfeld* [2008] in order to resemble the hydraulics and water quality performance of the original system.

5.2. Multi-stage optimization for the WDS-MSS

5.2.1. Differential evolution algorithm applied to each sub-network (first stage optimization)

The differential evolution (DE) algorithm, introduced by *Storn and Price* [1995] has performed well when used to find optimal solutions in a number of numerical optimization case studies [*Vesterstrom and Thomsen* 2004]. *Vasan and Simonovic* [2010] and *Suribabu* [2010] first applied DE to the optimization of WDSs, and concluded that the performance of the algorithms was at least as good as, if not better, than other EAs such as GAs and Ant Colony Optimization. *Zheng et al.* [2011a, 2011b] further investigated the performance of DE algorithms and reported that DE was effective in finding optimal solutions for WDS. Three operators including mutation, crossover and selection operators are involved in the application of DE in an optimization problem. Three parameters need to be pre-specified: the population size (N), mutation weighting factor (F) and the crossover rate (CR). The general ranges of these three parameters are $1D \leq N \leq 10D$ (where D is the number of

decision variables) $0.1 \leq F \leq 1.0$ and $0.1 \leq CR \leq 1.0$ [Storn and Price 1995]. The pseudo-code for the DE algorithm applied to WDS optimization is given in Figure 4.

```

Step 0. Specify following inputs of the differential evolution (DE): the population size ( $N$ ), the crossover rate ( $CR$ ), the mutation weighting factor ( $F$ ), the maximum allowable number of generations ( $MG$ ), and the number of decision variables ( $D$ )
Step 1: Randomly generate  $N$  initial solutions  $X_{i,G} = \{ x_{i,G}^1, x_{i,G}^2, \dots, x_{i,G}^D \}, i=1, \dots, N, G=0$ .
Step 2: Evaluate the objective function of the  $N$  initial solutions  $f(X_{i,G})$ .

Count=1

REPEAT

UNTIL Count  $\geq$   $MG$ 
Step 3: Perform the DE mutation operator to generate  $N$  mutant solutions  $V_{i,G} = \{ v_{i,G}^1, v_{i,G}^2, \dots, v_{i,G}^D \}$ 
FOR  $i=1, \dots, N$ 
 $V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$ , where  $r1 \neq r2 \neq r3$  and they are randomly generated for each  $i$ .
END FOR
Step 4: Perform the DE crossover operator to generate trial solutions  $U_{i,G} = \{ u_{i,G}^1, u_{i,G}^2, \dots, u_{i,G}^D \}$ 
FOR  $i=1, \dots, N$ 
FOR  $j=1, \dots, D$ 
IF  $Rand_j(0,1) \leq CR$ , where  $Rand_j(0,1)$  is a uniformly distributed random number between 0 and 1.
 $u_{i,G}^j = v_{i,G}^j$ 
ELSE
 $u_{i,G}^j = x_{i,G}^j$ 
END IF
END FOR
END FOR
Step 5: Alter the continuous pipe diameter solution to the nearest discrete diameter for each decision variable and then evaluate  $N$  trial solutions  $f(U_{i,G})$ .
Step 6: Select the next generation  $X_{i,G+1} = \{ x_{i,G+1}^1, x_{i,G+1}^2, \dots, x_{i,G+1}^D \} i=1, \dots, N$ 
FOR  $i=1, \dots, N$ 
IF  $f(X_{i,G}) \leq f(U_{i,G})$ 
 $X_{i,G+1} = X_{i,G}$ 
ELSE
 $X_{i,G+1} = U_{i,G}$ 
END IF
END FOR

Count=Count+1

```

Figure 4 Pseudo-code for the differential evolution (DE) algorithm.

The basic DE algorithm is a continuous global optimization search algorithm [Storn and Price 1995], and requires modification when used to solve discrete WDS optimization problems. In this study, the modification made to the DE algorithm was based on the approach used in Suribabu [2010]. To handle the head constraints, constraint tournament selection [Deb 2001] was used in the DE algorithm.

During the first stage of the optimization process, each sub-network is optimized by a separate DE. The sub-network optimization problem formulation is similar to that for the original whole network [Equations (1) to (4)]. Only the pipes (the objective function) and nodes (head constraints) in the sub-network are handled by each individual DE algorithm. Because the dimensionality of each sub-network is significantly reduced compared to the original network, the DE algorithm is expected to be able to more efficiently find optimal solutions for each sub-network than for the whole network.

For the water network given in Figure 2, 14 pipe diameters including {150, 200, 250, 300, 350, 400, 450, 500, 600, 700, 750, 800, 900, 1000} mm can be selected for each pipe and all the pipes are assigned to have an identical Hazen-Williams coefficient of 130. The unit costs for each pipe diameter are given by Kadu *et al.* [2008]. Two separate DEs were employed to optimize the two sub-networks ($S_1=\{R1, 1, [1]\}$, $S_2=\{R2, 2, 3, 4, [4], [5], [6]\}$) as shown in Figure 2(c) obtained by removing the optimal source partitioning cut-set $\Omega=\{2, 3\}$. The DE optimal solutions for S_1 and S_2 were \$37,910 and \$166,896 respectively and the pipe diameters for the optimal solutions are $[1]=250$ mm, $[4]=450$ mm, $[5]=300$ mm and $[6]=500$ mm. It is noted that the optimal cut-set $\Omega=\{2, 3\}$ was not included in the first stage of the proposed multi-stage optimization method.

5.2.2. Creation of the seeding table

In the proposed method, the optimal solutions for K sub-networks are obtained after the first stage optimization, and an optimal pipe diameter is assigned for each link in all sub-networks. As the optimal source partitioning cut-set Ω of the original

complete network is not included during the first stage optimization, the minimum allowable pipe diameters are therefore assigned to all the links in the Ω in this study. Each link of the complete network is given a pipe diameter by combining the optimal solutions of the sub-networks and assigning the minimum allowable pipe diameters for the Ω . This therefore creates an approximate optimal solution (or a near optimal in a topological sense) for the complete network. For the example given in Figure 2, the approximate optimal solutions were \$240,374 and the corresponding network configuration is [1]=250 mm, [2]=150mm, [3]=150 mm, [4]=450 mm, [5]=300 mm and [6]=500 mm (note 150 mm is the minimum allowable pipe diameter).

The approximate optimal solution is now used to create a tailored seeding table to enable the second stage of the optimization. For each link in this seeding table, three pipe diameters are included, namely (i) the pipe diameter from the approximate optimal solution of the whole network, (ii) and the pipe diameters that are immediately smaller, and (iii) the pipe diameters that are immediately larger than the diameter provided by the approximate optimal solution. For a pipe that is already the minimum or maximum allowable diameters, the three adjacent smallest or largest pipe diameters are assigned to the seeding table for this pipe.

Table 5 is used to illustrate the process of the creation of the seeding table based on the approximate optimal solution of the water network given in Figure 2. The pipe diameters of the approximate optimal solution obtained after the first stage optimization are given in column 2 of Table 5. As shown in Table 5, for links 1, 4, 5 and 6, three adjacent pipe diameters are included in the seeding table and the middle one is the pipe diameter for the approximate optimal solution (column 2 of Table 5). For links 2, and 3, three adjacent smallest pipe diameters are assigned to the seeding table as the diameter of links 2 and 3 given in column 2 of Table 5 are already the minimum allowable diameter (150mm). This proposed method for the creation of the seeding table is applied to each case study in this paper.

Table 5 The process for creating the seeding table (applies to any sized network)

Links	Diameters for the approximate optimal solutions (mm)	Link membership	Pipe diameters in the seeding table (mm)	Pipe diameters in the total choice table (mm)
1	400	Belongs to R_1	300, 400, 450	200, 300, 400, 450, 500, 600
2	200	Cut-set	200, 300, 400	200, 300, 400, 450, 500, 600
3	200	Cut-set	200, 300, 400	200, 300, 400, 450, 500, 600
4	450	Belongs to R_2	400, 450, 500	200, 300, 400, 450, 500, 600
5	200	Belongs to R_2	200, 300, 400	200, 300, 400, 450, 500, 600
6	600	Belongs to R_1	450, 500, 600	200, 300, 400, 450, 500, 600

5.2.3. Final optimal solution for the original WDS-MSS (second stage optimization)

In the proposed decomposition and multi-stage optimization method, another DE algorithm (denoted the final DE algorithm) is used in the second stage of optimization in order to find the optimal solutions for the original WDS with multiple supply sources. It is noted that the first stage optimization does not include the pipes in the optimal source partitioning cut-set Ω . In the proposed approach, an approximate optimal solution was generated by combining the sub-network optimal solutions and setting the pipes in the Ω to be the minimum allowable pipe diameters. However, this approximate optimal solution is not acceptable for the original whole network. This is because (i) the network reliability will be reduced by simply assigning the pipes in the Ω to be the minimum allowable diameter size as these pipes are the connections between sub-networks; and (ii) the approximate optimal solution produced in the first stage optimization may be infeasible for the original whole network with the inclusion of the minimum diameter pipes in the Ω . Thus, the approximate optimal solution obtained in the first stage optimization need to be further polished. This is achieved by applying the DE at the second stage optimization of the proposed method.

During the second stage optimization phase [the formulation is given by Equations (1) to (4)], the final DE algorithm is seeded by a tailored seeding table (column 4 of Table 5) rather than the total choice table (14 pipe diameter options). Thus, the initial solutions of the final DE algorithm are randomly located in the search space specified

by the tailored seeding table rather than the whole search space. The final DE algorithm therefore focuses on exploring promising regions specified by the tailored seeding table and hence avoids wasting computational effort investigating infeasible or unnecessarily high cost regions within the search space. It is expected therefore that the final DE algorithm is able to locate better quality solutions for the original WDS-MSS with great efficiency and reliability as it has been seeded with good initial estimates [*Grefenstette 1987; Harik and Goldberg 2000*].

The second stage DE was applied to the original full water network as shown in Figure 2 (a) but it is initialized by the seeding table in the column 4 of Table 5. A further better optimal solution with a cost of \$239,034 was obtained after the second stage optimization and this optimal solution was feasible when determined by EPANET2.0.

6. Case studies

The algorithms for identifying the optimal source partitioning cut-set, creating the seeding table and the DE algorithm were all coded in C++ using MinGW Developer Studio 2.05. The program EPANET2.0 [*Rossman 2000*] was used as a network solver in this study. Four case studies have been used to verify the effectiveness of the proposed decomposition and multi-stage optimization approach: two artificial double-reservoir WDSs; a real-world three-reservoir WDS; and a realistic four-reservoir WDS. It should be noted that the water network layout for each case study is drawn at different scales. In addition the cost for each diameter used for each case study is the sum of the pipe material cost and the pipe construction cost.

6.1. Case study 1: Two-reservoir WDS

The layout of the two-reservoir WDS is given in Figure 2 and the network data is included in Table 2. The global optimal solution for this small network was \$239,034 by using the full enumeration approach. In order to investigate the impact of the different decomposition strategies on the final solution, this water network

decomposed by all cut-sets obtained by the full enumeration were optimized by the proposed multi-stage DE method. A DE algorithm (SDE) seeded by the total choice table (14 pipe options) was also applied to this network in order to enable the performance comparison with the proposed approach. Table 6 presents the statistical results of different algorithms. It is noted that the parameters of the DE ($N=30$, $F=CR=0.5$) were fine-tuned. A maximum number of allowable evaluations was set to be 6,000 for this case study.

Table 6 Algorithm performance for the two-reservoir WDS ($F=CR=0.5$)

Methods	Number of trial runs	Best solution found (\$)	Percentage of trials with best solution found (%)	Average number of equivalent full two-reservoir WDS evaluations to find best solution
CS1 Proposed cut-set based on friction slope method with $\Omega=\{2, 3\}$	100	239,034	100	376
CS2 Alternative cut-set 1 with $C_1=\{2, 4\}$	100	239,034	54	1,568
CS3 Alternative cut-set 2 with $C_2=\{2, 5\}$	100	239,034	14	658
- SDE	100	239,034	98	792

As shown in Table 6, for this small network, all the algorithms are able to find the global optimal solution with a cost of \$239,034. The proposed multi-stage DE method with $\Omega=\{2, 3\}$ (denoted as CS1) significantly outperformed the proposed multi-stage DE but with the cut-sets $C_1=\{2, 4\}$ (CS2) and $C_2=\{2, 5\}$ (CS3) in terms of the solution quality and the efficiency. This is proven by the fact that CS1 found the global optimal solution with a success rate of 100%, which is significantly higher than CS2 (54%) and CS3 (14%). In addition, the proposed multi-stage DE method with $\Omega=\{2, 3\}$ performed slightly better than the SDE in terms of the percent with the best solution found.

The computational overhead for a hydraulic evaluation of one sub-network with EPANET 2.0 is different from the computational effort required to evaluate the original whole network because of the smaller size of the sub-network. In order to enable a fair comparison, the computational overhead for the evaluation of each sub-network has been converted to the equivalent number of evaluations for the whole

network. Each sub-network and the full network were run 1000 times with randomly selected pipe configurations using the code developed for this proposed method. Then the average computational time for one sub-network simulation was converted to the equivalent number of corresponding full network simulations. This approach has been used for each case study investigated in this paper. The code was developed in C++ (linked to EPANET2.0 through the Toolkit) and run on a Pentium PC (Inter R) at 3.0 GHz.

In terms of comparing the efficiency, CS1 performed the best as it only required an average of 376 equivalent full network evaluations to find the optimal solutions. This is only 24%, 57%, and 47% of those required by CS2, CS3, and SDE respectively.

6.2. Case study 2: Double-reservoir WDS

The double-reservoir network (DRN) was first presented by *Kadu et al.* [2008]. The DRN consists of 24 demand nodes, 34 pipes and 9 loops, and is fed by two reservoirs with 100 and 95 meters of fixed head respectively. The layout of the DRN is given in Figure 4.

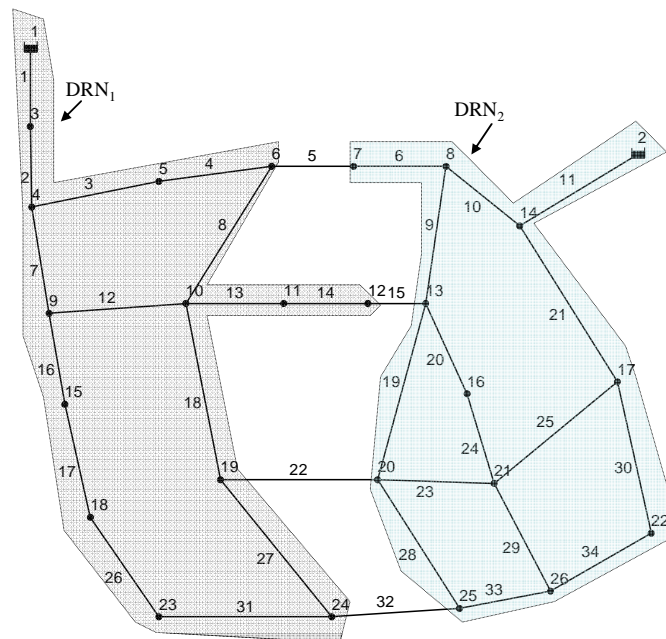


Figure 5 Layout, the optimal source partitioning cut-set (Ω) and the sub-networks (DRN_1 and DRN_2) of the two-reservoir network (DRN)

A total of 14 pipe diameters are available in the DRN case study and hence the total choice table includes 14 pipe diameters for each pipe. The search space is therefore $14^{34} \approx 9.2972 \times 10^{38}$. Details of this network and the cost of the pipes are given by *Kadu et al.* [2008]. The optimal source partitioning cut-set for the DRN identified through the developed graph decomposition approach (Figure 5), included pipes 5, 15, 22 and 32 ($\Omega = \{5, 15, 22, 32\}$). The original DRN was therefore partitioned into two sub-networks (as shown in Figure 5): sub-network one (DRN_1) and sub-network two (DRN_2). DRN_1 included reservoir 1, 13 nodes and 15 pipes on the left side of the optimal source partitioning cut-set. DRN_2 was composed of reservoir 2, with 11 nodes and 15 pipes on the right side of the optimal source partitioning cut-set.

In order to enable a performance comparison, the runs of the standard DE algorithm (SDE) seeded by the total choice table (14 pipe diameters) with different starting random number seeds were also conducted for the DRN case study. Table 7 provides the parameter values used for the DE algorithm applied to the DRN case study. As shown in Table 7, a population size (N) of 50 and a maximum number of allowable evaluations of 30,000 were used for the DE applied to sub-networks DRN_1 and DRN_2 (the first stage optimization of the proposed method). For the DE algorithm used in the second stage optimization phase and the SDE applied to the original whole DRN, a population size of 100 and a maximum number of allowable evaluations of 400,000 were used. Values of $F=0.6$ and $CR=0.5$ were utilized for all DE used in the proposed method and the SDE applied to the DRN case study. These values were selected based on trials of a number of different parameter values.

Table 7 The DE algorithm parameter values applied to different sub-networks and the whole DRN ($F=0.6$, $CR=0.5$)

Network	No. of decision variables (Pipes)	Population size (N)	Maximum number of allowable evaluations
DRN_1	15	50	30,000
DRN_2	15	50	30,000
DRN (the second phase DE algorithm)	34	100	400,000
DRN (the SDE)	34	100	400,000

A total of 100 runs of the proposed method with different starting random number seeds were performed for the DRN case study. A typical run of the proposed method is illustrated in Table 8.

As shown in Table 8, DRN_1 and DRN_2 were optimized by DE algorithm during the first optimization stage of the proposed method and hence optimal solutions with costs of \$1.405 million and \$1.191 million were obtained for DRN_1 and DRN_2 respectively (see columns 2 and 3 of Table 8). By assigning the optimal source partitioning cut-set with the minimum allowable pipe diameters (150 mm for the DRN case study), an approximate optimal solution was produced for the original full DRN with a cost of \$2.752 million, which is given in the column 4 of Table 8. A seeding table was constituted based on the obtained approximate optimal solution (column 5 of Table 8) and this seeding table was used to initialize the DE for the second stage optimization of the proposed method.

The final solution yielded by the proposed method after the second phase optimization was \$2.750 million (column 6 of Table 8), which is lower than the approximate optimal solution obtained after the first optimization stage. It should be highlighted here that the approximate optimal solution with a cost of \$2.752 million was slightly infeasible as determined by EPANET2.0 with the maximum head deficit of 0.5 meters. This is because that (i) the water flow distribution was slightly changed after combining the sub-networks; and (ii) the optimal source partitioning cut-set was simply assigned the minimum allowable pipe diameters.

However, this slightly infeasible solution was located at the vicinity of the final optimal solution. This is reflected by the fact that 28 out of a total of 34 pipes had the same diameters for the approximate optimal solution and the final optimal solution (as shown in Table 8). In addition, the pipe diameters for each link of the final optimal solution are located in the seeding table that was created based on the approximate optimal solution.

Table 8 A typical run of the proposed method for DRN case study

Links	Sub-network optimization results (the first stage optimization) (mm)		Approximately optimal solution (mm) ^c	Creation of choice table	Final optimization results (the second stage optimization) (mm) ^c
	DRN ₁	DRN ₂	DRN ₁ +DRN ₂ + cut-set pipes	-	DRN
1	1000	-	1000	800, 900, 1000	900
2	900	-	900	800, 900, 1000	900
3	350	-	350	300, 350,400	350
4	300	-	300	250, 300, 350	300
5 ^a		-	150	150, 200, 250	150
6		250	250	200, 250, 300	250
7	800	-	800	750, 800,900	800
8	150	-	150	150, 200, 250	150
9		450	450	400, 450, 500	450
10		500	500	450, 500, 600	500
11		800	800	750, 800, 900	750
12	700	-	700	600, 700, 750	700
13	500	-	500	450, 500, 600	500
14	450	-	450	400, 450, 500	500
15 ^a		-	150	150, 200, 250	150
16	450	-	450	400, 450, 500	500
17	350	-	350	300, 350,400	350
18	400	-	400	350, 400, 450	400
19		150	150	150, 200, 250	150
20		150	150	150, 200, 250	150
21		700	700	600, 700, 750	700
22 ^a		-	150	150, 200, 250	150
23		450	450	400, 450, 500	450
24		350	350	300, 350,400	350
25		700	700	600, 700, 750	700
26	200	-	200	150, 200, 250	250
27	300	-	300	250, 300, 350	250
28		300	300	250, 300, 350	300
29		200	200	150, 200, 250	200
30		300	300	250, 300, 350	300
31	150	-	150	150, 200, 250	150
32 ^a		-	150	150, 200, 250	150
33		150	150	150, 200, 250	150
34		150	150	150, 200, 250	150
Cost (\$ million)	1.405	1.191	2.752 ^b	-	2.750
Minimum pressure surplus (m) and its corresponding node	0.08 (Node 23)	0.42 (Node 20)	-0.50 (Node 23)	-	0.15 (Node 12)

^aOptimal source partitioning cut-set pipes for the DRN. ^bInfeasible solution. ^cThe cost of the solution is the sum of the unit cost for each selected pipe multiplied by the length of this pipe.

The statistical results of the proposed method, the SDE and other previously reported feasible solutions (determined by EPANET2.0) for the DRN case study are given in Table 9.

Table 9 Algorithm performance for the DRN case study

Row	Algorithm	Number of trial runs	Best solution found (\$M)	Percentage of trials with best solution found (%)	Average cost solution (\$M)	Average number of original evaluations to find best solution	Average number of equivalent full DRN evaluations to find best solution	
1	Proposed method using Ω (This study)	DRN ₁	100	1.405	85	1.410	10,765	2,702
2		DRN ₂	100	1.191	80	1.206	7,955	2,991
3		DRN ₁ +DRN ₂ +cut-set pipes ^a	100	2.752 ^b	80	2.772	18,720	5,693
4		DRN	100	2.750^f	75	2.755	66,740	66,740
5		Total	100	-	-	-	-	72,433^c
6	SDE (This study)	100	2.750	32	2.762	201,457	201,457	
7	GA [Kadu <i>et al.</i> 2008]	10	2.847	0	NA	NA	NA	
8	GA-ILP [Haghighi <i>et al.</i> 2011]	NA	2.839	0	NA	NA	NA	
9	Proposed method using C ₁ ^d	100	2.898	0	2.901	-	78,965	
10	Proposed method using C ₂ ^e	100	2.755	0	2.783	-	156,620	

^aThe cost of the cut-set pipes is \$0.156 million by assigning them with the minimum pipe diameters (150 mm).

^bInfeasible solution determined by EPANET2.0 with the maximum head deficit of 0.5 meters.

^cThe total computational overhead required by the proposed method has been converted to the equivalent number of the whole network evaluations (DRN₁+DRN₂+DRN₃+cut-set+DRN).

^dThe proposed method applied to the DRN decomposed by cut-set C₁={4, 12, 31}.

^eThe proposed method applied to the DRN decomposed by cut-set C₂={6, 15, 19, 23, 33}.

^fThe best solution based on the new method proposed in this paper.

In this study, a new best solution (feasible when verified by EPANET2.0) was produced at a cost of \$2.750 million. Kadu *et al.* [2008] and Haghighi *et al.* [2011] found the previous best solutions for this case study with costs of \$2.847 million and \$2.839 million respectively. The new best known solution with a cost of \$2.750 million was found with a success rate of 75% by the proposed method, whereas the SDE only returned a success rate of 32%.

As shown in Table 9, the current best solutions for DRN₁ and DRN₂ found by the first stage optimization of the proposed method were \$1.405 million and \$1.191

million. These two optimal solutions for DRN_1 and DRN_2 were found with success rates of 85% and 80% respectively. The approximate optimal solutions for the original whole DRN were obtained by combining the optimal solutions for both sub-network and assigning the minimum pipe diameters for the optimal source partitioning cut-set. As can be seen from Table 9, the best approximate optimal solution provided after the first optimization stage was \$2.752 million and this solution was found with a success rate of 80%.

The average computational time of one evaluation for the DRN_1 and DRN_2 was equivalent to 0.251 and 0.376 evaluations for the whole DRN network respectively. Since the original average number of evaluations for DRN_1 and DRN_2 during the first stage optimization were 10,756 and 7,955 (column 7 of Table 9), the equivalent number of full DRN evaluations was, therefore, 2,702 and 2,991 respectively (column 8 of Table 9).

The computational time required to find the optimal source partitioning cut-set was also converted to the equivalent number of whole network evaluations. For the DRN case study, the computational time required to find the optimal source partitioning cut-set was equivalent to 19 evaluations of the whole DRN network.

As shown in Table 9, the total equivalent average number of evaluations required to find the optimal solutions using the proposed approach was 72,433, which is only 36% of the number of evaluations required by the SDE algorithm. This shows that the proposed method significantly outperforms the SDE algorithm in terms of efficiency. It was observed that the first optimization stage found the approximate optimal solutions that are extremely close to the final best solution (\$2.750 million) using only 5,693 equivalent full DRN evaluations.

A convergence comparison between a DE algorithm seeded with the initial seeding table (the proposed method) and a SDE algorithm is given in Figure 6. It is evident that that the proposed algorithm converges significantly faster than the SDE algorithm.

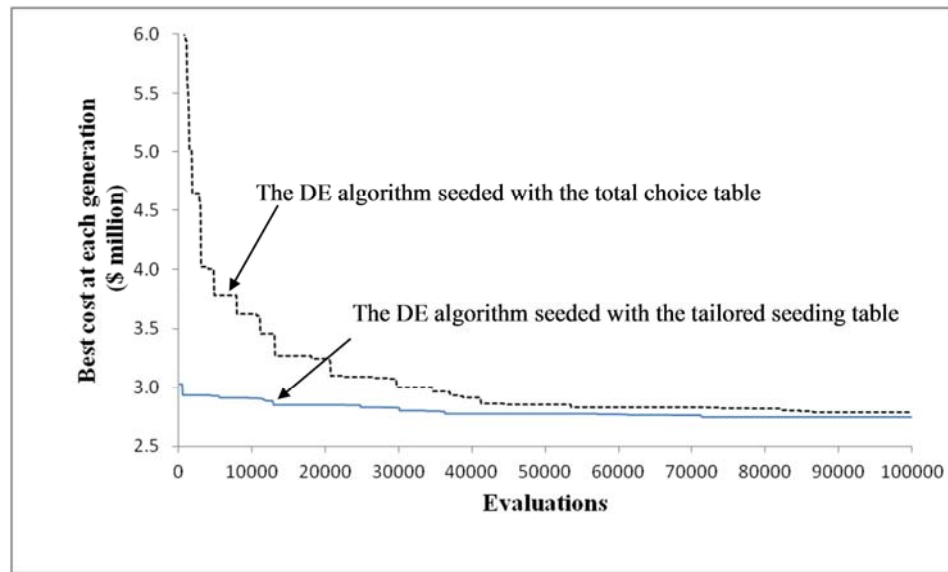


Figure 6 A convergence comparison between DE algorithm seeded with tailored seeding table and the DE algorithm seeded with total choice table

In order to further investigate the impact of the different decomposition strategies on the final solution, the proposed method was also applied to the DRN case study decomposed by $C_1=\{4, 12, 31\}$ and $C_2=\{6, 15, 19, 23, 33\}$ respectively (C is the source partitioning cut-set), and the results are included in Table 9. As shown in Table 9, the best solutions found by the proposed method with decomposition cut-sets $C_1=\{4, 12, 31\}$ and $C_2=\{6, 15, 19, 23, 33\}$ were \$2.898 million and \$2.755 million respectively, which are both larger than the current best known solution of the DRN case study.

In contrast, the proposed method using the optimal source partitioning cut-set $\Omega=\{5, 15, 22, 32\}$ was able to find the current best known solution with a success rate of 75% (see Row 4 of Table 9). In addition, the proposed method with $\Omega=\{5, 15, 22, 32\}$ used fewer average equivalent full DRN evaluations (72,433 in Row 5 of Table 9) to find optimal solutions than the proposed method with $C_1=\{4, 12, 31\}$ (78,965 in Row 9 of Table 9) and $C_2=\{6, 15, 19, 23, 33\}$ (156,620 in Row 10 of Table 9).

Based on the results of case study 1 (Table 6) and case study 2 (Table 9), it can be concluded that (i) the search performance of the proposed method in terms of both

solution quality and efficiency is significantly affected by the decomposition strategy used; and (ii) the proposed optimal source partitioning cut-set Ω , as developed in this paper, is effective in terms of decomposing the water network for design optimization.

6.3. Case study 3: Three-reservoir WDS

The three-reservoir network (TRN) is an actual water network supplied by three reservoirs located in an eastern province of China. This case study is the first time that it has been investigated. The three reservoirs are denoted as R_1 , R_2 and R_3 as shown in Figure 7, and have fixed heads of 44, 45 and 47 meters respectively. The TRN has 287 pipes, 199 demand nodes and 86 primary loops. At each demand node, a minimum pressure of 20 meters is required. All the pipes are assigned to have an identical Hazen-Williams coefficient of 130. The objective of this case study is to determine the least-cost design of this water network, while satisfying the pressure constraints. A total of 14 commercially available pipe diameters ranging from 150 mm up to 1000 mm are available for selection for each pipe (as in case study 1). Thus, the total search space is $14^{287} \approx 8.6845 \times 10^{328}$.

Utilizing the proposed algorithm, 14 links were identified to form the optimal source partitioning cut-set for the TRN case study. Hence, the original TRN was disassembled into three sub-networks, denoted TRN_1 , TRN_2 and TRN_3 as shown in Figure 7. Reservoir 1 (R_1), with 73 demand nodes and 91 pipes, was assigned to TRN_1 . Reservoir 2 (R_2), with 65 demand nodes and 98 pipes, was assigned to TRN_2 . The remaining reservoir (R_3), with 61 demand nodes and 84 pipes was given to TRN_3 . These three sub-networks are shown in Figure 7 in different shades of grey.

The computational time required to identify the optimal source partitioning cut-set for the TRN case study was the equivalent of 15 evaluations of the original TRN (using EPANET 2.0). As for the same method used for the DRN case study, the evaluations of TRN_1 , TRN_2 and TRN_3 were found to be the equivalent of 0.11, 0.10

and 0.091 of the whole TRN evaluation in terms of average computational time based on 1000 runs with randomly selected pipe configuration.

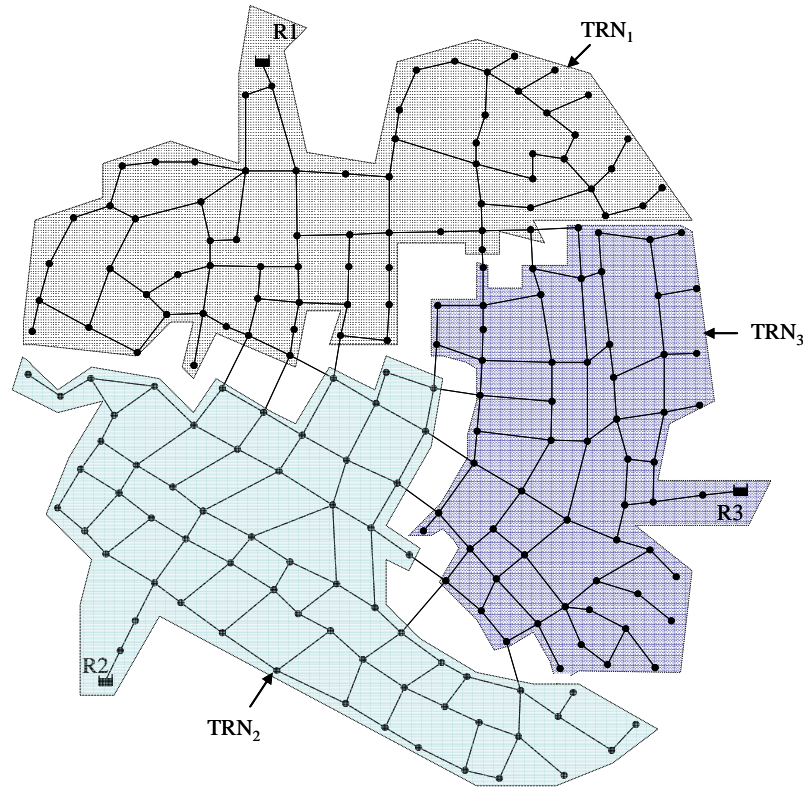


Figure 7 Layout, the optimal source partitioning cut-set and the sub-networks (TRN₁, TRN₂ and TRN₃) of the three-reservoir network (TRN).

For the TRN case study, ten runs of the proposed method and ten SDE algorithm runs with different starting number seeds were performed in order to compare the performance of the two methods. Table 10 provides the parameter values used for the DE algorithm applied to the TRN case study.

As displayed in Table 10, for sub-network optimization, the population size (N) of the DE algorithms was 150 and the maximum number of allowable evaluations used was 150,000. A population size of $N=200$ was used for the DE algorithm in the second stage of the proposed method and two population sizes of $N=200$ and 500 were used for the SDE algorithm. The maximum number of allowable evaluations for DE algorithms applied to optimize the complete TRN (including the SDE and the DE

used in the second phase optimization of the proposed method) was 2.5 million. Values of $F=0.3$ and $CR=0.5$ were used for all DE algorithm runs for this case study based on a few parameter trials.

Table 10 The DE algorithm parameter values applied to different sub-networks and the whole TRN ($F=0.3$, $CR=0.5$)

Network	No. of decision variables (Pipes)	Population size (N)	Maximum number of allowable evaluations
TRN ₁	91	150	150,000
TRN ₂	98	150	150,000
TRN ₃	84	150	150,000
TRN (the second stage DE algorithm)	287	200	2,500,000
TRN (the SDE)	287	200/500	2,500,000

The solution distribution obtained by the proposed method and the SDE algorithm applied to the TRN case study is given in Figure 8. It should be noted that the number of evaluations of the proposed method shown in Figure 8 has been converted to the equivalent number of evaluations for the complete TRN using the same approach as for the DRN case study.

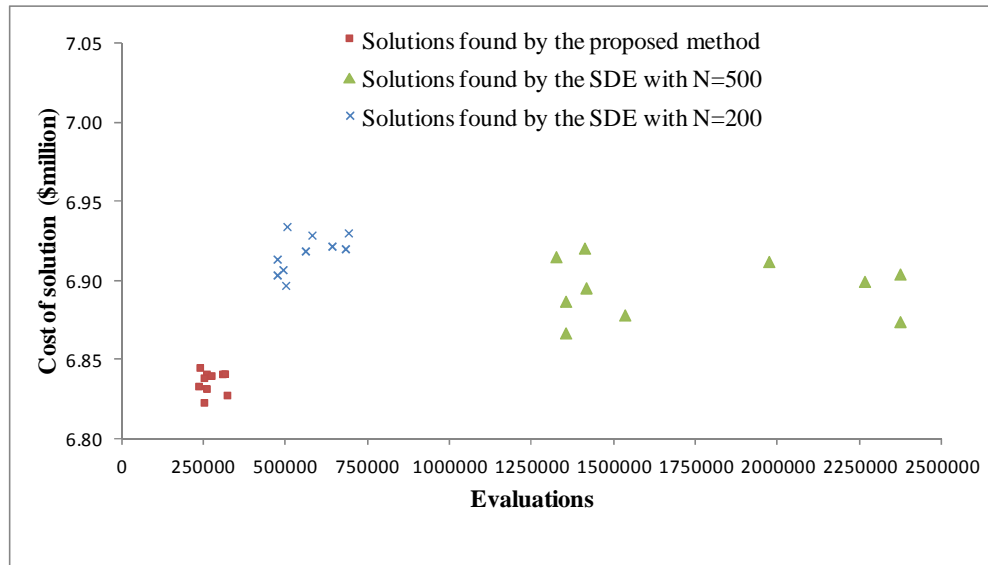


Figure 8 Solution distributions of proposed method and the SDE applied to the TRN case study.

As can be seen from Figure 8, the proposed method exhibits superior performance when compared with the SDE algorithm in term of solution quality and efficiency.

The SDE algorithm with $N=500$ was able to find better quality solutions than the SDE algorithm with $N=200$, but at expense of significantly more evaluations. The final solutions found by the SDE algorithm trial runs with different starting random number seeds are more scattered in distribution than those found by the proposed method. This demonstrates that the performance of the proposed method is less sensitive to the randomized starting points of the search. The statistical results for this case study are shown in Table 11.

Table 11 Algorithm performance for the TRN case study

Algorithm		Number of trial runs	Best solution found (\$M)	Percentage of trials with best solution found (%)	Average cost solution (\$M)	Average number of original evaluations to find best solution	Average number of equivalent full TRN evaluations to find best solution
Proposed method (This study)	TRN ₁	10	2.311	10	2.322	101,190	11,131
	TRN ₂	10	2.291	10	2.294	76,535	7,654
	TRN ₃	10	2.050	10	2.058	61,820	5,626
	TRN ₁ +TRN ₂ +TRN ₃ +cut-set pipes ^a	10	6.874	10	6.883	239,545	24,411
	TRN	10	6.823	10	6.844	245,760	245,760
	Total	10	-	-	-	-	270,171^c
SDE ($N=500$, this study)		10	6.874	0	6.904	1,737,300	1,737,300
SDE ($N=200$, this study)		10	6.902	0	6.923	559,860	559,860

^aThe cost of the cut-set pipes is \$0.211 million by assigning them with the minimum pipe diameters (150 mm).

^bInfeasible solution determined by EPANET2.0 with the maximum head deficit of 0.2 meters.

^cThe total computational overhead required by the proposed method has been converted to the equivalent number of the whole network evaluations (TRN₁+TRN₂+TRN₃+cut-set+TRN).

As shown in Table 11, the proposed method found the current best solution for the TRN case study with a cost of \$6.823 million. The best solutions found by the SDE algorithms with $N=500$ and $N=200$ were \$6.874 million and \$6.902 million respectively, which are 0.73% and 1.17% higher than the current best solution found by the proposed method. It was also found that the proposed method performed better than the SDE algorithm in terms of the average cost of solution quality based on ten different runs. The most noticeable advantage of the proposed method was that it

converged to the optimal solutions with significantly greater speed than the SDE algorithm. This is reflected by the fact that the proposed method required an average 270,171 total equivalent full TRN evaluations to find the optimal solutions, while the SDE algorithm with $N=200$ and $N=500$ used average 559,860 and 1,737,300 evaluations respectively as shown in Table 11.

The best and the average approximate optimal solution obtained by the first stage optimization were \$6.874 million and \$6.883 million, which is only 0.75% and 0.88% larger than the current best solution found by the proposed method after the second stage optimization (\$6.823 million). In addition, these approximate optimal solutions were located extremely quickly since they only required an average number of 24,411 equivalent full TRN evaluations as presented in Table 11.

For this case study, a sensitivity analysis for variations in the nodal demands and Hazen-Williams coefficients (HWs) has been conducted to investigate the impact on the final solution. A nodal demand multiplier (R) was used to adjust the demands for each node. For example, $R=0.9$ indicates the new demands of each node are 0.9 times the current demand. In this study, values of $R=0.9$ and 1.1 were used to undertake the sensitivity analysis on the nodal demands, while maintaining a consistent Hazen-Williams coefficients value ($HW=130$).

Additionally, the values of HW of 100 and 115 were used to analyze the sensitivity of the final solution on the Hazen-Williams coefficient (HW) for the TRN case study. The nodal demands for each node were kept constant ($R=1.0$). Finally, each node was randomly assigned a value of R in the range of $[0.9, 1.1]$ and each link was assigned a value of HW in the range of $[100, 130]$ for the TRN case study. The results of the proposed decomposition and multi-stage method applied to the TRN case study with the variation of demands and Hazen-Williams coefficient values are presented in Table 12.

As shown in Table 12, for a $HW=130$, the cost of the final optimal solutions obtained by the proposed method increases for an R value that is greater. The cost of the best

solution and the average cost solution for the TRN case study with $R=1.0$ increases by 4.3% and 4.5% respectively compared to those with $R=0.9$, while it decreases by 4.0% and 3.8% compared to those with $R=1.1$. When the nodal demand was constant ($R=1$), the proposed method found the lower cost solutions as the value of HW increases as displayed in Table 12. This is to be expected as a larger HW value reflects a smoother pipe.

Table12 Sensitivity analysis for the TRN case study (HW : Hazen-Williams coefficient. R =nodal demand multiplier)

Values of HW and R		Number of trial runs	Best solution found (\$M)	Average cost solution (\$M)	Average number of equivalent full TRN evaluations to find best solution
$HW=130$	$R=0.9$	10	6.542	6.549	279,985
	$R=1.0$	10	6.823	6.844	270,171
	$R=1.1$	10	7.100	7.107	315,700
$R=1.0$	$HW=100$	10	7.629	7.637	280,720
	$HW=115$	10	7.177	7.182	303,600
	$HW=130$	10	6.823	6.844	270,171
$R=[0.9, 1.1], HW=[100,130]$		10	7.176	7.186	288,520

The best solution obtained for the TRN with $HW=100$ is \$7.629 million ($R=1$), which is 6.3% and 11.8% higher than those found for the TRN with $HW=115$ and $HW=130$ respectively. The best solution found by the proposed method for the TRN with randomly assigned R values (in the range of [0.9, 1.1]) for each node and randomly assigned HW values (in the range of [100, 130]) for each link is \$7.176 million, which is 5.2% higher than the best solution found for the original TRN with $R=1.0$ and $HW=130$ (\$6.823 million).

The average number of equivalent full TRN evaluations required by the proposed method applied to each network with variations of demands and HW values are similar. This shows that the search efficiency of the proposed method is not significantly affected by network parameter variations (demands and HW values).

6.4. Case study 4: Four-reservoir WDS (Balerma network)

The four-reservoir network (FRN) is the Balerma network, which was first investigated by *Reca and Martínez* [2006]. It consists of 4 reservoirs, 8 loops, 454 pipes and 443 demand nodes as shown in Figure 9. Ten PVC commercial pipes with nominal diameters from 125 mm to 600 mm are to be selected for this network and hence the search space is 10^{454} . All the pipes are assumed to have an absolute roughness height of $k=0.0025$ mm and the minimum required pressure at each node is 20 meters. Pipe costs are given by *Reca and Martínez* [2006]. For this case study, the total choice table is composed of 10 pipe diameters for each pipe.

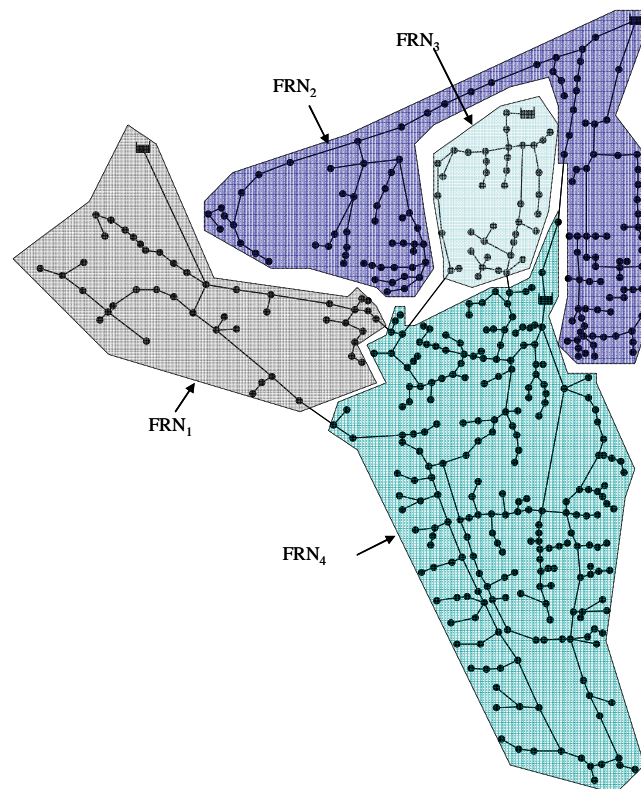


Figure 9 Layout, the optimal source partitioning cut-set and the sub-networks (FRN₁, FRN₂, FRN₃ and FRN₄) of the four-reservoir network (FRN)

The optimal source partitioning cut-set for the FRN case study was identified to be composed of five pipes using the proposed method given in Figure 3. The whole FRN was partitioned into four sub-networks after removal of the optimal source

partitioning cut-set. These include FRN_1 , FRN_2 , FRN_3 and FRN_4 as shown in Figure 9. There were 45 demand nodes and 45 pipes in FRN_1 ; 130 demand nodes and 132 pipes in FRN_2 ; 41 demand nodes and 41 pipes in FRN_3 ; and 227 demand nodes and 231 pipes in FRN_4 . For the FRN case study, the computational time to identify the optimal source partitioning cut-set was equivalent to 32 whole FRN evaluations. The average computational time for one evaluation of FRN_1 , FRN_2 , FRN_3 and FRN_4 was equivalent to 0.031, 0.20, 0.031 and 0.52 whole FRN evaluations respectively based on 1000 runs using the same method as for the DRN case study. The pipe configuration for each sub-network and the full network was randomly generated for the 1000 runs.

For the FRN case study, because the size of the sub-networks varies significantly, the population size (N) and the maximum number of allowable evaluations of DE algorithms applied to different sub-network optimizations need to be slightly tuned. Table 13 gives the parameter values used for the DE algorithms run for the optimization of each sub-network and for the whole FRN optimization. These parameters values were selected based on a few trials. As can be seen from Table 13, the larger sub-network was given a larger population size and the maximum number of allowable evaluations. Two SDE algorithms with population sizes of $N=500$ and $N=2000$ were applied to the FRN case study. Values of $F=0.3$ and $CR=0.5$ were used for all the DE algorithms in this case study. The statistical results for these different algorithms and the published results for this case study are provided in Table 14.

Table 13 The DE algorithm parameter values applied to different sub-networks and the whole FRN (Balerma network)

Network	No. of decision variables (Pipes)	Population size (N)	Maximum number of allowable evaluations
FRN_1	45	100	20,000
FRN_2	132	200	200,000
FRN_3	41	100	20,000
FRN_4	231	300	800,000
FRN (the second phase DE algorithm)	454	200	10,000,000
FRN (the SDE)	454	500/2000	10,000,000

As displayed in Table 14, the current best known solution for the FRN case study was first reported by *Zheng et al.* [2011a] with a cost of €1.923 million using a NLP-DE method. This best solution was also found by the proposed method in this paper, however, using only an average of 639,960 total equivalent full FRN evaluations based on ten different runs, compared to 1,427,850 evaluations required by the NLP-DE method [*Zheng et al.* 2011a]. The best solution found by HD-DDS [*Tolson et al.* 2009] were €1.940 million using 30 million evaluations budget. The SDE algorithm with $N=500$ produced the best solution of €1.988 million spending 2,042,000 evaluations and the SDE algorithm with $N=2000$ yielded the best solution of €1.982 million with 9,230,000 evaluations.

Table 14 Algorithm performance for the FRN case study (Balerma network)

Algorithm		Number of trial runs	Best solution found (€M)	Average cost solution (€M)	Worst solution found (€M)	Average number of original evaluations to find best solution	Average number of equivalent full FRN evaluations to find best solution
Proposed method (This study)	FRN ₁	10	0.182	0.182	0.182	14,867	461
	FRN ₂	10	0.710	0.712	0.714	122,889	24,578
	FRN ₃	10	0.133	0.133	0.133	15,400	477
	FRN ₄	10	0.883	0.884	0.884	567,366	295,030
	FRN ₁ +FRN ₂ +FRN ₃ +FRN ₄ +cut-set pipes ^a	10	1.930 ^b	1.931	1.931	720,522	320,546
	FRN	10	1.923	1.931	1.935	319,360	319,360
	Total	10	-	-	-	-	639,906 ^c
NLP-DE [<i>Zheng et al.</i> 2011a]		10	1.923	1.927	1.934	1,427,850	1,427,850
HD-DDS [<i>Tolson et al.</i> 2009]		1	1.940	-	-	NA	30,000,000
SDE ($N=2000$, this study)		10	1.982	1.985	1.987	9,294,666	9,294,666
SDE ($N=500$, this study)		10	1.988	2.208	2.050	1,814,700	1,814,700
HS [<i>Geem</i> 2009]		1	2.018	-	-	10,000,000	10,000,000
GANOME GA [<i>Reca and Martínez</i> 2006]		10	2.302	2.334	2.350	10,000,000	10,000,000

^aThe cost of the cut-set pipes is \$19,674 by assigning them with the minimum pipe diameters.

^bInfeasible solution determined by EPANET2.0 with the maximum head deficit of 2.2 meters.

^cThe total computational overhead required by the proposed method has been converted to the equivalent number of the whole network evaluations (FRN₁+FRN₂+FRN₃+FRN₄+cut-set+FRN).

Reca and Martínez [2006] and *Geem* [2009] employed the GANOME GA and HS to find the best solutions of €2.302 million and €2.018 million for this case study respectively running a total of 10 million evaluations. As shown in Table 14, the worst solution found by the proposed method based on the ten different runs is €1.935 million, which is still lower than the best solutions found by the majority of other algorithms presented in Table 14. From these results, it is concluded that the proposed method is able to find better solutions for this case study with higher reliability than the majority of other optimization techniques.

In terms of efficiency (total equivalent number of evaluations), the proposed method found the best solution 1.23 times faster than the NLP-DE method; 44.8 times faster than the HD-DDS; 13.6 times faster than the SDE algorithm with population size of $N=2000$; 1.83 times faster than the SDE algorithm with population size of $N=500$; and 14.3 times faster than GANOME GA and HS. This implies that the proposed decomposition and multi-stage optimization approach is able to find optimal solutions for such a relatively large case study (454 decision variables) with substantially improved efficiency compared to all other algorithms presented in Table 14.

It is interesting to note that the best approximate optimal solution generated by the first stage optimization of the proposed method was €1.930 million, which is only 0.7% higher than the current best solution for the FRN case study produced by the proposed method after the second stage optimization. The average cost of the ten approximate optimal solutions was €1.931 million, which is also extremely close to the current best solution. In addition, these approximate optimal solutions were found with extremely good efficiency as shown in Table 14. Although these approximate optimal solutions were infeasible when determined by EPANET2.0, they are able to specify promising regions for the second stage optimization of the proposed method, thereby allowing the good quality solutions for the whole FRN to be found efficiently.

6.5. Sub-network optimization analysis (first stage optimization)

Table 15 summarizes the number of pipes for which the diameters in the approximate optimal solutions (produced by the sub-network optimization during the first stage optimization) are different from the current best known solutions for each case study.

Table 15 Summary of the number of different pipe diameters for the approximate optimal solutions and the current best known solutions for each case study

Case study	Number of pipes	Number of pipes in optimal source partitioning cut-set (Ω)	Number of different runs	Number of pipes different in diameters between the approximate solution and the current best known solution
Two-reservoir WDS	6	2	100	1 to 2
TRN	287	14	10	29 to 35
FRN	454	5	10	52 to 61

As can be seen from Table 15, the number of different pipes diameters range from only 1 to 2 for the two-reservoir WDS case study (6 total pipes), 6 to 8 for the DRN case study (34 total pipes) based on 100 different runs, from only 29 to 35 for the TRN case study (287 total pipes) and from only 52-61 for the FRN case study (454 total pipes) based on ten different runs. Thus the majority of the pipes in the approximate optimal solution obtained in the first stage optimization have the same diameters as for those in the current best known solution for each case study. This demonstrates that the proposed source partitioning approach for a WDS with multiple supply sources is effective in terms of providing good initial estimates for the whole-of-network optimization. This is proven in that the network configuration obtained by combining each sub-network's design is extremely close to that provided by the final optimal solution as shown in Table 15.

Thus, it can be concluded that during the first stage optimization phase of the proposed decomposition and multi-stage optimization approach, the approximate optimal solutions for the whole network were efficiently found with very satisfactory quality in terms of both cost and network configuration compared to the current best known solution for each case study. The benefits are attributed to two factors

including: (i) each DE algorithm is used to deal with a portion of the whole network and hence explore a significantly reduced search space in the proposed method. This allows good quality solutions for each sub-network to be located with substantially improved efficiency; and (ii) the sum of computational overhead for each sub-network's hydraulic evaluation is smaller than that of one whole network evaluation.

7. Conclusions and future work

A novel decomposition and multi-stage optimization method is proposed to optimize the design of water distribution systems with multiple supply sources. The proposed method begins by identifying an optimal source partitioning cut-set for a given water network with K supply sources based on the available friction slopes at each node. The whole water network is then partitioned into K disconnected sub-networks after the removal of the optimal source partitioning cut-set. A total of K independent DE algorithms are used to optimize the K sub-networks individually during the first stage optimization. The optimal solutions for each sub-network plus the optimal source partitioning cut-set with the minimum allowable pipe diameter are used to create a tailored seeding table. Another DE algorithm is seeded with this given seeding table to optimize the design of the original whole network during the second stage optimization.

The proposed method was applied to four case studies and the results were compared with those of standard DE algorithms seeded with the total choice table also applied to these three case studies. It was found that the proposed method (decomposition followed by two-stage optimization) significantly outperforms the SDE algorithms in terms of solution quality and efficiency. Based on the results of the proposed method applied to the three case studies, the following observations can be made.

(1) The proposed partitioning approach for a WDS with multiple supply sources based on the available friction slopes at each node is effective. This is reflected by the fact that (i) the approximate optimal solutions obtained from the sub-network

optimizations were extremely close to the current best solution for each case study in terms of both solution costs and network configurations and (ii) the good quality solutions for each case study were found efficiently by a DE seeded by the tailored seeding table obtained from sub-network optimization.

(2) The computational overhead required to find the optimal source partitioning cut-set for a given WDS with multiple supply sources is negligible compared with that required by the whole optimization process (smaller than 0.01% of the total time). This indicates that the proposed algorithm given in Figure 3 used to identify the optimal source partitioning cut-set for a WDS with multiple supply sources is extremely efficient.

(3) The DE algorithm seeded with the tailored seeding table based on the approximate optimal solution efficiently produces better quality solutions than the standard DE algorithm seeded with the total choice table.

(4) The proposed method found the new current best solution for the DRN with a cost of \$1.750 million and the current best known solution for the FRN case study with the best known efficiency. The proposed method produced a current best solution for the TRN case study, with a value of \$6.823 million ($R=1.0$ and $HW=130$).

The performance of the proposed method has been compared with other previously reported optimization techniques based on the three case studies. It was found that the newly proposed method (decomposition followed by two-stage optimization) yielded better optimal solutions than other optimization techniques such as GAs and the HD-DDS with an extremely faster convergence speed.

It is important to note that the computational time for each sub-network optimization was added to the total computational time for the whole proposed optimization process in this study. This is due to the fact that sub-network optimization is individually completed in a predetermined sequence. However, sub-network

optimization using this proposed method can actually be undertaken utilizing parallel computing technology or multiple computers. In this case, all the sub-network optimizations could be started simultaneously, therefore further improving the efficiency of the whole optimization process. Thus, the proposed method provides an opportunity to exploit parallel computing techniques for the design optimization of a WDS with multiple supply sources.

The proposed decomposition and multi-stage optimization method presented in this paper has been demonstrated to be effective in finding the least-cost design (single objective optimization) for water distribution systems with multiple supply sources (WDS-MSS). A further future extension to the proposed method would be to deal with multi-objective optimization problems for WDS-MSS, in which say both the network cost and reliability will be considered. For the purpose of multi-objective optimization for WDS-MSS, a multi-objective optimization technique (such as NSGA-II: *Deb et al.* [2002] or Borg: *Hadka and Reed* [2012]) could be used to deal with sub-networks separately during the first stage optimization phase. Then another multi-objective optimization run would be seeded by the results obtained from the first stage optimization in order to generate multi-objective optimal solutions for the original whole WDS-MSS. This extension could be a focus of future work.

It should be noted that in this study the proposed decomposition and multi-stage optimization method aims to optimize a regional water supply system with multiple supply sources (WDS-MSS). The proposed method may not be transferred directly to deal with the optimization of a local supply system. For example if pressure reducing valves are used to partition a local water supply system into different zones, then application of the proposed method will require significant modification. Again, another future focus will be to extend the proposed method to deal with the optimization design of more complex networks (local water supply systems), for which the pumps, valves, tanks and multiple demand loading cases may be involved.

Although the proposed decomposition and multi-stage optimization method has been applied to find the optimal design for water distribution systems with multiple supply sources (WDS-MSS) in this paper, this concept (i.e., decomposition followed by multi-stage optimization) could be also transferred or extended to deal with other optimization problems, such as leakage hotspot detection [Wu and Sage 2006], contaminant detection [Weickgenannt et al. 2010] and the real-time optimization problems for WDSs [Kang and Lansey 2010].

8. Acknowledgements

The first author gratefully acknowledges the scholarship provided by Australian government and the University of Adelaide.

References

- Alperovits, E., and U. Shamir (1977), Design of water distribution systems, *Water Resour. Res.*, 13(6), 885-900, doi:10.1029/WR013i006p00885.
- Bolognesi, A., C. Bragalli, A. Marchi, and S. Artina (2010), Genetic heritage evolution by stochastic transmission in the optimal design of water distribution networks, *Advances in Engineering Software*, 41(5), 792-801.
- Broad, D. R., G. C. Dandy, and H. R. Maier (2005), Water distribution system optimization using metamodels, *Journal of Water Resources Planning and Management*, 131(3), 172-180.
- Canon, J., J. Gonzalez, and J. Valdes (2009), Reservoir operation and water allocation to mitigate drought effects in crops: A multilevel optimization using the drought frequency index, *Journal of Water Resources Planning and Management*, 135(6), 458-465.
- Cisty, M. (2010), Hybrid genetic algorithm and linear programming method for least-cost design of water distribution systems, *Water Resources Management*, 24(1), 1-24, doi: 10.1007/s11269-009-9434-1.
- Dandy, G. C., A. R. Simpson, and L. J. Murphy (1996), An improved genetic algorithm for pipe network optimization, *Water Resour. Res.*, 32(2), 449-457, doi:10.1029/95wr02917.

- Deb, K. (2000), An efficient constraint handling method for genetic algorithms, *J Comput. Methods Appl. Mech. Engineering*, 186,311-338, doi:10.1016/s0045-7825(99)00389-8.
- Deb, K., A. Pratap, S. Agawal and T. Meyarivan (2002), A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Trans Evol. Comput.*, 6(2), 182-197.
- Deo, N (1974). *Graph theory with applications to engineering and computer science*. Prentice-Hall, Englewood Cliffs, N. J.
- Deuerlein, J. W. (2008), Decomposition model of a general water supply network graph, *Journal of Hydraulic Engineering*, 134(6), 822-832, doi: 10.1061/(asce)0733-9429(2008)134:6(822).
- Di Nardo, A., and M. Di Natale (2010), A heuristic design support methodology based on graph theory for district metering of water supply networks, *Engineering Optimization*, 43(2), 193-211, doi: 10.1080/03052151003789858.
- Eiger, G., U. Shamir, and A. Ben-Tal (1994), Optimal design of water distribution networks, *Water Resour. Res.*, 30(9), 2637-2646, doi: 10.1029/94wr00623.
- Eusuff, M. M., and K. E. Lansey (2003), Optimisation of water distribution network design using shuffled frog leaping algorithm, *J. Water Resources Planning and Management*, 129(3), 210-225.
- Fujiwara, O., and D. B. Khang (1990), A two-phase decomposition method for optimal design of looped water distribution networks, *Water Resour. Res.*, 26(4), 539-549, doi: 10.1029/WR026i004p00539.
- Geem, Z. W., J. H. Kim, and G. V. Loganathan (2002), Harmony search optimization: application to pipe network design, *Int. J. Model. Simulat.*, 22(2), 125-133.
- Geem, Z. W. (2009), Particle-swarm harmony search for water network design, *Engineering Optimization*, 41(4), 297-311
- Grefenstette, J. J. (1987), *Incorporating problem specific knowledge into genetic algorithms*. In: Davis, L. (Ed.), *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann, Los Altos, CA.
- Haghighi, A., H. Samani, and Z. Samani (2011), GA-ILP method for optimization of water distribution networks, *Water Resources Management*, 25(7), 1791-1808, doi: 10.1007/s11269-011-9775-4.
- Haimes, Y. Y., 1971, Modeling and control of the pollution of water resources systems via multilevel approach, *Journal of the American Water Resources Association*, 7(1), 93-101, doi: 10.1111/j.1752-1688.1971.tb01681.x.

- Harik, G. R., and D. E. Goldberg (2000), Linkage learning through probabilistic expression, *Computer Methods in Applied Mechanics and Engineering*, 186(2-4), 295-310, doi: 10.1016/s0045-7825(99)00388-6.
- Hass, J. E. (1970), Optimal taxing for the abatement of water pollution, *Water Resour. Res.*, 6(2), 353-365, doi: 10.1029/WR006i002p00353.
- Hadka, D., and P. Reed (2012), Borg: An auto-adaptive many-objective evolutionary computing framework, *Evolutionary Computation* doi: 10.1162/EVCOa00075.
- Kadu, M. S., G. Rajesh, and P. R. Bhave (2008), Optimal design of water networks using a modified genetic algorithm with reduction in the search space, *J. Water Resources Planning and Management*, 134(2), 147-160.
- Kang, D., and K. Lansley (2010), Real-time optimal valve operation and booster disinfection for water quality in water distribution systems, *Journal of Water Resources Planning and Management*, 136(4), 463-473.
- Keedwell, E., and S.-T. Khu (2006), Novel cellular automata approach to optimal water distribution network design, *Journal of Computing in Civil Engineering*, 20(1), 49-56, doi: 10.1061/(ASCE)0887-3801(2006)20:1(49).
- Krapivka, A., and A. Ostfeld (2009), Coupled genetic algorithm-linear programming scheme for least-cost pipe sizing of water-distribution systems, *J. Water Resources Planning and Management*, 135(4), 298-302, doi:10.1061/(ASCE)0733-9496(2009)135:4(298).
- Lin, M.-D., Y.-H. Liu, G.-F. Liu, and C.-W. Chu (2007), Scatter search heuristic for least-cost design of water distribution networks, *Engineering Optimization*, 39(7), 857 – 876, doi: 10.1080/03052150701503611.
- Lippai, I., J. P. Heaney, and M. Laguna (1999), Robust water system design with commercial intelligent search optimizers, *Journal of Computing in Civil Engineering*, 13(3), 135-143, doi:10.1061/(ASCE)0887-3801(1999)13:3(135).
- Loganathan, G. V., J. J. Greene, and T. J. Ahn (1995), Design heuristic for globally minimum cost water-distribution systems, *Journal of Water Resources Planning and Management*, 121(2), 182-192, doi:10.1061/(ASCE)0733-9496(1995)121:2(182).
- Maier, H. R., A. R. Simpson, A. C. Zecchin, W. F. Foong, K. Y. Phang, H. Y. Seah, and C. L. Tan (2003), Ant colony optimization for the design of water distribution systems, *J. Water Resources Planning and Management*, 129(3), 200-209, doi:10.1061/(ASCE)0733-9496(2003)129:3(200).

- Mohan, S., and K. S. J. Babu (2010), Optimal water distribution network design with honey-bee mating optimization, *Journal of Computing in Civil Engineering*, 24(1), 117-126, doi: 10.1061/(ASCE)cp.1943-5487.0000018.
- Ostfeld, A. (2005), Water distribution systems connectivity analysis, *Journal of Water Resources Planning and Management*, 131(1), 58-66, doi: 10.1061/(ASCE)0733-9496(2005)131:1(58).
- Perelman, L., and A. Ostfeld (2007), An adaptive heuristic cross-entropy algorithm for optimal design of water distribution systems, *Engineering Optimization*, 39(4), 413 – 428, doi: 10.1080/03052150601154671.
- Reca, J., and J. Martínez (2006), Genetic Algorithms for the design of looped irrigation water distribution networks, *Water Resour. Res.*, 42, W05416, doi:10.1029/2005WR004383.
- Simpson, A. R., G. C. Dandy, and L. J. Murphy (1994), Genetic algorithms compared to other techniques for pipe optimization, *J. Water Resources Planning and Management*, 120(4), 423-443, doi: 10.1061/(ASCE)0733-9496(1994)120:4(423).
- Storn, R., and K. Price (1995), Differential evolution-A simple and efficient adaptive scheme for global optimization over continuous space, *Technical Report*, International Computer Science Institute, Berkeley, CA.
- Suribabu, C. R., and T. R. Neelakantan (2006), Design of water distribution networks using particle swarm optimization, *Urban Water J.*, 3(2), 111-120, doi:10.1080/15730620600855928.
- Suribabu, C. R. (2010), Differential evolution algorithm for optimal design of water distribution networks, *Journal of Hydroinformatics*, 12 (1), 66-82.
- Tolson, B. A., M. Asadzadeh, H. R. Maier, and A. C. Zecchin (2009), Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization, *Water Resour. Res.*, 45, W12416, doi:10.1029/2008WR007673.
- Vasan, A., and S. P. Simonovic (2010), Optimization of water distribution network design using differential evolution, *J. Water Resource Planning and Management*, 136(2), 279-287, doi: 10.1061/(ASCE)0733-9496(2010)136:2(279).
- Vesterstrom, J., and R. Thomsen (2004), A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, *Proceeding of Evolutionary Computation Conference (CEC2004)*, 2, 1980 -1987, doi: 10.1109/CEC.2004.1331139.

- Weickgenannt, M., Z. Kapelan, M. Blokker, and D. A. Savic (2010), Risk-based sensor placement for contaminant detection in water distribution systems, *Journal of Water Resources Planning and Management*, 136(6), 629-636, doi: 10.1061/(asce)wr.1943-5452.0000073.
- Wu, Z. Y., and P. Sage (2006), Efficient pressure dependent demand model for large water distribution system analysis, *Proceeding of 8th Annual International Symposium on Water distribution system analysis (ASCE)*, Cincinnati, Ohio., August 27-30.
- Yazdani, A., and P. Jeffrey (2010), Robustness and vulnerability analysis of water distribution networks using graph theoretic and complex network principles, *Proceeding of Water Distribution System Analysis 2010*, Tucson, AZ, USA, Sept. 12-15, doi:[http://dx.doi.org/10.1061/41203\(425\)85](http://dx.doi.org/10.1061/41203(425)85).
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2011a), A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems, *Water Resour. Res.*, 47(8), W08531,doi: 10.1029/2011wr010394.
- Zheng, F., A. R. Simpson, and A. Zecchin (2011b), Performance study of differential evolution with various mutation strategies applied to water distribution system optimization, *Proceedings of World Environmental and Water Resources Congress 2011*, ASCE, Palm Springs, California.
- Zhu, T., G. F. Marques, and J. R. Lund (2005), Multistage stochastic optimization for agricultural and urban water management with imperfect forecasts, *Proceedings of World Environmental and Water Resources Congress 2005*, ASCE, Anchorage, Alaska. doi:[http://dx.doi.org/10.1061/40792\(173\)37](http://dx.doi.org/10.1061/40792(173)37).

Chapter 9. Journal Paper 7-Graph Decomposition Optimization of WDS

9.1 Synopsis

A graph decomposition based approach for water distribution network optimization

Normally, a water network can be viewed as a connected graph $G(V,E)$, where V is a set of links and E is a set of nodes in the WDN. Thus, it is natural to introduce graph theory algorithms to facilitate WDN analysis. Traditionally, graph theory was used for water network connectivity and reliability analysis. This thesis develops a number of graph decomposition concepts for water networks in order to facilitate the optimization of the design. These decomposition concepts include (i) the determination of the shortest-distance tree for the water network presented in Chapter 6; (ii) the identification of the trees and core for the water network outlined in Chapter 7; (iii) the optimal source partitioning cut-set approach presented in Chapter 8, and (iv) the sub-network identification for the whole network which is given in Chapter 9.

The shortest-distance tree shown in Chapter 6 is used to decompose the original looped water network into a tree and chords. Consequently, non-linear programming (NLP) is used to obtain the optimal solution for the shortest-distance tree, which in turn is utilized to seed a differential evolution (DE) algorithm to optimize the original whole network.

The full water network is decomposed into trees and core in Chapter 7, where no loops are involved in the trees and loops are only included in the core. Then binary linear programming (BLP) is used to exclusively optimize the design for the trees and a DE is employed to find the optimal design for the core. Using this decomposition method, different components of the water network are optimized by different optimization techniques.

The optimal source partitioning concept proposed in Chapter 8 is used to decompose a complex water network with multiple supply sources into sub-networks based on the number of supply sources. One and only one supply source and a set of nodes and links are assigned to each sub-network. Rather than optimizing the full water network, the sub-

networks are optimized separately and the solutions from all sub-networks are combined to form an approximate optimal solution for the full network. This approximate optimal solution is used to initialize a DE in order to find further better solution for the original full network.

The sub-network identification concept presented in this Chapter differs from those in Chapters 6, 7 and 8. In this Chapter, a complete water network is decomposed into sub-networks based on the connectivity of the network's components. Each resulting sub-network contains a single block, bridges and trees if applicable. It should be highlighted here that the sub-network definition in this Chapter differs to the sub-network definition in Chapter 8. The differences include: (i) the sub-network identification in this Chapter is based on network's connectivity properties while the sub-network identification in Chapter 8 is based on the number of supply sources and (ii) no pipes are removed to identify the sub-networks in this Chapter while the source partitioning cut-set has to be removed in order to constitute the sub-networks in Chapter 8.

In the proposed decomposition based optimization approach in this Chapter, the original whole network is simplified to a directed augmented tree after the sub-network identification, in which the sub-networks are substituted by augmented nodes and directed links are created to connect them. A DE is then employed to optimize the design for each sub-network based on the sequence specified by the assigned directed links in the augmented tree. Rather than optimizing the original network as a whole, therefore, the sub-networks are sequentially optimized by the DE algorithm.

In the proposed graph decomposition based optimization method, a solution choice table is established for each sub-network (except for the root sub-network that includes the reservoir) and the optimal solution of the original whole network is finally obtained by use of the solution choice tables. Furthermore, in order to improve the efficiency of the optimization process, a pre-conditioning algorithm is developed to optimize the sub-networks, thereby producing an approximately optimal solution for the original whole

network. This solution specifies promising regions for the final optimization algorithm to further optimize the sub-networks, allowing better quality solutions to be found.

The proposed approach takes advantage of the fact that the evolutionary algorithm (DE in this research) is effective in exploring a relatively small search space. As the number of decision variables for each sub-network is significantly less than the original whole network, the DE is able to exploit the substantially reduced search space effectively and quickly. This allows good quality optimal solutions for each sub-network to be found with great efficiency. Another substantial benefit of the proposed method is that it provides a way to exploit parallel computing techniques for the design optimization of a WDS

This work has been submitted to *Water Resources Research* and the paper and the algorithm details are given as follows.

Citation of Paper

Zheng, F., A. R. Simpson, A. C. Zecchin and J. Deuerlein (2012b). “A graph decomposition based approach for water distribution network optimization.” *Water Resources Research*, submitted June 2012.

STATEMENT OF AUTHORSHIP (SUBMITTED)

Journal paper title: **A graph decomposition based approach for water distribution network optimization.**

Authors: Feifei Zheng¹, Angus Simpson², Aaron Zecchin³ and Jochen Deuerlein⁴

Corresponding author:

¹Feifei Zheng (Candidate): PhD student, School of Civil, Environmental and Mining Engineering, University of Adelaide

Wrote the manuscript, performed all analysis, developed the model and theory and acted as corresponding author.

I hereby certify that the statement of contribution is accurate.

Signed _____ Date _____

²Angus Simpson: Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide

Extensive discussion of concepts and reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

³Aaron Zecchin: Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide

Extensive discussion of concepts and reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

⁴Jochen Deuerlein: Senior Project Engineer, 3S Consult GmbH, Karlsruhe, Germany & Adjunct Senior Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide.

Extensive discussion of concepts and reviewed the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed _____ Date _____

Submitted to Water Resources Research, 2 June 2012

9.2 Journal Paper 7: A graph decomposition based approach for water distribution network optimization (Submitted to *Water Resources Research*)

Feifei Zheng, Angus R. Simpson, Aaron C. Zecchin and Jochen Deuerlein

Abstract

A novel optimization approach for water distribution network design is proposed in this paper. Using graph theory algorithms, a full water network is first decomposed into different sub-networks based on the connectivity of the network's components. The original whole network is simplified to a directed augmented tree, in which the sub-networks are substituted by augmented nodes and directed links are created to connect them. Differential evolution (DE) is then employed to optimize each sub-network based on the sequence specified by the assigned directed links in the augmented tree. Rather than optimizing the original network as a whole, the sub-networks are sequentially optimized by the DE algorithm. A solution choice table is established for each sub-network (except for the sub-network that includes a supply node) and the optimal solution of the original whole network is finally obtained by use of the solution choice tables. Furthermore, a pre-conditioning algorithm is applied to the sub-networks to produce an approximately optimal solution for the original whole network. This solution specifies promising regions for the final optimization algorithm to further optimize the sub-networks. Five water network case studies are used to demonstrate the effectiveness of the proposed optimization method. A standard DE algorithm (SDE) and a genetic algorithm (GA) are applied to each case study without network decomposition to enable a comparison with the proposed method. The results show that the proposed method consistently outperforms the SDE and GA (both with tuned parameters) in terms of both the solution quality and efficiency.

1. Introduction

The optimization of water distribution network (WDN) design has been investigated over the past few decades, and a number of optimization techniques have been developed to tackle WDN optimization problem. These include linear programming (LP) [Alperovits and Shamir 1977], nonlinear programming (NLP) [Fujiwara and Khang 1990], and evolutionary algorithms [Simpson *et al.* 1994; Dandy *et al.* 1996; Montesinos *et al.* 1999; Reza and Martínez 2006; Geem *et al.* 2002; Maier *et al.* 2003; Suribabu and Neelakantan 2006; Tolson *et al.* 2009; Suribabu 2010]. However, it has been found that each optimization algorithm has its own advantages and disadvantages.

For LP and NLP, optimal solutions can be located efficiently, while only local minimums are provided. Evolutionary algorithms (EAs) are able to find good quality solutions but are computationally expensive. A number of advanced methods have been proposed to reduce the computational intensity required by EAs in terms of WDN optimization [van Zyl *et al.* 2004; Tu *et al.* 2005; Keedwell and Khu 2006; Reis *et al.* 2006]. Combining optimization techniques with water network decomposition is one of those advanced methods.

Normally, a WDN can be viewed as a connected graph $G(V,E)$, where V is a set of links and E is a set of nodes in the WDN. Thus, it is natural to introduce graph theory algorithms to facilitate WDN analysis. Traditionally, graph theory was used for water network connectivity and reliability analysis. Gupta and Prasad [2000] used linear graph theory for the analysis of the pipe networks. Deuerlein [2008] proposed a graph theory algorithm to decompose a WDN into forests, bridges and blocks. This method provides a tool to simplify complex WDNs and provides a better understanding of the interactions between their different parts of the network.

In terms of WDN optimization, Kessler *et al.* [1990] developed a graph theory based algorithm to optimize the design of WDNs. In their work, the design process

consisted of three distinct stages. In the first stage alternative paths were allocated using graph theory algorithms. In the second stage the minimum hydraulic capacity (diameters) of each path was determined using an LP model. In the third stage the obtained solution from the second stage was tested by a network solver for various demand patterns.

Sonak and Bhave [1993] introduced a combined graph decomposition-LP algorithm for WDN design. In this combined algorithm, all the trees of the looped WDN were first identified by a graph theory algorithm and optimized by a LP, allowing the global optimum tree solution to be located. The final optimal solution for the original WDN was then determined by assigning the chords of the global optimum tree the minimum allowable pipe diameters. *Savic et al.* [1995] used graph theory to partition a water network into ‘tree’ and ‘co-tree’ to enable an optimization problem that involved minimizing the heads by setting regulation valves.

Kadu et al. [2008] proposed a genetic algorithm (GA) combined with a graph theory algorithm to optimize water distribution systems. In their method, graph theory is used to identify the critical path for each node in order to reduce the search space for the genetic algorithm. *Krapivka and Ostfeld* [2009] proposed a coupled GA-LP scheme for the least-cost pipe sizing of water networks. A spanning tree identification algorithm was introduced in their work. *Zheng et al.* [2011] proposed a combined NLP-DE algorithm to optimize WDNs. In this algorithm, a graph theory algorithm was first used to identify the shortest-distance tree for the original whole WDN. Then an NLP was implemented to optimize the tree network. The optimal solution obtained from the NLP optimization was finally utilized to seed a DE to optimize the original whole network.

Improvements in terms of efficiency and solution quality have consistently been reported by researchers when these optimization techniques are combined with graph theory algorithms and applied to WDN case studies. It was observed that, for the existing graph theory based optimization techniques, graph theory is normally used to

identify the critical path or the spanning tree for the WDN in order to facilitate the optimization.

For the proposed method here, a complete WDN is decomposed into sub-networks (rather than spanning trees) based on the connectivity of the network's components. The resulting sub-network may consist of a single block, bridges to this block and trees connected to this block if applicable. For relatively simple networks, (such as networks that have only one block and multiple trees attached to this block (case studies 2 and 3 in this paper)), the trees can be viewed as sub-networks. The sub-network containing the water supply node (reservoir) is designated the root sub-network. The definitions of block, bridge and tree for the water network are given by *Deuerlein* [2008], who described a block in a WDN as a maximal biconnected subgraph; a bridge is a link joining two disconnected parts of a graph; and a tree is a connected subgraph without any circuits or loops.

After the sub-networks have been identified, each one is represented as an augmented node and these augmented nodes are connected using *directed links* to form a directed augmented tree (*AT*), in which the directed links are used to specify the sub-network optimization sequence. In order to improve the efficiency of the optimization process, a pre-conditioning approach is developed to approximately optimize the sub-networks in order to produce an approximate optimal solution for the original full network. The obtained approximate solution is able to specify promising regions within the entire search space. A final optimization method is then used to exploit these promising regions in order to generate further improved solutions for the original full network. The details of the proposed methodology are given later.

2. Formulation of water distribution network optimization problem

Typically, a single-objective optimization of a WDN is the minimization of system costs (pipes, tanks and other components) while satisfying head constraints at each node. In

this paper, the proposed graph decomposition based optimization method is verified using WDN case studies with pipes only. Thus, the formulation of the WDS optimization problem can be given by:

$$\text{Minimize} \quad F = a \sum_{i=1}^{np} D_i^b L_i \quad (1)$$

Subject to:

$$H_k^{\min} \leq H_k \leq H_k^{\max} \quad k = 1, 2, \dots, nj \quad (2)$$

$$G(H_k, D) = 0 \quad (3)$$

$$D_i \in \{A\} \quad (4)$$

where F =network cost that is to be minimized [Simpson *et al.* 1994]; D_i =diameter of the pipe i ; L_i =length of the pipe i ; a , b =specified coefficients for the cost function; np =total number of pipes in the network; nj =total number of nodes in the network; $G(H_k, D)$ =nodal mass balance and loop (path) energy balance equations for the whole network, which is solved by a hydraulic simulation package (EPANET2.0 in this study) ; H_k =head at the node $k=1, 2, \dots, nj$; H_k^{\min} and H_k^{\max} are the lower and upper head limits at the nodes; A = a set of commercially available pipe diameters.

3. Methodology

Four steps are involved in the proposed method for optimizing a WDN.

Step 1 The sub-networks for the full WDN that is being optimized are identified using a graph decomposition algorithm.

Step 2 A directed augmented tree (AT) is built for the original full WDN. In the AT, the sub-networks appear as augmented nodes connected by directed links. The direction of the directed links in the AT determines the sub-network optimization sequence in the proposed method.

Step 3 The sub-networks are then preconditioned using a DE algorithm to produce an approximate optimal solution for the original full network.

Step 4 The sub-networks are further optimized by a DE algorithm based on the approximate optimal solution obtained in Step 3.

The details of each step are as follows.

3.1. Sub-network identification for the full water network (Step 1)

Deuerlein [2008] proposed a graph theory algorithm to decompose a water network graph (G) into forest, blocks and bridges according to its connectivity properties. In the method proposed here, however, the original network graph (G) is decomposed into a series of sub-networks (S). Each of the sub-networks may consist of one block, bridges to this block and trees attached to this block if applicable, or purely trees (if blocks are not applicable). Figure 1 illustrates the decomposition results of a water network using the proposed new method.

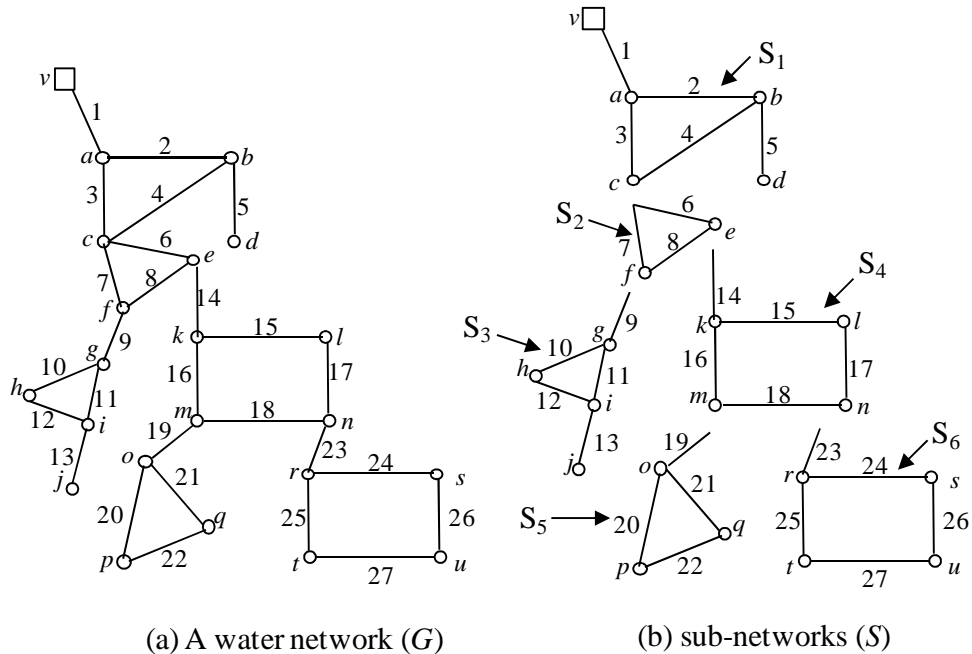


Figure 1 An example of 27-pipe water network decomposition. (a): the original water network; (b) the proposed decomposition results.

For the water distribution network (G) given in Figure 1(a), six sub-networks are identified specified as follows by a set of nodes and pipes, including $S_1=\{a, b, c, d, v, 1, 2, 3, 4, 5\}$, $S_2=\{e, f, 6, 7, 8,\}$, $S_3=\{g, h, i, j, 9, 10, 11, 12, 13\}$, $S_4=\{k, l, m, n, 14, 15, 16, 17, 18\}$, $S_5=\{o, p, q, 19, 20, 21, 22\}$ and $S_6=\{r, s, t, u, 23, 24, 25, 26, 27\}$. S_1 is denoted as root sub-network as it includes the supply source node v of the original water network.

As shown in Figure 1(b), each sub-network contains one and only one block, bridges to this block if applicable, and the trees attached to this block if applicable. The sub-networks overlap at some nodes as can be seen from Figure 1, i.e. $S_1 \cap S_2=c$, $S_2 \cap S_3=f$, $S_2 \cap S_4=e$, $S_4 \cap S_5=m$ and $S_4 \cap S_6=n$. In this study, nodes c, f, e, m and n are denoted as sub-network cut nodes (C), i.e. $C=\{c, f, e, m, n\}$. A depth first search (DFS) is employed to identify sub-network cut nodes [Tarjan 1972; Deuerlein 2008] to enable the network decomposition.

3.2. Directed augmented tree (AT) construction for the original WDN (Step 2)

In order to assist in visualizing the proposed optimization method, the decomposed water network G is reconstructed as a *directed augmented tree (AT)* by imagining each of the sub-networks as an *augmented node* and connecting the augmented nodes using directed links. The directed augmented tree AT of water network G given in Figure 1(a) is presented in Figure 2.

As shown in Figure 2, reflecting graph theory terminology, S_1 is the root augmented node in the AT since sub-network S_1 is the root sub network in Figure 1. S_2 and S_4 are located in the middle of the AT , while S_3, S_5 and S_6 are located at the leaves of the AT .

The AT is now used to illustrate the two novel features of the proposed optimization method, which are (i) *the optimization is carried out for each sub-network separately (rather than for the original full network as a whole) in a predetermined sequence specified by the directed links in the AT*; and (ii) *each sub-network design optimization incorporates the solutions for all the sub-networks that are immediately*

attached to this sub-network based on the direction of the directed links in the AT. Referring the novel feature (i), as specified by the directed links in the AT given in Figure 2, S_3 , S_5 and S_6 are first separately optimized, followed by S_4 ; then S_2 and finally is S_1 . That is, sub-network optimization takes place from the leaves to the root of the AT, which is opposite to the flow direction of the AT (that is from the root to the leaves as the supply source node is included in the root augmented node).

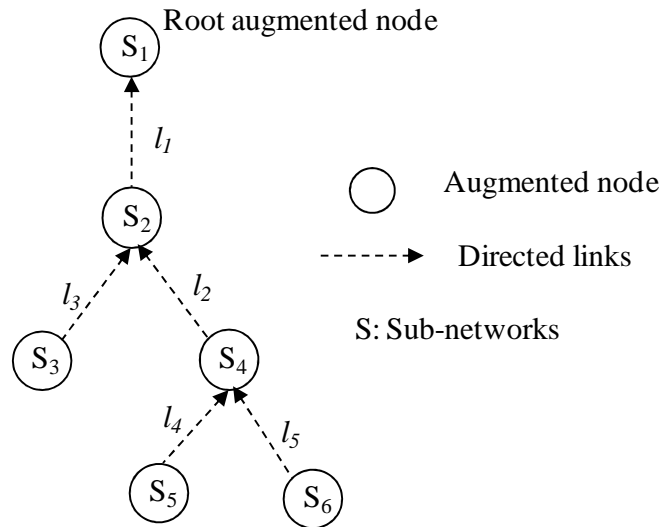


Figure 2 The directed augmented tree (AT) of the water network G given in Figure 1(a)

In order to facilitate the implementation of the novel feature (ii), for each sub-network represented by an augmented node in the AT, all the other sub-networks that are immediately attached to this sub-network based on the direction of the directed links are defined as its correlated sub-networks φ . Based on this definition, the correlated sub-networks for each sub-network given in Figure 2 is $\varphi(S_1)=\{S_2\}$, $\varphi(S_2)=\{S_3, S_4\}$, $\varphi(S_3)=\emptyset$, $\varphi(S_4)=\{S_5, S_6\}$, $\varphi(S_5)=\emptyset$ and $\varphi(S_6)=\emptyset$. Based on the novel feature (ii) of the proposed method, each sub-network design optimization needs to include the solutions for all the sub-networks in its φ .

By applying the two novel features to the water network given in Figure 1 (its AT is presented in Figure 2), S_3 , S_5 and S_6 should be first individually optimized and they do not consider other networks during optimization since their $\varphi=\emptyset$. Then S_4 is

optimized while incorporating the solutions for S_5 and S_6 since $\varphi(S_4)=\{S_5, S_6\}$. Subsequently, S_2 is optimized and S_3 and S_4 are included during the optimization ($\varphi(S_2)=\{S_3, S_4\}$). Finally, S_1 is optimized and S_2 is included ($\varphi(S_1)=\{S_2\}$).

As previously mentioned, two distinct optimization steps are utilized in the proposed method when dealing with the optimization design for a WDN, which are preconditioning optimization for the sub-networks (Step 3) and the final optimization for the sub-networks (Step 4). The details of these two proposed optimization algorithms are discussed in the later section. The water network given in Figure 1(a) (denoted N_I) is used to illustrate the proposed optimization approach. The water demands for each node and the length for each pipe are given in Table 1.

Table 1 Nodal and pipe information of N_I

Link	Length (m)	Node	Water demand (L/s)
1	800	v	Reservoir
2	750	a	25
3	600	b	27
4	485	c	32
5	452	d	15
6	478	e	48
7	492	f	20
8	562	g	124
9	145	h	14
10	785	i	32
11	456	j	13
12	325	k	17
13	148	l	22
14	478	m	42
15	528	n	89
16	400	o	26
17	258	p	23
18	547	q	11
19	500	r	19
20	200	s	17
21	200	t	16
22	900	u	32
23	654		
24	698		
25	250		
26	700		
27	254		

The elevation of all the demand nodes is 10 meters and the head provided by the supply source node (v) is 45 meters. The minimum head requirement for each demand node is 35 meters. The Hazen-Williams coefficient for each new pipe is 130. A total of 14 diameters ranging from 150 mm to 1000 mm are used for the N_I design. The pipe diameters and the cost for each diameter are given by *Kadu et al.* [2008].

3.3. Preconditioning optimization for the sub-networks (Step 3)

Three typical sub-networks can be defined for the decomposed network in the proposed method, including the sub-networks at the leaves ($L(AT)$), sub-networks in the middle of the directed augmented tree ($M(AT)$) and the root sub-network ($Rt(AT)$). For the sub-networks represented by augmented nodes in Figure 2, $\{S_3, S_5, S_6\} \in L(AT)$, $\{S_2, S_4\} \in M(AT)$ and $S_1 \in Rt(AT)$.

Sub-networks at the leaves [$S \in L(AT)$] differ from other sub-networks as their $\varphi = \emptyset$. The root sub-network [$S \in Rt(AT)$] is characterized by its known available head, since it includes the supply source node of the original WDN. The available heads of the sub-networks in the middle of the directed augmented tree ($S \in M(AT)$) are unknown and their $\varphi \neq \emptyset$, which are different from $S \in L(AT)$ and $S \in Rt(AT)$. In the proposed method, the optimization process for each type of the sub-network varies.

3.3.1 Optimization for the sub-network at the leaves of the AT

The sub-networks at the leaves ($S \in L(AT)$) are first optimized in the proposed method. Since no supply source node exists for each $S \in L(AT)$, each sub-network *cut node* connecting the $S \in L(AT)$ and the $S \in M(AT)$ is assumed to be a supply source node for $S \in L(AT)$. Therefore, the sub-network cut nodes f , m and n represent the supply source nodes for S_3 , S_5 , and S_6 respectively as shown in Figure 1 (b).

Since the available head (H) at a sub-network cut node is unknown, a series of sequential heads (H) between H_{min} and H_{max} are assigned for the sub-network cut node, where H_{min} is the maximum value of all minimum required nodal heads across the whole sub-network that is being optimized and H_{max} is the allowable head

provided by the supply source node of the original network. The logic behind setting the head range [i.e., $H \in (H_{min}, H_{max})$] is that no feasible solution can be found if the available head at the sub-network cut node is smaller than the maximum value of the minimum head constraints at all sub-network nodes, and the maximum head of the sub-network cut node cannot be greater than the head of the supply source node. A series of different H , $H \in (H_{min}, H_{max})$, with a particular interval (say one meter) are used for the sub-network cut node in order to enable sub-network optimization.

For each value of H assigned to a sub-network cut node, a differential evolution (DE) algorithm combined with a hydraulic simulation model (EPANET2.0) is used to optimize the sub-network design, while satisfying the head requirements for each node within the sub-network. The minimum pressure head excess H_{excess} ($H_{excess} \geq 0$) across the sub-network is obtained for each optimal solution associated with a particular value of H at the sub-network cut node. This indicates that the head at the sub-network cut node can be further reduced by H_{excess} while maintaining the feasibility of this optimal solution. The head H at the sub-network cut node is then adjusted to H^* , where $H^* = H - H_{excess}$, which is the minimum head requirement at the sub-network cut node for the optimal solution associated with the minimum pressure head excess H_{excess} .

Consequently, a solution choice table (ST) is constituted for the sub-network that is being optimized by assigning a series of different values of H to its assumed supply source node, sub-network cut node. In the ST , H^* , optimal solution costs and the sub-network configurations (pipe diameters) of optimal solutions are included and each unique H^* is associated with a unique optimal solution (including the cost and the sub-network configuration).

The sub-network S_6 in N_I is used to illustrate the proposed optimization method for the $S \in L(AT)$. The H_{min} and H_{max} values for S_6 are 35 and 45 meters respectively, where H_{min} is the maximum head requirement for all the nodes across S_6 (35 m) and the H_{max} is the allowable head provided by the actual supply source node (45 m). A

series of H ranging from 35 to 45 m with an increment of 1 m, i.e., $H = \{36, 37, 38, \dots, 45\}$ is used for the sub-network cut node n to optimize the design for S_6 . Note that no feasible solution can be found if $H=35$ m is assigned to node n as the minimum head requirement for S_6 is 35 m. Thus, the value of $H=35$ m is not included in the series of H values assigned for the sub-network cut node n . The optimal solution for each value of H , the minimum pressure head excess (H_{excess}) and the H^* value for each optimal solution for S_6 are given in Table 2.

As can be seen from Table 2, with values of H given at the sub-network cut node n from the smallest to the largest (the first column of Table 2), values of H^* are also ordered from the smallest to the largest, while its corresponding optimal solution is ordered from the largest to the smallest in terms of cost. This is due to the fact that a lower cost solution is achieved if a higher head is provided at the sub-network cut node. The solution choice table for S_6 of N_I includes the H^* values, the cost of the optimal solutions and the pipe diameters for optimal solutions as shown in the third, fourth and fifth columns in Table 2. This solution choice table is denoted as ST_n since the sub-network cut node n is the assumed supply source for S_6 .

Table 2 Optimal solutions for S_6 of N_I

H at sub-network cut node n (m)	Minimum pressure head excess H_{excess} (m)	$H^* = H - H_{excess}$ (m)	Cost of optimal solutions (\$)	Pipe diameters for each optimal solution (mm) ¹
36	0.014	35.986	155,487	450, 250, 300, 150, 300
37	0.231	36.769	130,288	400, 200, 300, 150, 250
38	0.157	37.843	115,622	350, 200, 250, 150, 300
39	0.120	38.880	108,175	350, 150, 250, 150, 250
40	0.397	39.603	105,079	350, 150, 250, 150, 200
41	0.513	40.487	98,175	300, 150, 250, 150, 250
42	0.790	41.210	95,079	300, 150, 250, 150, 200
43	0.402	42.598	92,032	300, 150, 200, 150, 200
44	1.402	42.598	92,032	300, 150, 200, 150, 200
45	0.160	44.840	89,168	250, 150, 200, 150, 250

¹The pipe diameters are for links 23 to 27 of N_I network (Figure 1 (a)) from the first to the last pipe respectively. [Note that only one solution is recorded in the table for the identical solutions (having the same H^* , optimal cost and pipe diameter for links)].

It is found in ST_n that the value of H^* , the cost and the pipe diameters for each link are the same for the optimal solutions generated when the heads at the sub-network

cut node n are 43 and 44 meters. This indicates that no further improved optimal solutions are to be found when the head at the sub-network cut node increases from 43 to 44 meters. It should be noted here that the identical solutions (having the same H^* , optimal cost and the pipe diameters for links) are removed from the solutions choice table (only one is left in the solution choice table).

Each $S \in L(AT)$ is optimized using the same approach as for S_6 described above and hence a solution choice table is constituted for each one after optimization. For N_I case study, in addition to S_6 , S_3 and S_5 are also sub-networks at the leaves of the directed augmented tree (see Figure 2). For S_3 and S_5 , $H_{min}=35\text{m}$ and $H_{max}=45\text{m}$, hence a series of values for $H = \{36, 37, 38, \dots, 45\}$ are used for the sub-network cut nodes f and m to optimize the design for the S_3 and S_5 respectively. As previously explained, $H=35\text{m}$ is not assigned to the sub-network cut nodes as no feasible solution can be found with this assumed head value (the minimum head requirement is 35m for the N_I case study). The obtained solution choice tables for S_3 and S_5 are presented in Table 3 (the identical solutions have been removed from solution choice tables).

Table 3 Solution choice tables for S3 and S5 of N_I .

Sub-network	H at assumed supply source node (m)	$H^* = H - H_{excess}$ (m)	Cost of optimal solutions (\$)	Pipe diameters for each optimal solution (mm) ¹
Solution Choice table for S_3 [$ST(f)$] where node f is the assumed supply source node for S_3	36	35.845	90,200	500, 150, 350, 200, 200
	37	36.939	73,900	400, 150, 300, 150, 200
	38	37.765	67,620	400, 150, 250, 150, 200
	39	38.886	63,553	350, 150, 250, 150, 150
	40	39.916	62,915	300, 150, 250, 150, 200
	41	40.903	60,483	400, 150, 200, 150, 150
	42	41.547	57,995	350, 150, 200, 150, 200
	43	42.575	57,357	300, 150, 200, 150, 200
	44, 45	43.054	55,778	300, 150, 200, 150, 150
Solution Choice table for S_5 [$ST(m)$] where node m is the assumed supply source node for S_5	36	35.995	74,686	350, 250, 150, 150
	37	36.864	64,603	300, 200, 150, 150
	38	37.925	62,469	300, 150, 150, 150
	39	38.649	57,717	250, 200, 150, 150
	40, 41, 42, 43, 44	39.710	55,583	250, 150, 150, 150
	45	44.607	51,623	200, 200, 150, 150

¹The pipe diameters are for links 9 to 13 of N_I network (Figure 1 (a)) in S_3 and for links 19 to 22 of N_I network in S_5 from the first to the last respectively.

As shown in Table 3, the solutions choice tables for S_3 [$ST(f)$] and S_5 [$ST(m)$] include a total of nine and six various optimal solutions respectively. For S_3 , the optimal solutions for the values of H at the assumed supply source node f being 44 and 45 m were the same obtained by the DE algorithm in terms of cost and network configuration. With $H=40, 41, 42, 43$ and 44 m at the assumed source node m for S_5 , the optimal solutions were the same and hence only one solution was included in the solution choice table [$ST(m)$] as shown in Table 3.

3.3.2 Optimization for the sub-network in the middle of the AT

The optimization for the $S \in M(AT)$ is carried out once the optimization for $S \in L(AT)$ has been finished. For each $S \in M(AT)$, the water demands at each sub-network cut node have to be increased by the flows in the directed links to this sub-network that is being optimized (note the direction of the flows is opposite to the directed links). For the example given in Figure 1(b), the water demands at sub-network cut nodes f , m and n [$f \in S_2, \{m, n\} \in S_4, \{S_2, S_4\} \in M(AT)$] are increased by the flows in directed link l_3 , l_4 and l_5 respectively (see Figure 2), which are actually the demands of sub-networks S_3 , S_5 and S_6 respectively. The water demand at sub-network cut node e is added by the flows in directed link l_2 , which are the total demands of sub-network S_4 , S_5 and S_6 , as shown in Figure 2.

It is noted that each $S \in L(AT)$ is connected to the original entire network via only one sub-network cut node, while each $S \in M(AT)$ is attached to the whole system with multiple sub-network cut nodes. For the example in Figure 1, S_3 , S_5 and S_6 belong to $L(AT)$ and each of them is connected to the whole network with only one sub-network cut node, which are nodes f , m and n respectively. In contrast, $S_2, S_4 \in M(AT)$ are attached to the whole network with more than one sub-network cut nodes, for which nodes e , m and n combine S_4 with other parts of the whole network, and c , f and e are used to connect S_2 to the original entire network.

Among these sub-network cut nodes attached to each $S \in M(AT)$, the one that is located at the upstream end based on the flow direction is assumed as supply source.

Thus, sub-network cut nodes c and e are the assumed supply sources for S_2 and S_4 respectively for the water network given in Figure 1. A series of different H , $H \in (H_{min}, H_{max})$, with a particular interval (of again say one meter) are assigned to the sub-network cut node for optimizing the $S \in M(AT)$, which is the same approach as for optimizing $S \in L(AT)$ described in Section 2.3.1.

It is important to note that for each $S \in M(AT)$, at least one sub-network is located at its immediately adjacent downward side based on the direction of the directed links in the AT , i.e. $\varphi \neq \emptyset$. In the proposed method, the optimization of each $S \in M(AT)$ needs to include all the sub-networks in its φ and the solutions for the sub-networks in its φ are selected from their corresponding solution choice tables during optimization. The formulation of the optimization problem for each $S \in M(AT)$ is given by:

$$\text{Minimize} \quad F' = F(S) + \sum f(\varphi(S)), S \in M(AT) \quad (5)$$

Subject to:

$$H_{S,k}^{\min} \leq H_{S,k} \leq H_{S,k}^{\max} \quad k = 1, \dots, nsj \quad (6)$$

$$G(H_{S,k}, D_s) = 0 \quad (7)$$

$$f(\varphi(S)) \in ST(\varphi(S)) \quad (8)$$

where F' = total cost (to be optimized); $F(S)$ = cost of the sub-network S ($S \in M(AT)$); $\sum f(\varphi(S))$ = total costs for all other sub-networks connected to the subnetwork S (φ); $G(H_{S,k}, D_s)$ = nodal mass balance and loop (path) energy balance equations for the sub-network S , which is handled by a hydraulic simulation package (EPANET2.0 in this study); $H_{S,k}$ = the nodal head of the node $k=1, \dots, nsj$; nsj = number of nodes within the sub-network S ; $H_{S,k}^{\min}$ and $H_{S,k}^{\max}$ are the lower and upper head boundaries at the nodes of S . $ST(\varphi(S))$ = the solution choice tables of sub-networks in the φ .

As shown from Equations (5) to (8), although the total costs of the $S \in M(AT)$ and all sub-networks in its φ are to be minimized, only the cost and the nodal head constraints of the

$S \in M(AT)$ are explicitly handled by an optimization algorithm (DE used in this study). This is because the optimal solutions for the sub-networks in φ [denoted as $f(\varphi(S))$] are selected from their corresponding solution choice tables $ST(\varphi(S))$ during optimization [Equation (8)]. In addition, head constraints of sub-networks in the φ are also handled by their corresponding solution choice tables. This is one of the novel aspects of the proposed optimization method. The details of the proposed method in terms of selecting optimal solutions from solutions choice tables and handling constraints during the optimization for the $S \in M(AT)$ are given as follows.

The optimization of S_4 in N_I is now used to illustrate the proposed methods for optimizing the $S \in M(AT)$. For the water network given in Figure 1 and its AT shown in Figure 2, $\varphi(S_4) = \{S_5, S_6\}$ and hence S_5 and S_6 are included when S_4 is optimized. For S_4 optimization, different values of $H = \{36, 37, 38, \dots, 45\}$ are used for the assumed supply source e ($H_{min}=35\text{m}$ and $H_{max}=45\text{m}$) and then a DE is employed to optimize the design for S_4 for each H value.

The total cost, including the cost of S_5 , the cost of S_6 and the cost of S_4 is to be minimized for the DE applied to optimize S_4 [$\varphi(S_4) = \{S_5, S_6\}$]. For each individual solution in the DE algorithm, the head at the sub-network cut nodes m (H_m) and n (H_n) are tracked after the hydraulic simulation for S_4 (EPANET2.0). Then the optimal solution for S_5 and S_6 are selected from their corresponding solution choice tables ST_m and ST_n based on assigning H_m and H_n to the sub-network cut nodes m and n . As the H_m and H_n may not equal precisely any particular H^* values in ST_m and ST_n , an approach is proposed in this study to select the appropriate optimal solutions based on the values of H_m and H_n . Figure 3 illustrates the details of this selection approach and the values of H^* versus the optimal solution costs in the solution choice table ST_n for S_6 is presented in Figure 3 to facilitate the explanation.

For each individual solution of the DE applied to optimize S_4 , H_n (head at the sub-network cut node n) is obtained after hydraulic simulation for S_4 . Based on the value of H_n , three cases exist for selecting the optimal solution for S_6 , as shown in Figure 3:

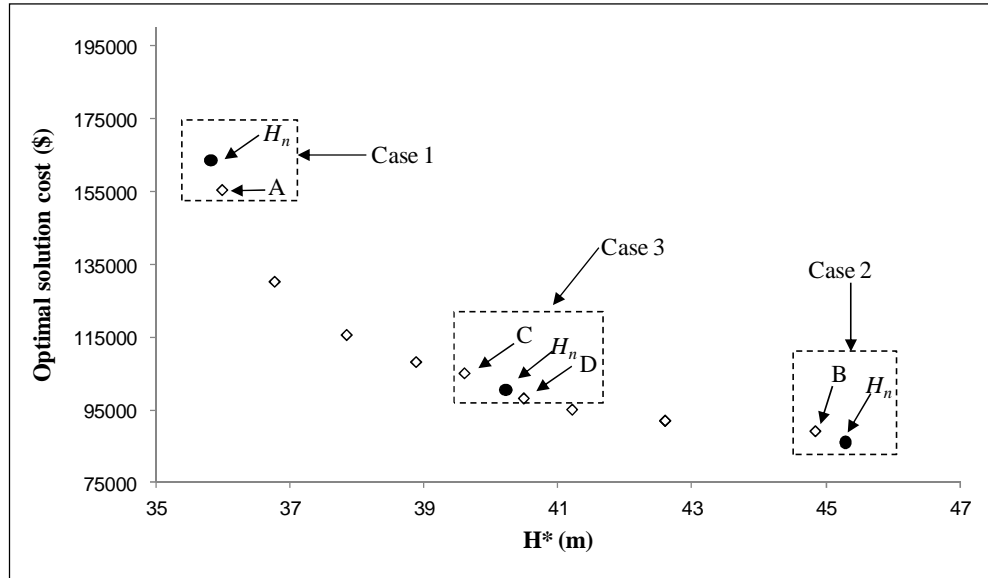


Figure 3 H^* versus the optimal solution cost for S_6 of N_I

Case 1: if H_n is smaller than the minimum H^* [$H^*(A)$] in ST_n , the cost associated with the minimum H^* (the cost of solution A in Figure 3) is added to the total cost of this individual solution and the network configuration (pipe diameters) associated with [$H^*(A)$] is assigned for S_6 . In addition, a penalty is applied to this individual solution as no feasible solution is found for S_6 .

Case 2: if H_n is greater than the maximum H^* ($H^*(B)$) in ST_n , the cost associated with the maximum H^* (the cost of solution B in Figure 3) is added to the total cost of this solution and the network configuration (pipe diameters) associated with [$H^*(B)$] is assigned for S_6 .

Case 3: if H_n is between two adjacent H^* values in ST_n , the solution has the H^* immediately smaller than the H_n is selected and its cost is added to the total cost of this individual solution. As shown in Figure 3, the solution C will be selected for S_6 if the individual solution has a H_n between $H^*(C)$ and $H^*(D)$, resulting in a pressure head excess of $H_n - H^*(C)$ for S_6 . As such, the solution selected from ST_n can be guaranteed to be feasible as the solution with H^* smaller than H_n is chosen. The network

configuration (pipe diameters) associated with $[H^*(C)]$ is assigned for S_6 in this case.

The approach described above is also used to include the cost of S_5 when a DE is used to optimize S_4 . As such, although only the pipes in S_4 are handled by the DE, the solutions in the DE actually include the total cost of S_4 , S_5 and S_6 . Once the DE has converged to the final optimal solution for S_4 , the minimum pressure head excess H_{excess} for this optimal solution is determined by:

$$H_{excess} = \min[H_{excess}^*, (H_m - H^*(ST_m)), (H_n - H^*(ST_n))] \quad (9)$$

where H_{excess}^* is the minimum pressure head excess across all the demand nodes for S_4 that is being optimized; $H^*(ST_m)$ and $H^*(ST_n)$ are the values of H^* associated with the solutions selected for S_5 and S_6 from ST_m and ST_n respectively based on the approach illustrated in Figure 3. The head H at the sub-network cut node e is then adjusted to H^* , where $H^* = H - H_{excess}$.

For each different value of H assigned to the sub-network cut node e , the optimal cost solution for S_4 , S_5 and S_6 is obtained by the DE algorithm. In addition, the H_{excess} is obtained using Equation (9) and hence the value of H^* ($H^* = H - H_{excess}$) is obtained for each optimal solution. As such, a solution choice table for S_4 is formed, in which, H^* , optimal solution cost and sub-networks configuration (pipe diameters for S_4 , S_5 and S_6) of the optimal solution are included, which is presented in Table 4.

As shown in Table 4, a total of nine different feasible optimal solutions were found by the DE applied to S_4 optimization with the heads at the assumed source node e being $\{36, 37, 38, \dots, 45\}$. No feasible solution was found with $H=36$ meters assigned to node e . In the solution choice table $ST(e)$ for S_4 , the values of H^* across the sub-networks of S_4 , S_5 and S_6 , the total cost of S_4 , S_5 and S_6 , the design for each of these three sub-networks are included.

Table 4 Solution choice table for S_4 of N_I

H cut node e (m)	$H^* = H - H_{excess}$ (m)	Cost of optimal solutions (\$)	Pipe diameters for each optimal solution (mm) ¹ in the solution choice table for S_4 [$ST(e)$]		
			S_4	S_5	S_6
36	-	infeasible	-	-	-
37	36.94	542,915	700, 600, 450, 600, 150	350, 250, 150, 150	450, 250, 300, 150, 300
38	37.94	484,396	600, 500, 400, 600, 150	350, 250, 150, 150	450, 250, 300, 150, 300
39	38.94	437,211	600, 500, 400, 500, 150	300, 200, 150, 150	400, 200, 300, 150, 250
40	39.76	414,439	600, 450, 350, 450, 150	300, 200, 150, 150	400, 200, 300, 150, 250
41	40.94	392,887	600, 450, 350, 450, 150	250, 200, 150, 150	350, 200, 250, 150, 300
42	41.86	380,809	500, 450, 350, 500, 150	300, 150, 150, 150	350, 200, 250, 150, 300
43	42.98	368,869	500, 500, 350, 400, 150	250, 200, 150, 150	350, 150, 250, 150, 250
44	43.78	348,862	500, 400, 300, 400, 150	250, 200, 150, 150	350, 200, 250, 150, 300
45	44.84	339,281	500, 400, 300, 400, 150	250, 150, 150, 150	350, 150, 250, 150, 250

¹The pipe diameters are for links 14 to 27 of N_I network from the first to the last respectively (see Figure 1 (a)).

As shown in Figure 2, $\varphi(S_2) = \{S_3, S_4\}$, thus S_3 and S_4 are included when S_2 is optimized in the proposed method. The sub-network S_4 is optimized before S_2 as the optimization sequence in the proposed method is from the leaves to the root based on the directed augmented tree. The approach described in Figure 3 was used to select the solutions for S_3 and S_4 from their corresponding solution choice tables when S_2 is optimized. A similar method presented in Equation (9) was utilized to obtain the H_{excess} for each optimal solution of S_2 . Since $H_{min}=35m$ and $H_{max}=45m$ for S_2 , $H = \{36, 37, 38, \dots, 45\}$ were used for the assumed supply source node c to optimize S_2 . In a similar way to that for S_4 , a solution choice table is formed for S_2 after the optimization, which is denoted as $ST(c)$ as the sub-network cut node c is the assumed supply source node. The final solutions in the $ST(c)$ are the optimal solutions for S_2 , S_3 and S_4 , which is actually the total optimal solutions for S_2 , S_3 , S_4 , S_5 and S_6 as the solutions in S_4 have already included the S_5 and S_6 . The designs for the optimal solutions of S_2 , S_3 , S_4 , S_5 and S_6 are also included in the $ST(c)$.

The formulation of the optimization problem given from Equations (5) to (8) and the approach used for S_4 optimization [Figure 3 and Equation (9)] are employed to optimize each $S \in M(AT)$, thereby a solution choice table is constituted for each sub-network in the middle of the directed augmented tree AT .

3.3.3 Optimization for the sub-network at the root of the AT

The root sub-network is the final one to be optimized in the proposed method. As the supply source node in the original full WDN is included in $S \in Rt(AT)$, the available head is known when optimizing $S \in Rt(AT)$. For the $S \in Rt(AT)$, $\varphi \neq \emptyset$ and hence the approach used for the optimization of $S \in M(AT)$ is also employed to deal with the optimization of the sub-network at the root of the AT . For the example given in Figure 1, $S_1 \in Rt(AT)$ and $\varphi(S_1) = S_2$, thus $ST(c)$ is used to provide the optimal solution for S_2 when S_1 is optimized.

An approximate optimal solution with a cost of \$1.021 million is obtained after the S_1 optimization, which is also the optimal solution for the whole N_I network. This is because S_5 and S_6 were included when S_4 was optimized, S_3 and S_4 were included when S_2 was optimized, and S_2 was in turn included when S_1 was optimized in the proposed method. Thus, the final optimal solutions from the optimization of S_1 are the optimization results for the original full network N_I .

During the pre-conditioning optimization for the sub-networks in the proposed method, a series of H with a relatively larger interval ($H \in (H_{min}, H_{max})$) is used for the sub-network cut nodes (1 meter in this study). This aims to approximately explore the search space of the original full network, thereby producing an approximate optimal solution. This approximate optimal solution is used to specify promising regions for the entire search space, allowing the next step (Step 4) of the final optimization for the sub-networks to be conducted. The final optimization for the sub-networks method is described in the next section.

3.4. Final optimization of the sub-networks (Step 4)

Based on the approximate optimal solution obtained by the preconditioning sub-networks optimization, an optimal head ($H^\#$) for each sub-network cut node can be determined. An optimal head range $\mathfrak{R}(H^\#)$ is created for each sub-network cut node

through expansion of the obtained optimal head, i.e. $\mathfrak{R}(H^\#)=[H^\# - \delta, H^\# + \delta]$. In this proposed method, $\delta=2$ meters is used to obtain the optimal head range $\mathfrak{R}(H^\#)$.

During the final optimization of the sub-networks, all the sub-networks are optimized employing the same approach used for preconditioning optimization for sub-networks, while the head assigned for the sub-network cut nodes is varied. For the preconditioning optimization for sub-networks, a whole range of possible H values between H_{min} and H_{max} at the sub-network cut nodes with a relatively large increment (1 meter) was used, while a series of H values within the optimal head range $\mathfrak{R}(H^\#)$ with a relatively small increment (e.g., 0.1 meter) was used for sub-network cut nodes during the final optimization of the sub-networks. The optimization sequence is also taken from the leaves to the root specified by the directed augmented tree in the final optimization step. The solution choice table for each sub-network created after the pre-conditioning optimization is updated during the final optimization step.

For the example given in Figure 1, the heads at the sub-network cut nodes n is 36.8 meters based on the approximately optimal solution obtained after the preconditioning sub-networks optimization (\$1.021 million). Thus the optimal heads range for node n is $\mathfrak{R}(H_n^\#)=[34.8, 38.8]$. The H^* versus the optimal solution cost for S_6 using the head given by the obtained optimal head range $\mathfrak{R}(H^\#)$ with an increment of 0.1 meter is given in Figure 4.

A total of 23 different optimal solutions were found for S_6 of the N_I case study with the head given at node n within the optimal head range $\mathfrak{R}(H^\#)$, compared to only nine different approximate optimal solutions generated during the pre-conditioning optimization step for S_6 . This shows that the proposed final optimization method is able to further exploit the promising regions specified by the optimal head range in the pre-conditioning phase, thereby allowing more optimal solutions to be located. This is also shown by Figure 4 that a number of additional optimal solutions were found by the final sub-network optimization process between two adjacent optimal solutions found initially by pre-conditioning.

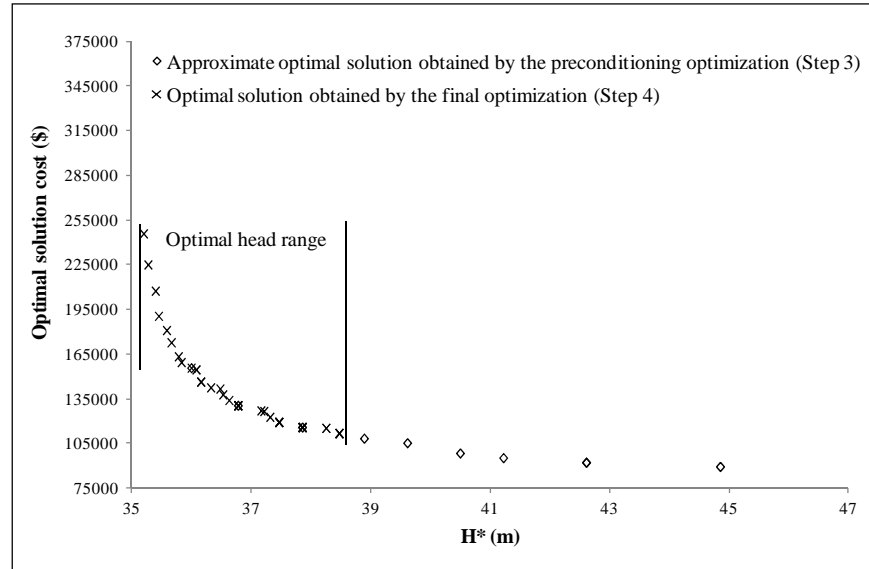


Figure 4 H^* versus the optimal solution cost for S_6 of N_I

All other sub-networks of N_I are optimized based on the obtained optimal head range for each sub-network cut node during the final optimization step. The final optimal solution for the N_I case study obtained after the final optimization step was \$1.016 million, a value lower than the optimal solution generated by the pre-conditioning optimization for sub-networks (Step 3) with a cost of \$1.021 million. This shows that the proposed final optimization of the sub-networks approach is effective in improving the quality of optimal solutions generated by the preconditioning optimization step.

3.5. Summary of the proposed method

The proposed method does not need to know the actual head constraints at the sub-network cut nodes, instead a series of assumed heads are assigned at sub-network cut nodes. Then the DE optimization is used to seek the least-cost design of the sub-network for each assumed head at the sub-network cut node, while satisfying the specified head requirement at each node (such as 35 m for the N_I network). This results in the development of a solution choice table for each sub-network (except the root sub-network). For each solution choice table, every H^* is associated with an optimally feasible solution (determined by EPANET2.0) for its corresponding sub-

network. Therefore, the final optimal solutions can be guaranteed to be feasible for the whole original WDN since all the selected optimal solutions from solution choice tables are feasible (i.e., all the head constraints are satisfied).

The proposed method recognizes the fact that, although decomposed, sub-networks in a WDN are in reality always interconnected and never truly independent of one another. Thus, for each sub-network optimization, all the sub-networks in its φ are considered. Therefore, the optimal solution obtained for each sub-network is actually the optimal solution as a whole of this sub-network and all the sub-networks in its φ . As the optimization is carried out *from the leaves to the root* along the assigned directed links in the directed augmented tree, the root sub-network contains all the sub-network optimization results by use of solution choice tables. Consequently, the optimal solution for the root sub-network is actually the final solution for the whole WDN.

In the proposed method, each sub-network optimization also considers all the sub-networks in its φ , while the number of decision variables handled is the number of pipes of the sub-network that is currently being optimized plus the number of solution choice tables that are associated with the sub-networks in the φ . This is because all the optimal solutions for the sub-networks in the φ are already provided by their corresponding solution choice tables.

4. Case study results and discussion

A total of five case studies are used to verify the effectiveness of the proposed optimization approach, including one artificial water network, two benchmark case studies and two real-world water networks. A DE combined with a hydraulic solver (EPANET2.0) was employed to optimize each sub-network design. In addition to the proposed graph decomposition optimization approach, a SDE and a GA with tuned parameters were applied to each case study in order to enable a performance comparison with the proposed method.

4.1. Case study 1: Artificial network 1 (N_I) (27 decision variables)

The layout and the network details of artificial network 1 (N_I) were previously provided in Figure 1(a) and Table 1 as examples of network decomposition. The decomposition results (sub-networks S_1 to S_6) and the directed augmented tree of N_I (directed links l_1 to l_5) are provided in Figure 1(b) and Figure 2 respectively. Table 5 summarizes the DE parameter values used for optimizing the full N_I and each network into which it has been decomposed by the graph theory algorithm. In addition, the computational times for running simulation on the whole N_I and each sub-network (S_1 , S_2 , S_3 , S_4 , S_5 and S_6) are provided. A mutation weighting factor (F) of 0.5 and a crossover rate (CR) of 0.5 were selected based on the results of a few parameter trials for the DE used in the proposed method, while the parameters of the SDE and GA have been fine-tuned through extensive parameter calibration. The best parameter values obtained were $F=0.6$, $CR=0.7$ for the SDE, and crossover probability (P_c) with 0.9 and mutation probability (P_m) with 0.03 were selected for the GA.

Table 5 Evolutionary algorithm parameter values and the hydraulic simulation time for each sub-network and the full N_I .

EAs	Network	No. of decision variables and the search space size	Population size (N)	Maximum number of allowable evaluations	The computational time for 1000 simulations (seconds) ¹
SDE	N_I	27 (8.82×10^{30})	100	500,000	0.765
GA	N_I	27 (8.82×10^{30})	200	800,000	0.765
DE used in the proposed method	S_1	5 (537,824)	20	2,000	0.105
	S_2	3 (2,744)	20	2,000	0.081
	S_3	5 (537,824)	20	2,000	0.110
	S_4	5 (537,824)	20	2,000	0.108
	S_5	4 (38,416)	20	2,000	0.095
	S_6	5 (537,824)	20	2,000	0.098

¹1000 simulations were based on randomly selected network configuration and conducted on the same computer configuration (Pentium PC (Inter R) at 3.0 GHz).

As previously mentioned, a total of 14 discrete diameters can be used for the N_I case study, thus the total search space size is $14^{27} \approx 8.82 \times 10^{30}$. The search spaces for sub-networks are significantly reduced compared to the original whole network as shown

in Table 5, Hence the population size (N) and maximum number of allowable evaluations assigned for the sub-network optimization are considerably less than those used by the original full network optimization as shown in Table 5.

The results of the proposed method and SDE applied to the N_I case study are provided in Table 6. As shown in Table 6, the current best solution for the N_I case study is \$1.016 million. This solution was found by the proposed method after the final optimization step with a success rate of 100% based on 50 different runs using different random number seeds, compared to 90% returned by the SDE. The best solution found by the GA was \$1.016 million, which is 0.3% higher than the current best solution (\$1.016 million) for this case study. In terms of average cost of solutions based on 50 runs, the proposed method exhibits similar performance with the SDE, but significantly outperformed the GA.

Table 6 Algorithm performance for the N_I case study

Algorithm	Number of different runs	Best solution found (\$M)	Percentage of trials with best solution found	Average cost solution (\$M)	Average number of equivalent evaluations to find best solution
Proposed method ¹	50	1.021	0%	1.021	15,608 ³
Proposed method ²	50	1.016	100%	1.016	78,039 ³
SDE ⁴	50	1.016	90%	1.017	152,854
GA ⁴	50	1.019	0%	1.027	392,676

¹The results of the proposed method after preconditioning sub-networks optimization (Step 3). ²The results of the proposed method after final sub-network optimization (Step 4). ³The total computational overhead required by the proposed method has been converted to the equivalent number of the whole network (N_I) evaluations. ⁴Parameters were tuned.

In order to enable a fair comparison in terms of efficiency, all the computational times required by the proposed method has been converted to the equivalent number of full N_I evaluations using the same computer configuration. These include the computational time used for identifying the sub-networks (equivalent to nine full N_I evaluations) and the computational time spent for the sub-networks optimization (Step 3 and 4). This conversion was made for each case study to allow an efficiency comparison between the proposed method and the SDE. As shown in Table 6, the

proposed method required an average number of full N_I evaluations of 78,039 to find the best solutions after the final optimization step.

The most noticeable advantage of the proposed graph decomposition optimization method is the significantly improved efficiency for finding the current best known solutions compared to the SDE and GA. The proposed method only required an average of 78,039 equivalent full network evaluations to find the optimal solutions, which is only 51% and 20% of those used by the SDE and GA respectively.

The results of the proposed method after the preconditioning optimization for the sub-networks optimization (Step 3) are also included in Table 6. An approximate solution with a cost of \$1.021 million was consistently located by the proposed method after the preconditioning optimization step, which is only 0.5% higher than the current best solution (\$1.016 million). However, this approximate solution was found only using 15,608 equivalent full N_I evaluations, which is only 10% of that required by the SDE. This shows that the proposed preconditioning optimization for the sub-networks (Step 3) is effective as it is able to specify promising regions for the final optimization of the sub-networks (Step 4) with great efficiency.

4.2. Case studies 2 and 3: Benchmark case studies (N_2 and N_3)

Two benchmark case studies including the New York Tunnels problem (NYTP: N_2) and the Hanoi problem (HP: N_3) have been used to demonstrate the effectiveness of the proposed method. The details of NYTP and HP case studies, including the head constraints, pipe costs and water demands are given by *Dandy et al.* [1996] and *Fujiwara and Khang* [1990] respectively. For the NYTP and HP case studies, the trees are viewed to be the sub-networks since the blocks are not applicable and the nodes connecting the trees with the other components of the network are viewed as sub-network cut nodes. The sub-networks and the directed augmented tree for the NYTP and HP case study are presented in Figure 5 and 6 respectively (the original NYTP and HP networks can be found in *Zheng et al.* [2011]).

For the NYTP case study, the optimization sequence for sub-networks is indicated by the directed augmented tree in Figure 5, with the S_2 and S_3 being optimized first, followed by the root sub-network S_1 ($\varphi(S_1)=\{S_2, S_3\}$). A series of heads with an interval of one foot were used for the sub-network cut nodes 9 and 12 during the preconditioning optimization for the sub-networks S_2 and S_3 ($H = \{272, 273, 274, \dots, 300\}$ feet for S_2 and $H = \{255, 256, 257, \dots, 300\}$ feet for S_3). The DE parameters used for the proposed method and computational simulation time for each sub-network are given in Table 7.

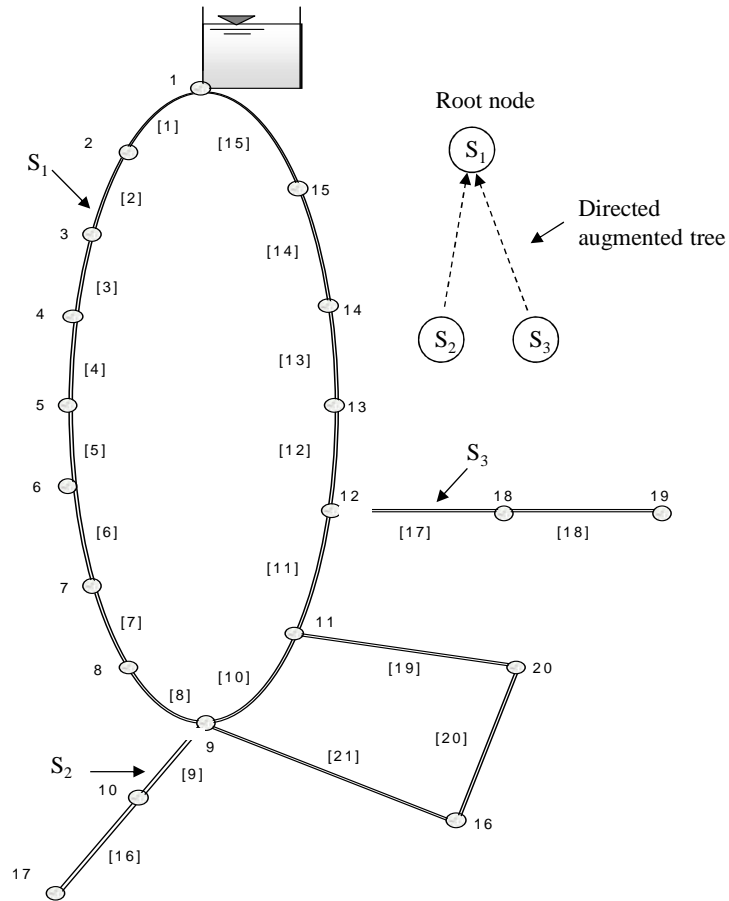


Figure 5 The full network, sub-networks and the directed augmented tree of the NYTP (N_2) network

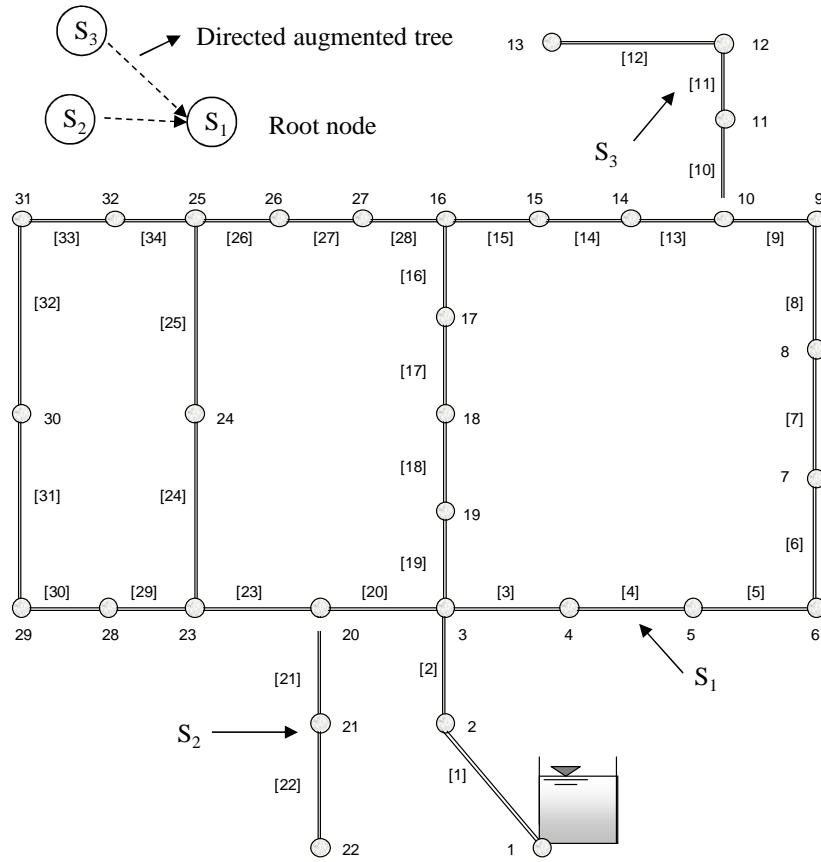


Figure 6 The full network, sub-networks and the directed augmented tree of the Hanoi (HP: N_3) network

Table 7 DE parameter values for each sub-network of the NYTP and HP case studies

Case study	Network	No. of decision variables and the search space size	DE parameter values	Maximum number of allowable evaluations	The computational time for 1000 simulations (seconds) ¹
NYTP	Full network	17 (1.94×10^{25})	-	-	0.95
	S_1	17 (2.95×10^{20})	$N=50, F=CR=0.5$	10,000	0.810
	S_2	2 (256)	$N=10, F=CR=0.5$	1,000	0.100
	S_3	2 (256)	$N=10, F=CR=0.5$	1,000	0.110
HP	Full network	34 (2.86×10^{26})	-	-	1.156
	S_1	29 (3.68×10^{22})	$N=80, F=0.7, CR=0.8$	50,000	0.908
	S_2	2 (36)	$N=10, F=CR=0.5$	1,000	0.140
	S_3	3 (216)	$N=10, F=CR=0.5$	1,000	0.141

¹1000 simulations were based on randomly selected network configuration and conducted on the same computer configuration (Pentium PC (Inter R) at 3.0 GHz).

The optimization results of the proposed graph decomposition optimization method are presented in Table 8. The previously published results for this case study are also included in Table 8 to enable a performance comparison with the proposed method. The current best known solution for the NYTP case study is \$38.64 million [Maier et al. 2003] and this best solution was found by the proposed method after the preconditioning optimization step (Step 3) with a success rate of 100% based on 100 runs starting with different random number seeds. The total computational overhead required by the proposed method has been converted to the equivalent number of full NYTP evaluations to enable the efficiency performance with other algorithms. The proposed method exhibits the best performance in terms of percent of trials with the best solution found and the efficiency for the NYTP case study as can be seen from Table 8. Based on 100 runs, the proposed method only required an average of 3,772 equivalent full network evaluations to find the current best known solution, which is significantly lower than those used by other methods shown in Table 8.

Table 8 Summary of the results of the proposed method and other algorithms applied to the NYTP (N_2) case study

Algorithm ¹⁰	No. of runs	Best solution (\$M)	Percent of trials with best solution found	Average cost (\$M)	Average evaluations to find first occurrence of the best solution
The proposed method ¹	100	38.64	100%	38.64	3,772 ⁸
NLP-DE ²	100	38.64	99%	38.64	8,277
GHEST ³	60	38.64	92%	38.64	11,464
HD-DDS ⁴	50	38.64	86%	38.64	47,000
Suribabu DE ⁵	300	38.64	71%	NA	5,492
Scatter Search ⁶	100	38.64	65%	NA	57,583
GA ⁷	100	38.64	45%	39.25	54,789

¹The results of the proposed graph decomposition optimization method after preconditioning sub-networks optimization (Step 3). ²Zheng et al. [2011]. ³Bolognesi et al. [2010]. ⁴Tolson et al. [2009]. ⁵Suribabu [2010]. ⁶Lin et al. [2007]. ⁷Zheng et al. [2012]. ⁸The total computational overhead required by proposed method has been converted to the equivalent number of full NYTP evaluations using the simulation time presented in Table 7.

The optimization sequence for sub-networks of the HP case study is shown in the directed augmented tree in Figure 6. Sub-networks S_2 and S_3 are optimized first and then the root sub-network S_1 ($\varphi(S_1)=\{S_2, S_3\}$) is optimized while incorporating the optimal solutions for S_2 and S_3 . A series of heads in the range of [30, 100] meters with an interval of one meter were used for the sub-network cut nodes 20 and 10

during the preconditioning optimization for S_2 and S_3 . The DE parameter values for the proposed method applied to sub-networks of the HP case study and the computational simulation time for each sub-network are shown in Table 7. Table 9 presents the optimization results of the proposed method applied to the HP case study and also the results obtained by previously published algorithms.

The current best known solution for the HP case study was first reported by *Reca and Martínez* [2006], with a cost of \$6.081 million. Similarly as for the NYTP case study, the proposed graph decomposition optimization method found the current best known solution for the HP case study after the preconditioning optimization step (Step 3). As can be seen from Table 9, the proposed method was able to locate the current best known solution for the HP case study 98% of the time based on 100 trials, which is higher than all the other algorithms presented in Table 9. In terms of efficiency, the proposed method also performed the best as it found the optimal solutions with an average of 26,540 equivalent full network evaluations, which is fewer than other algorithms in Table 9.

Table 9 Summary of the results of the proposed method and other algorithms applied to the HP (N_3) case study

Algorithm ¹⁰	No. of runs	Best solution (\$M)	Percent of trials with best solution found	Average cost (\$M)	Average evaluations to find first occurrence of the best solution
The proposed method ¹	100	6.081	98%	6.081	26,540 ⁸
NLP-DE ²	100	6.081	97%	6.082	34,609
Suribabu DE ³	300	6.081	80%	NA	48,724
Scatter Search ⁴	100	6.081	64%	NA	43,149
GHEST ⁵	60	6.081	38%	6.175	50,134
HD-DDS ⁶	50	6.081	8%	6.252	100,000
GA ⁷	100	6.112	0%	6.287	384,942

¹The results of the proposed graph decomposition optimization method after preconditioning sub-networks optimization (Step 3). ²*Zheng et al.* [2011]. ³*Suribabu* [2010]. ⁴*Lin et al.* [2007]. ⁵*Bolognesi et al.* [2010]. ⁶*Tolson et al.* [2009]. ⁷*Zheng et al.* [2012]. ⁸The total computational overhead required by proposed method has been converted to the equivalent number of full HP evaluations using the simulation time presented in Table 7.

Based on the results of two benchmark case studies (the NYTP (N_2) and HP (N_3)), it can be concluded that the proposed method produced the current best known performance in terms of both the solution quality and efficiency.

4.3. Case study 4: Network 4 (N_4) (237 decision variables)

Network four (N_4) was taken from a town in the southeast of China. N_4 has 237 pipes, one reservoir and 192 demand nodes. The head provided by the reservoir is 65 meters. The minimum pressure requirement for each demand node is 18 meters. The Hazen-Williams coefficient for each pipe is 130. A total of 14 pipes ranging from 150 mm to 1000 mm are used for this network design and the cost of each diameter was provided by *Kadu et al.* [2008]. The original network layout of N_4 is given in Figure 7 and the sub-networks and the directed augmented tree obtained by the proposed decomposition method are presented in Figure 8.

As shown in Figure 8, seven sub-networks were identified by the proposed method. The optimization process has to be taken based on the direction from the leaves to the root of the directed augmented tree (Figure 8(b)).

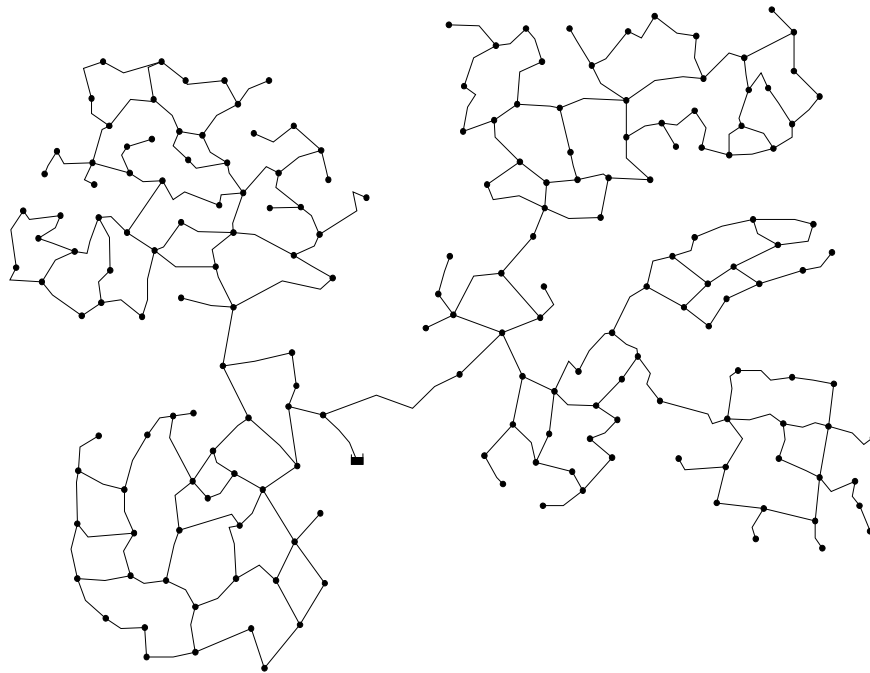


Figure 7 The original full network of N_4 case study

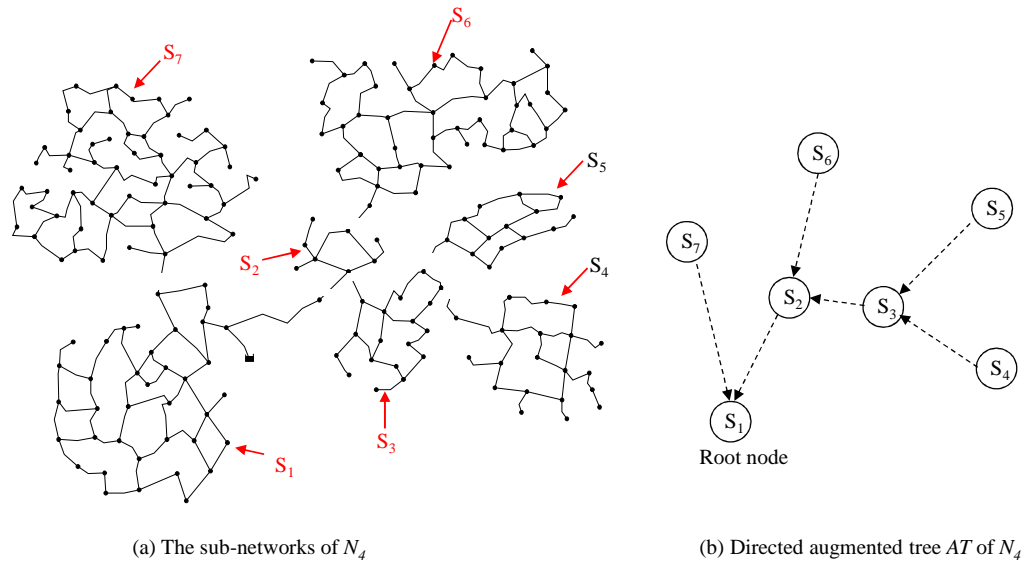


Figure 8 The sub-networks and the directed augmented tree of N_4 (a): the sub-networks. (b): the directed augmented tree

Table 10 presents the sizes of the networks (including the full network and sub-networks), the population sizes of the DE and GA and the computational time for simulating each network. Values of $F=0.3$ and $CR=0.7$ were selected for the SDE, and values $P_c=0.9$ and $P_m=0.005$ were selected for the GA based on an extensive parameter calibration phase. Values of $F=0.3$ and $CR=0.5$ were used for the DE applied to each sub-network in the proposed graph decomposition optimization method based on a preliminary parameter analysis. It is interesting to note from Table 10 that the total computational running time for hydraulically simulating each sub-network 1000 times is 8.75 seconds, which is only 31% of that required by 1000 original full network simulation.

The search space sizes for the original N_4 case study and each sub-network are included in Table 10. The original search space size for the whole network is $14^{237} \approx 4.29 \times 10^{271}$, while the search space for each sub-network is significantly reduced. Thus, the DE optimization for the sub-network requires a lesser number of population size (N) and the maximum number of allowable evaluations compared to the optimization for the original full N_4 network.

Table10 Evolutionary algorithm parameter values and the hydraulic simulation time for each sub-network and the full N_4

EAs	Network	No. of decision variables and the search space size	Population size (N)	Maximum number of allowable evaluations	The computational time for 1000 simulations (seconds) ¹
SDE	N_4	237 (4.29×10^{271})	500	5,000,000	28.20
GA	N_4	237 (4.29×10^{271})	500	5,000,000	28.20
DE used in the proposed method	S_1	51 (2.83×10^{58})	100	50,000	2.19
	S_2	9 (2.07×10^{10})	50	5,000	0.32
	S_3	21 (1.17×10^{24})	100	50,000	0.62
	S_4	23 (2.30×10^{26})	100	50,000	0.78
	S_5	18 (4.27×10^{20})	50	25,000	0.62
	S_6	52 (3.97×10^{59})	200	400,000	2.19
	S_7	63 (1.61×10^{72})	200	400,000	2.03

¹1000 simulations were based on randomly selected network configuration and conducted on the same computer configuration (Pentium PC (Inter R) at 3.0 GHz).

Ten different runs with different starting random number seeds were performed for the proposed method and the SDE applied to N_4 case study. The solutions are presented in Figure 9 and the statistical results of these solutions are given in Table 11. It should be noted that the number of evaluations given in Figure 9 for the proposed method is the equivalent number of full N_4 evaluations that was converted by the total computational running time of the proposed method. The computational time used for identifying the seven sub-networks is equivalent to 178 full N_4 evaluations.

As shown in Figure 9, the proposed method is able to find significantly better solutions than the SDE and GA after the final sub-network optimization (Step 4) with fewer number of equivalent evaluations. In addition, the optimal solutions produced by the proposed method are less scattered than those found by the SDE in terms of distribution. This implies that the proposed method was capable of consistently locating extremely similar or the same final optimal solutions with different starting random number seeds. The optimal solutions found by the proposed method after preconditioning optimization for the sub-networks (Step 3) were higher than those yielded by the SDE and the GA as displayed in Figure 9.

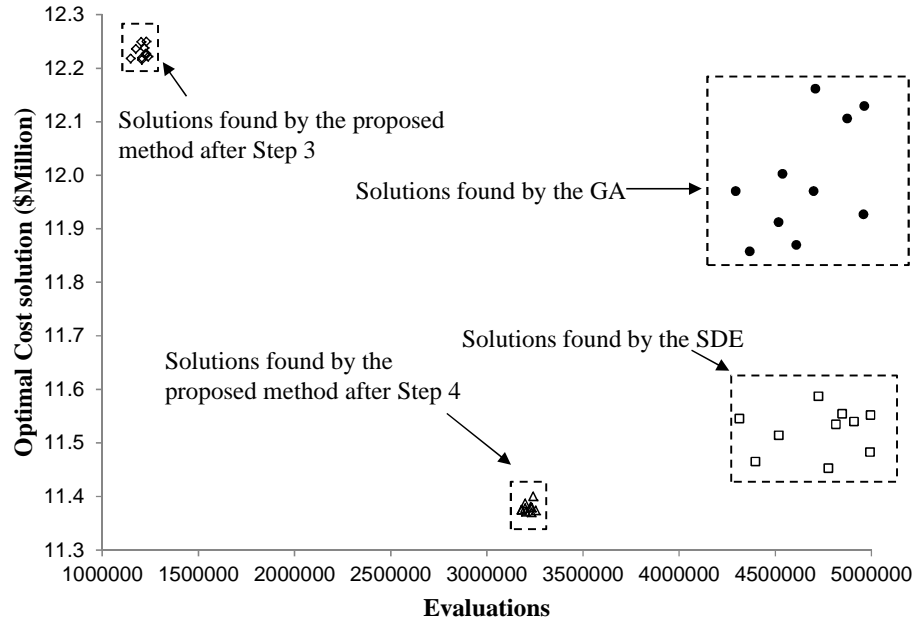


Figure 9 Solutions of the proposed method, the SDE and the GA applied to N_4 case study

Table 11 Algorithm performance for the N_4 case study

Algorithm	Number of different runs	Best solution found (\$M)	Percentage of trials with best solution found	Average cost solution (\$M)	Average number of equivalent evaluations to find best solution
Proposed method ¹	10	12.22	0%	12.23	1,208,324 ³
Proposed method ²	10	11.37	30%	11.38	3,215,685 ³
SDE ⁴	10	11.45	0%	11.52	4,730,200
GA ⁴	10	11.85	0%	11.99	4,654,000

¹The results of the proposed method after the preconditioning sub-network optimization (Step 3). ²The results of the proposed method after the final sub-network optimization (Step 4). ³The total computational overhead required by the proposed method has been converted to the equivalent number of the whole network (N_4) evaluations. ⁴Parameters were tuned.

As can be seen from Table 11, the proposed method after the final optimization of the sub-networks (Step 4) found the current best solution for N_3 case study with a cost of \$11.37 million, which is 0.7% and 4.2% lower than the best solutions yielded by the SDE and the GA respectively. The current best solution was found three times out of a total of ten different runs by the proposed method after Step 4. The average cost solution generated by the proposed method after Step 4 was \$11.38 million, which is only 0.09% higher than the current best solution while 1.2% and 5.4% lower than the average cost solutions of the SDE and the GA.

In terms of the average number of equivalent evaluations, the proposed method after the preconditioning sub-network optimization (Step 3) required only 26% of that used by the SDE. Although the solutions found by the proposed method after Step 3 were slightly worse than those located by the SDE and GA, they provided promising regions quickly to allow the further exploitation by the final optimization step (Step 4). After the final sub-network optimization of the proposed method (Step 4), the solution quality was substantially improved and the efficiency was still significantly better than the SDE and GA as shown in Table 11.

4.4. Case study 5: Network 5 (N_5) (433 decision variables)

A network (N_5) having 433 pipes and 387 demand nodes has been used in order to verify the effectiveness of the proposed method in terms of dealing with more large and complex networks. The network topology of N_5 was taken from Battle of the Water Networks II (BWN-II) presented in Water Distribution Systems Analysis Conference 2012. The pumps and valves in the original BWN-II network have been replaced by pipes as the aim of this paper is to demonstrate the utility of the proposed method in terms of optimizing the design for the pipes-only network. For this network, the head provided by the reservoir is 75 meters and the minimum pressure requirement for each demand node is 25 meters. The Hazen-Williams coefficient for each pipe is assumed to be 130. As the same for case study N_4 , 14 pipe choices are used for this network design. The layout of the original N_4 is given in Figure 10, and the decomposed sub-networks and the directed augmented tree (AT) are presented in Figure 11.

A total of 12 sub-networks were identified using the proposed method for the N_5 network as shown in Figure 11 (a). The optimization sequence for the 12 sub-networks is indicated by the directed augmented tree in Figure 11 (b). A SDE and a GA were also applied to the full N_5 and their parameter values have been fine-tuned. Values of $F = 0.3$ and $CR=0.8$ were selected for the SDE, and the $P_c=0.9$ and $P_m=0.003$ were used for the GA.

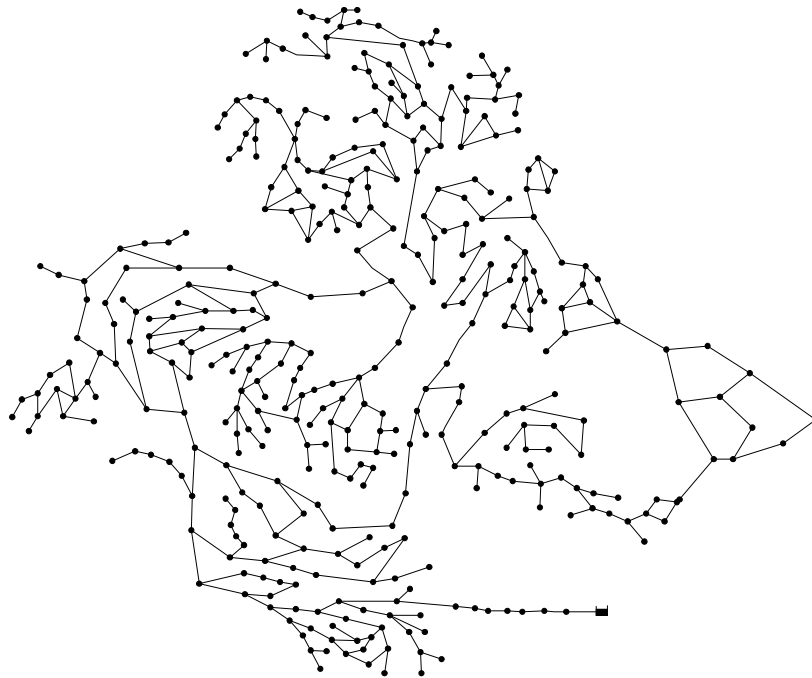


Figure 10 The original full network of N_5 case study

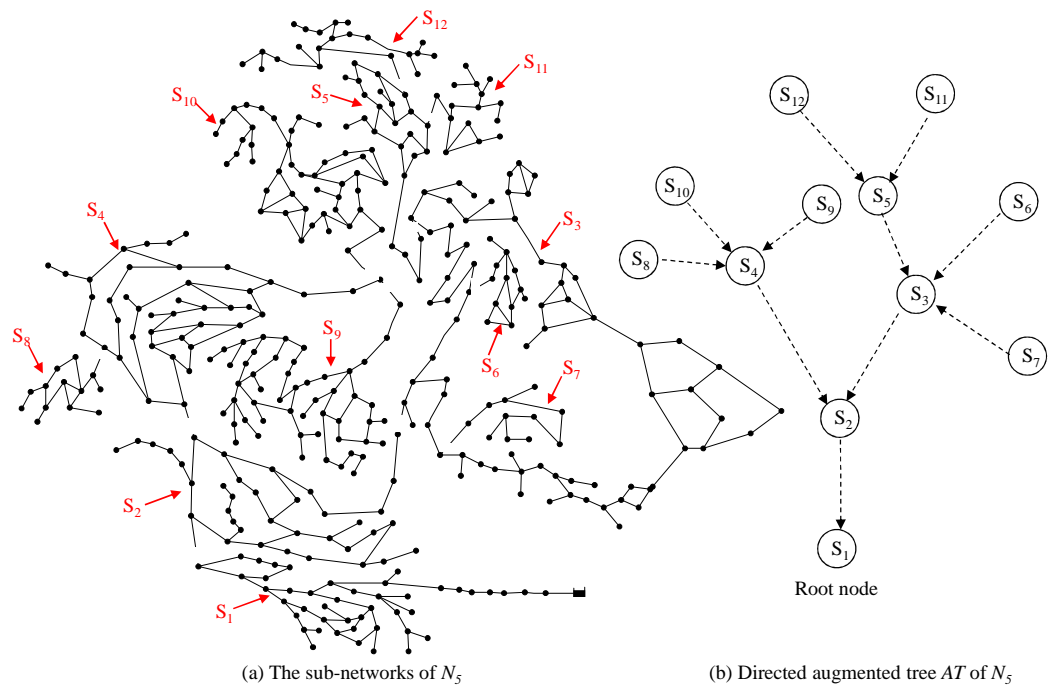


Figure 11. The sub-networks and the directed augmented tree of N_5 (a): the sub-networks. (b): the directed augmented tree

The sizes of the networks, the population sizes of the DE (including the SDE and the DE used in the proposed graph decomposition optimization method) and GA and the computational time for simulating each network are presented in Table 12. Values of $F=0.5$ and $CR=0.5$ were used for the DE applied to each sub-network in the proposed method. As can be seen from Table 12, for the N_5 case study, the total computational runtime for hydraulically simulating each sub-network 1000 times is 7.44 seconds, which is only 18% of that used by 1000 full network simulation. This indicates that the hydraulic simulation of the decomposed sub-networks is significantly faster than simulating the full network as a whole in terms of computational running time.

Table 12 Evolutionary algorithm parameter values and the hydraulic simulation time for each sub-network and the full N_5

EAs	Network	No. of decision variables and the search space size	Population size (N)	Maximum number of allowable evaluations	The computational time for 1000 simulations (seconds) ¹
SDE	N_5	433 (1.88×10^{496})	1000	10,000,000	42.06
GA	N_5	433 (1.88×10^{496})	1000	10,000,000	42.06
DE used in the proposed method	S_1	49 (1.44×10^{56})	200	200,000	0.72
	S_2	40 (7.00×10^{45})	200	200,000	0.61
	S_3	81 (6.86×10^{92})	200	500,000	2.13
	S_4	50 (2.02×10^{57})	200	200,000	0.81
	S_5	28 (1.23×10^{32})	100	100,000	0.30
	S_6	15 (1.56×10^{17})	50	50,000	0.23
	S_7	11 (4.05×10^{12})	50	50,000	0.14
	S_8	15 (1.56×10^{17})	50	50,000	0.23
	S_9	56 (1.52×10^{64})	200	200,000	0.92
	S_{10}	51 (2.83×10^{58})	200	200,000	0.74
	S_{11}	16 (2.18×10^{18})	50	50,000	0.22
	S_{12}	21 (1.17×10^{24})	100	100,000	0.27

¹1000 simulations were based on randomly selected network configuration and conducted on the same computer configuration (Pentium PC (Inter R) at 3.0 GHz).

For the N_5 case study, a total of ten different runs with different starting random number seeds were performed for the proposed method, the SDE and the GA. Figure 12 presents the solutions obtained by these three different optimization methods. The computational run time for each run of the proposed method has been converted to the equivalent number of full N_5 evaluations based on network simulation time in

Table 12. The computational time used for identifying the 12 sub-networks is equivalent to 215 full N_5 evaluations.

It may be clearly seen from Figure 12 that the proposed method after Step 4 was able to find lower cost solutions with significantly fewer number of full network evaluations compared to the SDE and GA. The optimal solutions found by the proposed graph decomposition optimization method after Step 3 are better than those obtained by the GA and comparable to those generated by the SDE, but with significantly improved efficiency. Similarly to that for the N_4 case study, the optimal solutions yielded by the proposed method for N_5 case study are closer to each other compared to the SDE and GA, showing greater robustness as similar cost solutions were found with different starting random number seeds.

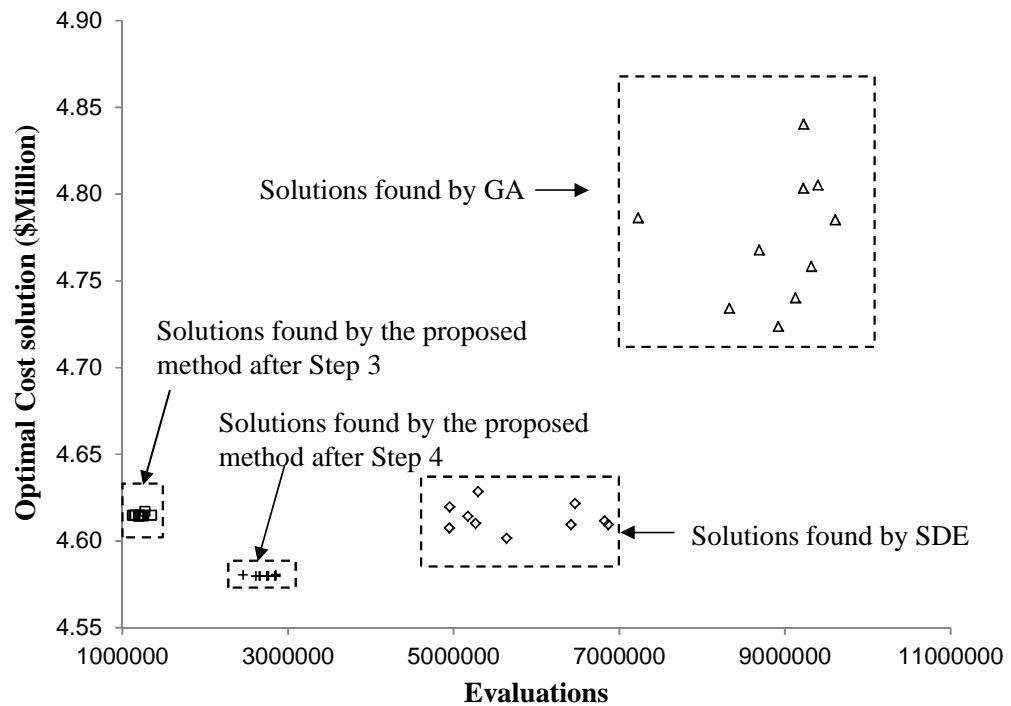


Figure 12 Solutions of the proposed method, the SDE and the GA applied to N_5 case study

Table 13 presents the statistical results of the proposed method, the SDE and GA. The current best solution was found by the proposed method after Step 4 with a cost

of \$4.57 million and this best solution was found eight times out of ten runs with different random number seeds. The best solutions yielded by the SDE and GA were \$4.60 million and \$4.72 million, which are 0.7% and 3.2% higher than the current best known solutions provided by the proposed method after Step 4. The proposed method exhibited the best performance in terms of comparing the efficiency to find optimal solutions as shown in Table 13. The average computational run time required by each run of the proposed method is equivalent to 2,720,668 full N_5 evaluations, which is 47% and 30% of those used by the SDE and GA.

Table 13 Algorithm performance for the N_5 case study

Algorithm	Number of different runs	Best solution found (\$M)	Percentage of trials with best solution found	Average cost solution (\$M)	Average number of equivalent evaluations to find best solution
Proposed method ¹	10	4.61	0%	4.61	1,220,924 ³
Proposed method ²	10	4.57	80%	4.58	2,720,668 ³
SDE ⁴	10	4.60	0%	4.61	5,786,300
GA ⁴	10	4.72	0%	4.77	8,909,500

¹The results of the proposed graph decomposition optimization method after the preconditioning sub-network optimization (Step 3). ²The results of the proposed method after the final sub-network optimization (Step 4). ³The total computational overhead required by the proposed method has been converted to the equivalent number of the whole network (N_5) evaluations. ⁴Parameters were tuned.

Interestingly, the proposed method after Step 3 was able to find lower cost solutions than the GA but with approximately five times the convergence speed. The best solutions found by the proposed method after Step 3 were only 0.2% higher than the best solution given by the SDE (the average costs of ten solutions for both are the same as shown in Table 13), while the average number of evaluations required by the proposed method after Step 3 is only 21% of that used by the SDE.

4.5. Summary of results

Traditionally, WDNs are optimized as a whole when they are being designed. In the proposed method, however, the WDN is treated as a graph and decomposed with a graph theory algorithm into sub-networks. A directed augmented tree is built for the decomposed network and used to specify the optimization sequence for the sub-networks. Optimization takes place from the leaves to the root based on the directed

augmented tree and is both sequential and cumulative. Therefore, when the optimization algorithm runs through the root sub-network, which is the last to be optimized, it brings with it all the best solutions from the sub-networks and hence produces the optimal solution for the original full network.

The proposed approach takes advantage of the fact that the evolutionary algorithm (DE in this paper) is effective in exploring a relatively small search space. As the number of decision variables for each sub-network is significantly less than the original whole network, the DE is able to exploit the substantially reduced search space quickly and effectively. This allows good quality optimal solutions for each sub-network to be found with great efficiency.

A pre-conditioning sub-network optimization step (Step 3) is used in the proposed method to identify the optimal head range for the sub-network cut nodes. The final sub-network optimization is then conducted using a series of heads within the specified optimal head range with a relatively small interval (0.1 meters in the proposed method) in order to find further better solutions. The results of the five case studies show that the preconditioning sub-network optimization found the optimal head range for each sub-network cut node effectively, and the final sub-network optimization runs on the pre-conditioned sub-networks were able to generate improved quality solutions.

In spite of conducting multiple DE runs on each sub-network, the total efficiency of the proposed method is still better than the SDE and GA. This can be attributed to the fact (i) the population size and the maximum allowable evaluations required by the DE applied to the sub-network optimization were significantly smaller than the SDE applied to the original whole network and (ii) the computational time for simulating the sub-networks was considerably reduced compared to the original whole network.

An important advantage of the proposed method is that with multiple sub-networks in place, optimization of the water distribution systems can be undertaken using parallel computing technology. For the optimization of sub-networks at leaves and in the

middle of the directed augmented tree, parallel computing technology can be employed to conduct the optimization for different heads at the sub-network cut nodes simultaneously. In addition, all the sub-networks at the leaves can also be optimized separately and simultaneously by parallel computing technology. As such, the efficiency of the whole optimization process can be massively improved in terms of computation time. This is a significant benefit when designing a real-world WDS, for which a large number of pipes and demand nodes are normally involved.

Another observation can be made in this study is that the SDE with the fine-tuned parameter values consistently outperformed the GA with calibrated parameter values for five case studies with the number of decision variables ranging from 21 to 433. This agrees with the conclusion made by *Zheng et al.* [2012] in that the DE algorithm appears to be better suited for optimizing water network designs than the widely used GA algorithm.

5. Conclusion and future work

A novel optimization approach for WDS design has been developed and described in this paper. In the proposed method, a graph theory algorithm is employed to identify the sub-networks for the original full water network. The sub-networks, rather than the original full water network, are individually optimized by a DE in a predetermined sequence. Five case studies have been used to verify the effectiveness of the proposed method. A DE and a GA have also been applied to the full network for each case study (SDE) to enable a performance comparison with the proposed method.

The results show that the proposed method is able to find the same lowest cost solution for the relatively small case study, while producing better optimal solutions for the relatively larger case studies than the SDE and GA. It was also noted that the proposed method was able to find extremely similar optimal solutions, if not identical, for each run with different starting random number seeds. This

demonstrates the great robustness of the proposed method. In terms of efficiency, the proposed method significantly outperformed the SDE and GA for each case study. In addition, the proposed method exhibits the current best known performance in terms of efficiency in locating the best known solutions for the NYTP and HP case studies. Another substantial benefit of the proposed method is that it provides a way to exploit parallel computing techniques for the design optimization of a WDS.

It should be noted that the proposed method presented in this paper is not applicable to the networks for which sub-network cut nodes do not exist (i.e. for networks that cannot be decomposed). However, it is very common for a water network to have multiple blocks and multiple trees in practice (in other words- that the network is decomposable) and the proposed method has advantages in efficiently finding good quality optimal solutions for this common type of network compared to other optimization methods as demonstrated in this paper.

The future research scope of the proposed method includes (i) applying the proposed method to more complex water networks that may include multiple reservoirs, pumps, valves, storage facilities and pipes; and (ii) extending the proposed method for multi-objective WDS optimization design.

References

- Alperovits, E., and U. Shamir (1977), Design of water distribution systems, *Water Resour. Res.*, 13(6), 885-900, doi:10.1029/WR013i006p00885.
- Dandy, G. C., A. R. Simpson, and L. J. Murphy (1996), An improved genetic algorithm for pipe network optimization, *Water Resour. Res.*, 32(2), 449-457, doi:10.1029/95wr02917.
- Deuerlein, J. W. (2008), Decomposition model of a general water supply network graph, *Journal of Hydraulic Engineering*, 134(6), 822-832, doi:10.1061/(ASCE)0733-9429(2008)134:6(822).
- Eiger, G., U. Shamir, and A. Ben-Tal (1994), Optimal design of water distribution networks, *Water Resour. Res.*, 30(9), 2637-2646, doi: 10.1029/94wr00623.

- Fujiwara, O., and D. B. Khang (1990), A two-phase decomposition method for optimal design of looped water distribution networks, *Water Resour. Res.*, 26(4), 539-549, doi: 10.1029/WR026i004p00539.
- Geem, Z. W., J. H. Kim, and G. V. Loganathan (2002), Harmony search optimization: application to pipe network design, *Int. J. Model. Simulat.*, 22(2), 125-133.
- Gupta, R., and T. D. Prasad (2000), Extended use of linear graph theory for analysis of pipe networks, *Journal of Hydraulic Engineering*, 126(1), 56-62, doi:10.1061/(asce)0733-9429(2000)126:1(56).
- Keedwell, E., and S.-T. Khu (2005), A hybrid genetic algorithm for the design of water distribution networks, *Engineering Applications of Artificial Intelligence*, 18(4), 461-472, doi:10.1016/j.engappai.2004.10.001.
- Kessler, A., L. Ormsbee, and U. Shamir (1990), A methodology for least-cost design of invulnerable water distribution networks, *Civil Engineering Systems*, 7(1), 20-28.
- Kadu, M. S., G. Rajesh, and P. R. Bhave (2008), Optimal design of water networks using a modified genetic algorithm with reduction in the search space, *J. Water Resources Planning and Management*, 134(2), 147-160, doi:10.1061/(ASCE)0733-9496(2008)134:2(147).
- Krapivka, A., and A. Ostfeld (2009), Coupled genetic algorithm-linear programming scheme for least-cost pipe sizing of water-distribution systems, *J. Water Resources Planning and Management*, 135(4), 298-302, doi:10.1061/(ASCE)0733-9496(2009)135:4(298).
- Maier, H. R., A. R. Simpson, A. C. Zecchin, W. F. Foong, K. Y. Phang, H. Y. Seah, and C. L. Tan (2003), Ant colony optimization for the design of water distribution systems, *J. Water Resources Planning and Management*, 129(3), 200-209, doi:10.1061/(ASCE)0733-9496(2003)129:3(200).
- Montesinos, P., A. García, and J. L. Ayuso (1999), Water distribution network optimization using modified genetic algorithm, *Water Resour. Res.*, 35(11), 3467-3473, doi: 10.1029/1999wr900167.
- Reca, J., and J. Martínez (2006), Genetic algorithms for the design of looped irrigation water distribution networks, *Water Resour. Res.*, 42, W05416, doi:10.1029/2005WR004383.
- Reis, L. F. R., F. T. Bessler, G. A. Waters, and D. Savic (2006), Water supply reservoir operation by combined genetic algorithm-linear programming (GA-LP) Approach, *Water Resources Management*, 20: 227-255, doi:10.1007/s11269-006-8049-z.

- Savic, D. A., and G. A. Walters (1995), An evolution program for optimal pressure regulation in water distribution networks, *Engineering Optimization*, 24(3), 197-219, doi: 10.1080/03052159508941190.
- Simpson, A. R., G. C. Dandy, and L. J. Murphy (1994), Genetic algorithms compared to other techniques for pipe optimization, *J. Water Resources Planning and Management*, 120(4), 423-443, doi: 10.1061/(ASCE)0733-9496(1994)120:4(423).
- Sonak, V. V., and P. R. Bhave (1993), Global optimum tree solution for single-source looped water distribution networks subjected to a single loading pattern, *Water Resour. Res.*, 29(7), 2437-2443, doi: 10.1029/93WR00289.
- Suribabu, C. R., and T. R. Neelakantan (2006), Design of water distribution networks using particle swarm optimization, *Urban Water J.*, 3(2), 111-120, doi:10.1080/15730620600855928.
- Suribabu, C. R. (2010), Differential evolution algorithm for optimal design of water distribution networks, *Journal of Hydroinformatics*, 12 (1), 66-82.
- Tarjan, R. (1972), Depth-first search and linear graph algorithms, *SIAM Journal on Computing*, 1(2), 146-160.
- Tolson, B. A., M. Asadzadeh, H. R. Maier, and A. C. Zecchin (2009), Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization, *Water Resour. Res.*, 45, W12416, doi:10.1029/2008WR007673.
- Tu, M.-Y., F. T. C. Tsai, and W. W. G. Yeh (2005), Optimization of water distribution and water quality by hybrid genetic algorithm, *Journal of Water Resources Planning and Management*, 131(6), 431-440, doi:10.1061/(asce)0733-9496(2005)131:6(431).
- van Zyl, J. E., D. A. Savic, and G. A. Walters (2004), Operational optimization of water distribution systems using a hybrid genetic algorithm, *Journal of Water Resources Planning and Management*, 130(2), 160-170, doi:10.1061/(asce)0733-9496(2004)130:2(160).
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2011), A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems, *Water Resour. Res.*, 47(8), W08531, doi: 10.1029/2011wr010394.

Chapter 10. Conclusions and Future Work

This thesis has proposed some advanced optimization techniques for water distribution system (WDS) optimization. These novel techniques have been assessed using both the benchmark case studies and real-world water networks, and the results show that they are able to find good quality optimal solutions for WDSs with great efficiency. Thus, this research provides advanced optimization approaches capable of outperforming existing optimization techniques for use by practising water engineers when designing new water systems or rehabilitating existing water distribution systems.

10.1 Thesis outcomes

The main contributions of this thesis are outlined in Section 1.3 of Chapter 1 (Introduction). The outcomes of this research are summarised below.

Chapter 2 presented a **detailed review** of the optimization techniques that have previously been used to optimize WDSs. The analysis for each type of optimization technique is outlined in Chapter 2 and the research gaps in terms of WDS optimization are identified in Section 2.6 of Chapter 2.

Chapters 3 and 4 outlined **two new genetic algorithm (GA) variants for WDS optimization**. These include a dynamically expanding choice table GA (Chapter 3) and a non-crossover dither creeping mutation GA (Chapter 4). These two GA variants were demonstrated to be more effective than the traditional GAs in terms of optimizing the design of WDSs. This is the first work to develop a non-crossover GA for WDS optimization.

Chapter 5 introduced a new **self-adaptive differential evolution algorithm (SADE) for WDS optimization**. In addition to the self-adaption strategy, a new convergence criterion is proposed in order to avoid the pre-specification of the computational budget for the DE run. It has been demonstrated that the proposed convergence criterion presented in Chapter 5 is able to avoid computational waste. The SADE only needs to tune the population size parameter when optimizing the design of a WDS. In addition, an approximate heuristic is described in Chapter 5 that

is able to determine the appropriate population size for the SADE applied to a new WDS case study. The proposed SADE provides a robust optimization tool for the optimization of the design of WDSs (or rehabilitation of an existing WDS).

Chapters 6 and 7 presented **two novel hybrid optimization models for WDS optimization**, which are the NLP-DE method (Chapter 6) and the BLP-DE approach (Chapter 7). In **Chapter 6**, a shortest-distance tree is proposed to decompose the looped water network and an efficient graph theory algorithm is used to determine the shortest-distance tree. The NLP-DE method proposed in Chapter 6 differs from the traditional hybrid optimization models. In the traditional hybrid optimization models, EAs have been used to determine the regions of optimal solutions, whereas deterministic methods (such as LP) have been used to further explore the interior of these regions identified by EAs. However, in the proposed NLP-DE model, NLP is used to identify the approximate region of the optimal solution, while an EA is employed to further search the interior of the region.

The utility of the NLP-DE method has been verified using four WDS case studies with the number of decision variables ranging from 21 to 454 (two of them are real-world WDSs) in Chapter 6. The consistent superior performance of the proposed optimization approach on four case studies illustrates that the proposed methodology is well suited for the least-cost design of WDSs.

Chapter 7 outlined a completely new optimization methodology that applies different optimization techniques to optimize different components of the water network. In the proposed BLP-DE method given in Chapter 7, the deterministic method BLP is only used to deal with the optimization of the trees (no loops) and DE is employed to optimize the core (loops are involved). As such, the proposed BML-DE method makes good use of both types of optimization techniques, which are deterministic methods suitable for tree network optimization, and the EAs are effective when exploring relatively small search spaces. (The search space of the core is significantly smaller than that of the full network since the trees are removed.)

Chapter 8 presented a **complete new water network decomposition concept**-optimal source partitioning cut-set in which complex water networks with multiple supply sources are decomposed into sub-networks based on the number of supply

sources. A multi-stage optimization technique is developed to optimize the design for WDSs, which is also the first known work in the field of WDS optimization. The new methodology (i.e., decomposition followed by multi-stage optimization) presented in Chapter 8 is demonstrated to be extremely efficient and effective in finding optimal solutions for real-world sized water networks.

Chapter 9 outlined another completely novel methodology for water network optimization, in which the original water network is not optimized as a whole **but the whole is optimized by optimizing the separate parts**. In Chapter 9, a new sub-network identification concept is proposed to decompose the original full network into sub-networks based on the connectivity of the network's components. The sub-networks are optimized separately and combined to form the final solution for the whole water network by use of solution choice tables.

10.2 Recommendations of optimization algorithms

In this thesis, a total of four advanced optimization methods have been developed including the NLP-DE methods (Chapter 6), the BLP-DE approach (Chapter 7), the decomposition and multi-stage optimization method (Chapter 8) and a graph decomposition based optimization method (Chapter 9). The three variants of the evolutionary algorithms presented in Chapters 3 to 5 are modifications of the existing algorithms.

It is noted that the standard differential evolution (SDE), rather than the self-adaptive differential evolution (SADE) algorithm described in Chapter 5, was used in Chapters 6 to 9, resulting in a need to tune the parameter values for the proposed optimization algorithms. The use of SDE is primarily due to the fact that the experimental runs for Chapters 6 to 9 were finished before the publication of the paper on the SADE algorithm (Zheng et al. (2012a) in the Journal of Computing in Civil Engineering). In addition, the focus of the methods described in Chapters 6 to 9 is the development the hybrid optimization techniques and graph decomposition methods to facilitate the optimization for WDSs.

However, for future applications it is recommended that the SADE algorithm be used to replace the SDE algorithm in practice for the optimization frameworks presented in Chapters 6 to 9. This is because: (i) the SADE algorithm developed in this research has been demonstrated to perform similarly (if not better) with the SDE with fine-tuned parameter values; and (ii) that is no need to tune parameter values for the SADE algorithm and hence its uses will remove the need for an extensive parameter-calibration process.

It is necessary to make a recommendation as to the best of the developed algorithms to adopt when dealing with a given water network. The recommendation is dependent on the property of the water network that is to be optimized, and is given as follows:

- (1) It is recommended that the NLP-SADE algorithm presented in Chapter 6 be employed to deal with single reservoir water networks.
- (2) For water networks having multiple trees (such as the trunk main distribution system), it is recommended to use the BLP-SADE algorithm described in Chapter 7. In addition, the BLP-SADE algorithm has been demonstrated to effectively deal with water networks with multiple demand loadings, which is normally the case for the trunk main distribution system.
- (3) For multi-reservoir water networks, such as an irrigation network or a regional supply system with multiple tanks, it is recommended to adopt the decomposition and multi-stage optimization methods presented in Chapter 8.
- (4) For single-reservoir water networks with a number of different blocks connected by bridges (as for regional supply systems), the graph decomposition based optimization framework described in Chapter 9 is recommended to be used to conduct the design optimization.

10.3 Scope for future work

Future work includes:

1. The optimization techniques developed in this research have been verified using real-world water networks (100 pipes or more). However, these real-world networks are relatively simple as only pipes are involved. Extension of the proposed optimization techniques to deal with more complex water networks that include pumps, valves and tanks should be conducted in the future.
2. The optimization techniques proposed in this thesis have been demonstrated to be effective for single objective optimization of WDSs. It would be appropriate to extend these optimization techniques to deal with multi-objective optimization for WDSs, for which, in addition to the network cost, reliability or greenhouse gases should also be considered.
3. Several optimization techniques and water network decomposition concepts were used to optimize the design of WDSs in the research described in this thesis. Implementing these methods to tackle other water network management problems, such as leakage hotspot detection, optimal valve operation, contaminant detection and operational optimization problems for WDSs, should also be considered in the future.
4. Although DE is used in two proposed hybrid optimization methods (Chapters 6 and 7) and two proposed advanced optimization techniques (Chapters 8 and 9) in this research, other EAs, such as GAs, and Ant Colony Optimization (ACO), could also be implemented in these proposed optimization frameworks.

References

- Alperovits, E., and Shamir, U. (1977). "Design of optimal water distribution systems." *Water Resources Research*, 13(6), 885-900.
- Bolognesi, A., Bragalli, C., Marchi, A., and Artina, S. (2010). "Genetic heritage evolution by stochastic transmission in the optimal design of water distribution networks." *Advances in Engineering Software*, 41(5), 792-801.
- Broad, D. R., Dandy, G. C., and Maier, H. R. (2005). "Water distribution system optimization using metamodels." *Journal of Water Resources Planning and Management*, 131(3), 172-180.
- Calhoun, C. A (1981). "Optimization of pipe systems by linear programming." *Control of Flow in Closed Conduits*, J. P. Tullis, ed., Colorado State University, Fort Collins, Colorado, 175-192.
- Cisty, M. (2010). "Hybrid genetic algorithm and linear programming method for least-cost design of water distribution systems." *Water Resources Management*, 24(1), 1-24.
- Dandy, G. C., Simpson, A. R., and Murphy, L. J. (1996). "An improved genetic algorithm for pipe network optimization." *Water Resources Research*, 32(2), 449-458.
- Dandy, G. C., Wilkins, A., and Rohrlach, H. (2010). "A methodology for comparing evolutionary algorithms for optimizing water distribution systems." *Proceedings of the 12th Water Distribution System Analysis Symposium*, Tucson, Arizona
- Deb, K. (2000). "An efficient constraint handling method for genetic algorithms." *Computer Methods in Applied Mechanics and Engineering*, 186(2-4), 311-338.
- Deuerlein, J. W. (2008). "Decomposition model of a general water supply network graph." *Journal of Hydraulic Engineering*, 134(6), 822-832.
- Eusuff, M. M., and Lansey, K. E. (2003). "Optimization of water distribution network design using the shuffled frog leaping algorithm." *Journal of Water Resources Planning and Management*, 129(3), 210-225.
- Fujiwara, O., Jenchaimahakoon, B., and Edirishinghe, N. C. P. (1987). "A modified linear programming gradient method for optimal design of looped water distribution networks." *Water Resources Research*, 23(6), 977-982.
- Fujiwara, O., and Khang, D. B. (1990). "A two-phase decomposition method for optimal design of looped water distribution networks." *Water Resources Research*, 26(4), 539-549.

- Geem, Z. W., J. H. Kim, and G. V. Loganathan (2002). "Harmony search optimization: application to pipe network design." *Int. J. Model. Simulat.*, 22(2), 125-133.
- Grefenstette, J. J. (1987). *Incorporating problem specific knowledge into genetic algorithms*. In: Davis, L. (Ed.), *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann, Los Altos, CA.
- Gupta, R., and Prasad, T. D. (2000). "Extended use of linear graph theory for analysis of pipe networks." *Journal of Hydraulic Engineering*, 126(1), 56-62.
- Harik, G. R., D. E. Goldberg (2000). "Linkage learning through probabilistic expression." *Computer Methods in Applied Mechanics and Engineering*, 186(2), 295-310.
- Haghighi, A., Samani, H., and Samani, Z. (2011). "GA-ILP method for optimization of water distribution networks." *Water Resources Management*, 25(7), 1791-1808.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Mich.
- Kadu, M. S., Gupta, R., and Bhave, P. R. (2008). "Optimal design of water networks using a modified genetic algorithm with reduction in search space." *Journal of Water Resources Planning and Management*, 134(2), 147-160.
- Kessler, A., Ormsbee, L., and Shamir, U. (1990). "A methodology for least-cost design of invulnerable water distribution networks." *Civil Engineering Systems*, 7(1), 20-28.
- Keedwell, E., and Khu, S.-T. (2006). "A novel evolutionary meta-heuristic for the multi-objective optimization of real-world water distribution networks." *Engineering Optimization*, 38(3), 319 - 333.
- Krapivka, A., and Ostfeld, A. (2009). "Coupled genetic algorithm---linear programming scheme for least-cost pipe sizing of water-distribution systems." *Journal of Water Resources Planning and Management*, 135(4), 298-302.
- Lansey, K. E., and Mays, L. W. (1989). "Optimization model for water distribution system design." *Journal of Hydraulic Engineering*, 115(10), 1401-1418.
- Lippai, I., Heaney, J. P., and Laguna, M. (1999). "Robust water system design with commercial intelligent search optimizers." *Journal of Computing in Civil Engineering*, 13(3), 135-143.

- Lin, M.-D., Liu, Y.-H., Liu, G.-F., and Chu, C.-W. (2007). "Scatter search heuristic for least-cost design of water distribution networks." *Engineering Optimization*, 39(7), 857 - 876.
- Loganathan, G. V., Greene, J. J., and Ahn, T. J. (1995). "Design heuristic for globally minimum cost water-distribution systems." *Journal of Water Resources Planning and Management*, 121(2), 182-192.
- Liu, J., and Lampinen, J. (2005). "A fuzzy adaptive differential evolution algorithm." *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 9(6), 448-462.
- Martínez, J. B. (2008). "Discussion of "Optimization of water distribution networks using integer linear programming" by Hossein M. V. Samani and Alireza Mottaghi." *Journal of Hydraulic Engineering*, 134(7), 1023-1024.
- Maier, H. R., Simpson, A. R., Zecchin, A. C., Foong, W. K., Phang, K. Y., Seah, H. Y., and Tan, C. L. (2003). "Ant colony optimization for design of water distribution systems." *Journal of Water Resources Planning and Management*, 129(3), 200-209.
- Morgan, D. R., and Goulter, I. C. (1985). "Optimal urban water distribution design." *Water Resour. Res.*, 21(5), 642-652.
- Mohan, S., and Babu, K. S. J. (2010). "Optimal water distribution network design with honey-bee mating optimization." *Journal of Computing in Civil Engineering*, 24(1), 117-126.
- Nicklow, J., Reed, P., Savic, D., Dessalegne, T., Harrell, L., Chan-Hilton, A., Karamouz, M., Minsker, B., Ostfeld, A., Singh, A., Engineering, E. Z. A. T. C. o. E. C. i. E., and Water, R. (2010). "State of the art for genetic algorithms and beyond in water resources planning and management." *Journal of Water Resources Planning and Management*, 136(4), 412-432.
- Perelman, L., and Ostfeld, A. (2007). "An adaptive heuristic cross-entropy algorithm for optimal design of water distribution systems." *Engineering Optimization*, 39(4), 413 - 428.
- Price, V. K., Storn, M. R., and Lampinen, A. J. (2005). *Differential evolution: A practical approach to global optimization*, Springer, Berlin.
- Quindry, G. E., Brill, E. D., and Liebman, J. C. (1981). "Optimization of looped water distribution systems." *Journal of the Environmental Engineering Division*, 107(4), 665-679.
- Rossman, L. A. (2000), *EPANET 2-User manual*, National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency, Cincinnati, OH.

- Samani, H. M. V., and Mottaghi, A. (2006). "Optimization of water distribution networks using integer linear programming." *Journal of Hydraulic Engineering*, 132(5), 501-509.
- Savic, D. A., and Walters, G. A. (1995). "An evolution program for optimal pressure regulation in water distribution networks." *Engineering Optimization*, 24(3), 197-219.
- Savic, D., and Cunha, M. d. C. (2008). "Discussion of "Optimization of water distribution networks using integer and linear programming" by Hossein M. V. Samani and Alireza Mottaghi." *Journal of Hydraulic Engineering*, 134(7), 1024-1025.
- Simpson, A. R., Dandy, G. C., and Murphy, L. J. (1994). "Genetic algorithms compared to other techniques for pipe optimization." *Journal of Water Resources Planning and Management*, 120(4), 423-443.
- Sonak, V. V., and Bhave, P. R. (1993). "Global optimum tree solution for single-source looped water distribution networks subjected to a single loading pattern." *Water Resources Research*, 29(7), 2437-2443.
- Stephenson, D. (1984). "Pipeflow analysis: optimum design of branched pipe networks by linear programming." *Developments in Water Science*, Elsevier 55-69.
- Storn, R., and Price, K. (1995). "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous space." *Technical Report*, International Computer Science Institute, Berkeley, CA.
- Suribabu, C. R., and Neelakantan T. R. (2006). "Design of water distribution networks using particle swarm optimization." *Urban Water J.*, 3(2), 111-120.
- Suribabu, C. R. (2010). "Differential evolution algorithm for optimal design of water distribution networks." *Journal of Hydroinformatics*, 12(1), 66-82.
- Tolson, B. A., Asadzadeh, M., Maier, H. R., and Zecchin, A. (2009). "Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization." *Water Resources Research*, 45(12), W12416.
- Vairavamoorthy, K., and Ali, M. (2000). "Optimal design of water distribution systems using genetic algorithms." *Computer-Aided Civil and Infrastructure Engineering*, 15(5), 374-382.
- Vairavamoorthy, K., and Ali, M. (2005). "Pipe index vector: A method to improve genetic-algorithm-based pipe optimization." *Journal of Hydraulic Engineering*, 131(12), 1117-1125.

- Vasan, A., and Simonovic, S. P. (2010). "Optimization of water distribution network design using differential evolution." *Journal of Water Resources Planning and Management*, 136(2), 279-287.
- Vesterstrom, J., and Thomsen, R. (2004) "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems." *Proc., Evolutionary Computation, CEC2004. Congress on, 1980-1987 Vol.1982*.
- Wu, Z. Y., and Simpson, A. R. (2001). "Competent genetic-evolutionary optimization of water distribution systems." *Journal of Computing in Civil Engineering*, 15(2), 89-101.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2010). "A method for assessing the performance of genetic algorithm optimization for water distribution design." *Proc., Water Distribution System Analysis 2010 (WDSA2010)*, ASCE, Tucson, AZ, USA.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2011a). "A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems." *Water Resources Research*, 47(8), W08531.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2011b). "Dynamically expanding choice-table approach to genetic algorithm optimization of water distribution systems." *Journal of Water Resources Planning and Management*, 137(6), 547-551.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2011c). "Performance study of differential evolution with various mutation strategies applied to water distribution system optimization." *Proc., World Environmental and Water Resources Congress 2011*, ASCE, Palm Springs, California.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2011d). "Optimal rehabilitation for large water distribution systems using genetic algorithms." *Proc., Australia Water Association (AWA) OzWater 2011*, Adelaide, SA, Australia.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2011e). "Parametric analysis of differential evolution algorithm applied to water distribution system optimization." *Proc., 11th International Conference on Computing and Control for the Water Industry (CCWI 2011)*, Exeter, UK.
- Zheng, F., A. C. Zecchin and A. R. Simpson (2012a). "A self adaptive differential evolution algorithm applied to water distribution system optimization." *Journal of Computing in Civil Engineering*, doi:10.1061/(ASCE)CP.1943-5487.0000208.
- Zheng, F., A. R. Simpson, A. C. Zecchin and J. Deuerlein (2012b). "A graph decomposition based approach for water distribution network optimization." *Water Resources Research*, submitted June 2012.

- Zheng, F., A. R. Simpson, and A. C. Zecchin (2012c). "A combined binary linear programming and differential evolution algorithm approach for water distribution system optimization." *Journal of Water Resources Planning and Management*, submitted June 2012.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2012d). "A decomposition and multi-stage optimization approach applied to optimization of water distribution systems with multiple sources." *Water Resources Research*, doi:10.1029/2012WR013160.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2012e). "A performance comparison of differential evolution and genetic algorithm variants applied to water distribution system optimization." *Proc., World Environmental & Water Resources Congress (EWRI 2012)*, Albuquerque, New Mexico.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2012f). "A new hybrid optimization model for water distribution system optimization." *Proc., Water Distribution System Analysis 2012 (WDSA 2012)*, ASCE, Adelaide, SA, Australia.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2012g). "A performance comparison between differential evolution variants and genetic algorithms for water network optimization." *Proc., Water Distribution System Analysis 2012 (WDSA 2012)*, ASCE, Adelaide, SA, Australia.
- Zheng, F., A. R. Simpson, A. C. Zecchin, M. F. Lambert (2013). "A non-crossover dither creeping mutation genetic algorithm for pipe network optimization." *Journal of Water Resources Planning and Management*, doi: 10.1061/(ASCE)WR.1943-5452.0000351.