Particle Swarm Optimization: Theoretical analysis, modifications, and applications to constrained optimization problems

Mohammadreza Bonyadi, B.Sc., M.Sc.

A thesis submitted for the degree of
Doctor of Philosophy (Ph.D.)

School of Computer Science
Faculty of Engineering, Computer, and Mathematical Sciences
The University of Adelaide



October 2014

I dedicate this thesis to my lovely wife, Maryam, and my adorable parents, Shahrokh and Mahvash.

Contents

Abstract ii

Statement of originality iv

Acknowledgement v

Statements of authorships vi

1 Contextual statement 1

2 Literature review 7

3 Theoretical analysis and modifications of PSO 58

3.1 Influence of rotation matrices on the performance of PSO variants 58
3.2 Rotation invariance and local convergence 95
3.3 Analysis of the standard particle swarm 2011 136

4 Application of PSO to continuous space COPs 161

4.1 PSO for COPs 161
4.2 Disjoint feasible ragions in a COP 170
4.2.1 Locating Disjoint feasible ragions in a COP 170
4.2.2 An adptive topology for PSO to address COPs 196
4.3 PSO on the edge of feasiblity 213
4.4 Application of PSO to discrete space COPs 222

5 Conclusions 231

Appendix 232

Abstract

This is a PhD thesis by publication. It includes five journal papers, three of them already published, and two submitted for publication in a very high quality international journal in the field of Evolutionary Computation (one of which – as an invited paper). Further, the thesis contains also four conference papers presented and published in the top peer reviewed conferences, as well as one peer reviewed chapter book.

In this thesis we studied an optimization algorithm called Particle Swarm Optimization (PSO) from theoretical and application point of views. The main focus of the theoretical analysis of the algorithm was towards understanding and addressing its limitations that were related to transformations of the search space, convergence to quality solutions, and stability. Through analysis of the algorithm under transformations of the search space we proposed a modification to the original PSO so that a stable performance was guaranteed when the search space was transformed, i.e., rotated, scaled, and translated.

We also studied the ability of the original PSO in locating optimum solutions (local and global optima). Our study showed that this algorithm cannot guarantee to find a local optimum. We introduced a general formulation of topology for the original PSO and identified conditions so that it did not only guarantee local convergence but also transformation invariance. Further, we proposed a specific formulation, extracted from the general formulation, and experimentally confirmed the theoretical findings.

We investigated the most recent standard version of SPSO, called standard PSO 2011 (SPSO2011), from stability, convergence, and transformation sensitivity perspectives. Our investigations revealed essential differences between stability conditions for SPSO2011 with earlier PSO variants. We introduced stability conditions for SPSO2011 and analyzed the behavior of the algorithm before collapsing on its equilibrium. Also, we proved that this algorithm cannot guarantee to find a local optimum in the search space. We introduced sufficient condition for a general formulation that represents a large class of PSO variants so that convergence to a local optimum is guaranteed. We then modified SPSO2011 so that the mentioned sufficient condition was satisfied and convergence to local optimum was guaranteed.

Apart from theoretical analysis, we also studied performance of the algorithm when it was applied to continuous space constrained optimization problems (COPs). We introduced a new aspect in dealing with constrained optimization problems namely *locating disjoint feasible regions* in a search space. We developed a variant of PSO that was able to locate disjoint feasible regions in a search space. This track

of research can potentially be of interest of many other researchers in the field of optimization in future.

As it has been emphasized in many articles, the boundaries of feasible and infeasible regions in a COP can lead optimization algorithms to high quality solutions. We introduced a new approach to concentrate the search on the boundaries of the feasible and infeasible space. As a case study we used a variant of PSO and we confirmed that the results of the new approach are competitive with that of existing approaches in dealing with constraints.

We also proposed a coding scheme to map a discrete space constrained optimization problem, namely multi-dimensional knapsack problem, to a continuous space constrained optimization problem. Then, we investigated the ability of a variant of PSO to deal with the mapped version of this problem.

Statement of originality

I certify 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 and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

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

The author acknowledges that copyright of published works contained within this thesis 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.

Mohammadreza Bonyadi

Signature:

Date:

Acknowledgement

This thesis would not be possible without the help and support of kind people, to only some of whom is possible to give particular mention here.

I would like to express my heartfelt gratitude to Professor Zbigniew Michalewicz, my PhD advisor, whom I have enjoyed the opportunity to learn from not only how to perform research in a scientific era, but also how to think out-of-the-box. He taught me not only how to do research, but also how to find out why to do research. He helped me to find the answer of many questions I have always had in my life during the period of my PhD. I could not be prouder of my academic roots and hope that I can in turn pass on the research values and the dreams that he has given to me.

I would like to thank my PhD co-supervisors, Associate Professor Frank Neumann and Associate Professor Xiaodong Li, who were truly supportive. They have always cared about me and my future and provided great insights for me to pick my path in the best direction. I am so proud of working with you, thank you both.

I am indebted to all my friends who have supported me over the last years: Touhid, Bahram, Farshid, Ehsan, Ali, Sepide, Iman, Mohammad, and Arash, thank you very much for all memorable moments you shared with me. I also have enjoyed many useful and entertaining discussions with my friends in the science team at SolveIT Software and Optimization and Logistics group at the University of Adelaide.

I especially thank my mom and dad. My hard-working parents have sacrificed their lives for me and provided unconditional love and care. I do believe that I would not have made it this far without them. I know I always have my parents to count on when times are rough.

I would not have contemplated this road if not for my lovely wife, Maryam, who instilled love in all senses within me. She has been the best of friends along this journey. Thanks my dearest for being patient with me, for being beside me when no one could be, and for rekindling my dreams when no one believed on me.

Statements of authorships

This thesis contains one chapter book (peer reviewed), five journal papers (indexed in ERA, ISI, and/or JSR), and four conference papers (all peer reviewed and ranked as A according to ERA 2012 report). I have provided a statement of authorship for each of these articles to certify that I was actively involved in the process of preparing each article.

The following is the list of all publications included in this thesis.

Bonyadi, M. R., Li, X., & Michalewicz, Z. A hybrid particle swarm with velocity mutation for constraint optimization problems. In *Genetic and Evolutionary Computation Conference, 2013* (pp. 1-8): ACM. doi:10.1145/2463372.2463378.

Bonyadi, M. R., Li, X., & Michalewicz, Z. (2014). A hybrid particle swarm with a time-adaptive topology for constrained optimization. *Swarm and Evolutionary Computation*, 18(1), 22-37.

Bonyadi, M. R., & Michalewicz, Z. A fast particle swarm optimization algorithm for the multidimensional knapsack problem. In *Congress on Evolutionary Computation, 2012* (pp. 1-8): IEEE.

Bonyadi, M. R., & Michalewicz, Z. (2014). A locally convergent rotationally invariant particle swarm optimization algorithm. *Swarm Intelligence, 8*(3), 159-198.

Bonyadi, M. R., & Michalewicz, Z. (2014). Locating potentially disjoint feasible regions of a search space with a particle swarm optimizer. In K. Deb, & R. Datta (Eds.), *Evolutionary Constrained Optimization*: Springer-Verlag.

Bonyadi, M. R., & Michalewicz, Z. On the edge of feasibility: a case study of the particle swarm optimizer. In *Congress on Evolutionary Computation, 2014* (pp. 3059-3066): IEEE

Bonyadi, M. R., & Michalewicz, Z. (2014). Particle swarm optimization for single objective continuous space problems: a review. *Evolutionary Computation (Under review)*.

Bonyadi, M. R., & Michalewicz, Z. SPSO2011 – Analysis of stability, local convergence, and rotation sensitivity. In *Genetic and Evolutionary Computation Conference, Vancouver, Canada, 2014* (pp. 9-16): ACM.

Bonyadi, M. R., & Michalewicz, Z. (2014). SPSO2011 – Analysis of stability, local convergence, and transformation sensitivity. *Evolutionary Computation, (Invited paper)*.

Bonyadi, M. R., Michalewicz, Z., & Li, X. (2014). An analysis of the velocity updating rule of the particle swarm optimization algorithm. *Journal of Heuristics, 20*(4), 417-452.

Statement of Authorship

Title of Paper	Particle swarm optimization for single objective continuous space problems: a review
Publication Status	○ Published, ○ Accepted for Publication, ◉ Submitted for Publication, ○ Publication style
Publication Details	M. R. Bonyadi and Z. Michalewicz, submitted on 07/2014 to the journal of Evolutionary Computation, MIT press

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Mohammadreza Bonyadi		
Contribution to the Paper	Read the existing articles, categorized them, and wrote the first draft of the paper. Also, he revised the paper according to the comments from the other author.		
Signature		Date	5/10/2014

Name of Co-Author	Zbigniew Michalewicz		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Statement of Authorship

Title of Paper	An analysis of the velocity updating rule of the particle swarm optimization algorithm
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	M. R. Bonyadi, Z. Michalewicz, and X. Li, "An analysis of the velocity updating rule of the particle swarm optimization algorithm," Journal of Heuristics, vol. 20, pp. 417-452, 2014

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Mohammadreza Bonyadi		
Contribution to the Paper	Came up with the idea, tested the idea to confirm its efficiency, wrote the first draft, and applied comments from other authors.		
Signature		Date	5/10/2014

Name of Co-Author	Zbigniew Michalewicz		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author	Xiaodong Li		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Statement of Authorship

Title of Paper	A locally convergent rotationally invariant particle swarm optimization algorithm
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	M. R. Bonyadi and Z. Michalewicz, "A locally convergent rotationally invariant particle swarm optimization algorithm," Swarm Intelligence, vol. 8, pp. 159-198, 2014

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Mohammadreza Bonyadi		
Contribution to the Paper	Came up with the idea, tested the idea to confirm its efficiency, provided related mathematical proofs, wrote the first draft, and applied comments from other authors.		
Signature		Date	5/10/2014

Name of Co-Author	Zbigniew Michalewicz		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on mathematical proofs and arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Statement of Authorship

Title of Paper	SPSO2011 – Analysis of stability, local convergence, and transformation sensitivity
Publication Status	○ Published, ○ Accepted for Publication, ◉ Submitted for Publication, ○ Publication style
Publication Details	M. R. Bonyadi and Z. Michalewicz, "SPSO2011 – Analysis of stability, local convergence, and transformation sensitivity," Evolutionary Computation, MIT press The paper has been invited by the editor-in-chief of this journal

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Mohammadreza Bonyadi		
Contribution to the Paper	Came up with the idea, tested the idea to confirm its efficiency, provided related mathematical proofs, wrote the first draft, and applied comments from other authors.		
Signature		Date	5/10/2014

Name of Co-Author	Zbigniew Michalewicz		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on mathematical proofs and arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Statement of Authorship

Title of Paper	A hybrid particle swarm with velocity mutation for constraint optimization problems
Publication Status	⦿ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	M. R. Bonyadi, X. Li, and Z. Michalewicz, "A hybrid particle swarm with velocity mutation for constraint optimization problems," in GECCO, ACM, 2013, pp. 1-8. The paper was presented in the GECCO conference (ERA A)

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Mohammadreza Bonyadi		
Contribution to the Paper	Came up with the idea, tested the idea to confirm its efficiency, wrote the first draft, and applied comments from the other author. Also, he presented the paper in the conference.		
Signature		Date	5/10/2014

Name of Co-Author	Xiang Li		
Contribution to the Paper	Read the paper and provided some comments to improve the presentation of the ideas in the paper.		
Signature		Date	5/10/2014

Name of Co-Author	Zbigniew Michalewicz		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Statement of Authorship

Title of Paper	A hybrid particle swarm with a time-adaptive topology for constrained optimization
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	M. R. Bonyadi, X. Li, and Z. Michalewicz, "A hybrid particle swarm with a time-adaptive topology for constrained optimization," Swarm and Evolutionary Computation, Elsevier, 2014, DOI: 10.1016/j.swevo.2014.06.001

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Mohammadreza Bonyadi		
Contribution to the Paper	Came up with the idea, tested the idea to confirm its efficiency, wrote the first draft, and applied comments from other authors.		
Signature		Date	5/10/2014

Name of Co-Author	Xiang Li		
Contribution to the Paper	Read the paper and provided some comments to improve the presentation of the ideas in the paper.		
Signature		Date	5/10/2014

Name of Co-Author	Zbigniew Michalewicz		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Statement of Authorship

Title of Paper	Locating potentially disjoint feasible regions of a search space with a particle swarm optimizer
Publication Status	○ Published, ◉ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	M. R. Bonyadi and Z. Michalewicz, "Locating potentially disjoint feasible regions of a search space with a particle swarm optimizer," in Evolutionary Constrained Optimization, K. Deb, Ed., ed: Springer-Verlag, 2014, To appear

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Mohammadreza Bonyadi		
Contribution to the Paper	Came up with the idea, tested the idea to confirm its efficiency, wrote the first draft, and applied comments from other authors.		
Signature		Date	5/10/2014

Name of Co-Author	Zbigniew Michalewicz		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Statement of Authorship

Title of Paper	On the edge of feasibility: a case study of the particle swarm optimizer
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	M. R. Bonyadi and Z. Michalewicz, "On the edge of feasibility: a case study of the particle swarm optimizer," in Congress on Evolutionary Computation, IEEE, Beijing, China, 2014, pp. 3059-3066 The paper was presented in the CEC conference (ERA A).

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Mohammadreza Bonyadi		
Contribution to the Paper	Read the existing articles, categorized them, and wrote the first draft of the paper. Also, he revised the paper according to the comments from the other author.		
Signature		Date	5/10/2014

Name of Co-Author	Zbigniew Michalewicz		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Statement of Authorship

Title of Paper	A fast particle swarm optimization algorithm for the multidimensional knapsack problem
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	M. R. Bonyadi and Z. Michalewicz, "A fast particle swarm optimization algorithm for the multidimensional knapsack problem," in Congress on Evolutionary Computation, IEEE, 2012, pp. 1-8 The paper was presented in the CEC conference (ERA A).

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Mohammadreza Bonyadi		
Contribution to the Paper	Came up with the idea, tested the idea to confirm its efficiency, wrote the first draft, and applied comments from other authors. Also, he presented the paper in the conference.		
Signature		Date	5/10/2014

Name of Co-Author	Zbigniew Michalewicz		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Statement of Authorship

Title of Paper	SPSO2011 – Analysis of stability, local convergence, and rotation sensitivity
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	M. R. Bonyadi and Z. Michalewicz, "SPSO2011 – Analysis of stability, local convergence, and rotation sensitivity," in Genetic and Evolutionary Computation Conference, ACM, Vancouver, Canada, 2014, pp. 9-16 This paper received the best paper award in the GECCO conference (ERA A)

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Mohammadreza Bonyadi		
Contribution to the Paper	Came up with the idea, tested the idea to confirm its efficiency, provided related mathematical proofs, wrote the first draft, and applied comments from other authors.		
Signature		Date	5/10/2014

Name of Co-Author	Zbigniew Michalewicz		
Contribution to the Paper	Read the paper and provided comments, including editorial comments and conceptual feedback on mathematical proofs and arguments, to improve the paper.		
Signature		Date	5/10/2014

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

1 Contextual statement

Particle Swarm Optimization (PSO) is a stochastic population-based optimization algorithm that was proposed by Kennedy and Eberhart [1] in 1995. It has been successfully applied to many problems such as artificial neural network training, function optimization, fuzzy system control, and pattern classification [2, 3], to name but a few. After almost 20 years since inception of PSO there are still many interesting questions for researchers related to this algorithm such as: How to ensure the stability of particles? How particles behave before convergence? What are the impacts of the coefficients/topology/population size on the behavior of particles? What are the best values of coefficients to deal with different optimization problems with different landscapes? What is the best strategy to make the algorithm deal with constraints? During my Ph.D. candidature I explored some of these questions and published a few papers on these topics.

I conducted a comprehensive survey [4] to find out what are the existing advances in PSO variants and what are open questions. The review contained three main parts: limitations of PSO variants, modifications of PSO, and applications of PSO variants to constrained optimization problems (COPs). The first part of the paper, limitations of PSO, contained two sections: limitations related to the convergence and limitations related to transformations. Limitations related to the convergence referred to theoretical investigations of PSO variants to answer the following questions: do particles converge (*aka* stability), and, if yes, where do they converge to (*aka* local convergence), and, if they converge to a local optimum for sure, how long does it take to do so with a given precision (aka expected first hitting time). Limitations related to transformations referred to the theoretical analysis that targeted performance of PSO variants under transformation of the search space (rotation, scale, and translation). The second part of the review paper, modifications of PSO, referred to studies in which PSO components, including parameters and update rules, were modified. Applications of the algorithm to COPs were studied in the last part of the paper. Through this survey paper, we revealed the lack of in depth theoretical analysis of different variants from different perspectives such as topology, coefficients, transformations sensitivity, and expected first hitting time. We also summarized few potential areas of improvements in PSO parameters (topology, population size, etc.), and the need of theoretical analysis of the algorithm's performance to deal with COPs.

As it was found in our survey paper, theoretical analysis of PSO variants under transformation of the search space has not been explored in depth. In an article related to this topic it was proven [5] that PSO in its original formulation is sensitive to the rotation of the search space. It was also proven that if the random matrices in the formulation of the original PSO are replaced by random rotation matrices or random scalars then this sensitivity is resolved. The usage of scalars was shown to

have another negative effect that was: particles stop moving at the very early stages of the search process. However, usage of random rotation matrices in original PSO can increase the performance of the algorithm in some cases. Due to the time complexity of generating random rotation matrices, an approximation method was used to generate them [5]. We further investigated [6] Wilke's findings and proposed a method to generate accurate random rotation matrices with the same computational cost. We showed that the usage of random rotation matrices is useful not only in original PSO but also in many other PSO variants. Also, we showed that although random rotation in all planes leads the algorithm to a good performance, a random rotation in one randomly selected plane is also as effective while the latter involves significant lower computational cost. In addition to exploring the findings in [5], we also revealed some issues in many PSO variants that are addressed by using the rotation matrices. As an example, in some situations, particles oscillate along with a dimension forever while searching other values of other dimensions becomes impossible (this issue was called *dimensional stagnation*).

In 2002 it was proven [7] that PSO in its original form is not a local optimizer, i.e., it does not guarantee to locate a local optimum in a search space. One example is that, after a long run, particles are *stagnated* and no more movement is possible while there is no guarantee that this point is really a high quality one. In addition, the performance of PSO impairs radically [8] when it is applied to large scale optimization problems even for smooth convex functions (sphere function for example). It was also shown [9] that the original PSO is not effective with small population size (2 particles for example). We investigated [8] six aforementioned issues (stagnation, dimensional stagnation, local convergence, population size, rotation sensitivity, and problem scale) in the original PSO from theoretical perspective. We argued that there is a relation between these issues in the following way: if we address local convergence, rotation sensitivity, and stagnation then all these six issues are effectively addressed. For a general form of topology for original PSO we introduced sufficient conditions together with mathematical proofs to guarantee local convergence and rotation invariance at the same time. We also proved that these sufficient conditions guarantee to address stagnation as well; hence, all six issues were addressed at the same time. To the best of our knowledge, this was the first study that addressed all of these issues at the same time.

A PSO variant called Standard PSO 2011 (SPSO2011) was proposed in 2011 [10] that was claimed to be rotation invariant. Although the formulation for this variant shared some similarities to that of original PSO, some differences were also apparent. Because of these differences, there were many aspects of the algorithm that needed to be investigated. We investigated [11] three important aspects of SPSO2011: stability, local convergence, and rotation sensitivity. For the stability, we were after boundaries for parameters of the algorithm so that particles are convergent, i.e., become stable in an equilibrium point. We introduced boundaries for the coefficients so that particles converge to their equilibrium. After making sure

particles are convergent, we needed to make sure that they do not just converge to a random solution somewhere in the search space but to quality solutions such as local optima. We proved that SPSO2011 cannot guarantee convergence to a local optimum even in a smooth convex search space. We also proved that the claim about rotation sensitivity of the algorithm is true. My article received the best paper award in one of the top conferences (Genetic and Evolutionary Computation Conference, GECCO) in the field of Evolutionary Computation in the track Ant Colony and Swarm Intelligence (ACSI).

We were invited to extend the above paper to be published by Evolutionary Computation journal, MIT press. The extended version [12] introduced two additional components to the original paper: analysis of the behavior of particles before convergence and addressing the local convergence issue. The former component, analysis of the behavior of particles before convergence, refers to identification of coefficients boundaries associated with different behaviors of particles positions before convergence. We categorized behaviors of particles into four different groups: non-oscillatory, smooth harmonic, zigzagging, and combination of harmonic and zigzagging. We showed that each of these behaviors correspond to a range of frequencies that particles positions oscillate with. We identified coefficients boundaries for the ranges of frequencies associated with each behavior. These boundaries help researchers to pick appropriate values for coefficients to deal with optimization problems more effectively. We also provided a sufficient condition (together with mathematical proofs) to guarantee local convergence for a formulation that represents a large class of PSO variants. Then we modified SPSO2011 so that this sufficient condition is guaranteed, and hence, the modified SPSO2011 is locally convergent.

Apart from theoretical analysis of PSO variants and addressing issues in the algorithm to improve its performance and scalability, we also used the algorithm to deal with a class of problems called Constrained Optimization Problems, COP. Note that the theoretical analysis conducted in the papers mentioned earlier were related to general continuous space optimization so that they can be also used to deal with COPs. We proposed [13] a PSO variant and a hybrid of PSO and CMA-ES [14] to deal with COPs. We showed that the proposed PSO variant is more effective than CMA-ES in finding feasible regions while CMA-ES performs better than that PSO variant in improving the results at the end. The PSO variant used a restart strategy that was experimentally shown to offer improvement in the final results. The reason for this improvement was explained as it enables the algorithm to find potentially disjoint feasible regions. Thus, as the quality of solutions in each of these regions might be different, there is a higher chance to locate better feasible regions (regions that contain higher quality solutions) by using this strategy.

We further extended [15] the idea of searching the search space for different feasible regions. In fact, we argued in [15] that it is very important to locate disjoint feasible regions because it improves the chance of finding the region that contains

the optimal solution, the *optimal region*. This was the first time that locating disjoint feasible regions was introduced in the evolutionary computation community. We mapped the problem of locating many feasible regions to another well-known problem in the field of optimization called *niching*. In niching it is required to locate as many local optima as possible. It was shown [16] that PSO has a good ability to locate niches. Some topologies of PSO were investigated and it was found that a non-overlapping topology is effective in niching as long as the number of dimensions is small. When the number of dimensions grows, however, this topology becomes the worst. We investigated this phenomenon and explained the reasons for this impairment in [15]. We showed that because of the population size issue [8], sub-swarms in non-overlapping topology cannot perform well when the number of dimensions grows. In fact, as each sub-swarm performs independently, the population size issue becomes apparent in the sub-swarms and the performance of the algorithm is impaired as the number of dimensions grows. Hence, because a strategy that addresses stagnation can address the population size issue as well [8], we proposed to use a random perturbation to prevent particles from stop moving with the hope of addressing the issue of performance drop in the non-overlapping topology. Our experiments showed significant improvement in niching ability of the non-overlapping topology when this strategy was used. We used this new PSO variant to locate disjoint feasible regions [15].

The idea of locating disjoint feasible regions by niching techniques was further extended in [17] where we proposed to use a time adaptive topology. The time adaptive topology started with non-overlapping small sub-swarms to locate as many feasible regions as possible and gradually changed the topology to concentrate on the best found region to improve the results. In fact, during iterations, more resources (particles) are assigned to the sub-swarms with better potentials so that better regions are explored by more resources. Although the idea was simple, the results were significantly better than other existing methodologies.

In many COPs in the real-world the boundary between feasible and infeasible space contains high quality solutions [18, 19]. The reason is that constraints formulate capacity of resources in real-world optimization problems. Because usually at high quality solutions some resources are used to their maximum capacity (they are referred to as bottlenecks [19]), the corresponding constraints are active (their values are close to their maximum allowed values) at those high quality solutions. The consequence of this phenomenon is that high quality solutions are expected to be found close to the boundaries of feasible regions. Thus, it is useful if we extend optimization algorithms (e.g. PSO) so that they can search the edges of feasible regions more effectively. In [20] we provided proper definitions for the boundary of feasible and infeasible regions. Then we introduced some equations that combine all constraints in a given COP and generate a new COP. It was proven that the feasible region of the new generated COP corresponds with the boundary of feasible regions of the original COP. The proposed formulations that can model

different problem specific knowledge, such as known active constraints or subset of active constraints, to make the search more restricted towards priory-known quality regions.

PSO was originally designed to deal with continuous space optimization problems. It is possible, however, to encode discrete optimization problems (e.g. multidimensional knapsack problem) to continuous space variable. We proposed [21] a coding scheme by which the multidimensional knapsack problem was coded as a continuous COP and PSO was used to solve the problem. Results of this study showed that the algorithm can find high quality solutions faster than many existing methods.

The rest of this thesis was organized into four sections. Section 2 contains a survey paper [4] that reviews earlier articles related to PSO. Section 3 provides some theoretical analysis related to stability, convergence to local optimum, and transformation sensitivity of the algorithm in its original form. This section contains three articles [6, 8, 12]. The article [12] is an extended version of [11] so that only the latter has been included in the appendix. In section 4, five articles [13, 15, 17, 20, 21] were placed in which the applications of PSO variants to constrained optimization problems are discussed. We conclude the thesis in section 5.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *International Conference on Neural Networks*, 1995, pp. 1942-1948.

[2] A. Engelbrecht, *Fundamentals of computational swarm intelligence*. Hoboken, NJ: Wiley, 2005.

[3] R. Poli, "Analysis of the publications on the applications of particle swarm optimisation," *Journal of Artifficial Evolution and Application,* vol. 2008, pp. 1-10, 2008.

[4] M. R. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective continuous space problems: a review," *Evolutionary Computation (Under review),* 2014.

[5] D. N. Wilke, S. Kok, and A. A. Groenwold, "Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on scale and frame invariance," *International journal for numerical methods in engineering,* vol. 70, pp. 985-1008, 2007.

[6] M. R. Bonyadi, Z. Michalewicz, and X. Li, "An analysis of the velocity updating rule of the particle swarm optimization algorithm," *Journal of Heuristics,* vol. 20, pp. 417-452, 2014.

[7] F. Van den Bergh and A. Engelbrecht, "A new locally convergent particle swarm optimiser," in *Systems, man and cybernetics*, Hammamet, Tunisia, 2002, pp. 96-101.

[8] M. R. Bonyadi and Z. Michalewicz, "A locally convergent rotationally invariant particle swarm optimization algorithm," *Swarm Intelligence,* vol. 8, pp. 159-198, 2014.

[9] F. Van den Bergh and A. P. Engelbrecht, "A convergence proof for the particle swarm optimiser," *Fundamenta Informaticae,* vol. 105, pp. 341-374, 2010.

[10] M. Clerc. (2011). *Standard Particle Swarm Optimisation* Available: http://hal.archives-ouvertes.fr/hal-00764996

[11] M. R. Bonyadi and Z. Michalewicz, "SPSO2011 – Analysis of stability, local convergence, and rotation sensitivity," in *Genetic and Evolutionary Computation Conference*, Vancouver, Canada, 2014, pp. 9-16.
[12] M. R. Bonyadi and Z. Michalewicz, "SPSO2011 – Analysis of stability, local convergence, and transformation sensitivity," *Evolutionary Computation, (Invited paper, Under review),* 2014.
[13] M. R. Bonyadi, X. Li, and Z. Michalewicz, "A hybrid particle swarm with velocity mutation for constraint optimization problems," in *Genetic and Evolutionary Computation Conference*, Amsterdam, The Netherlands, 2013, pp. 1-8.
[14] N. Hansen, "The CMA evolution strategy: a comparing review," *Towards a new evolutionary computation,* pp. 75-102, 2006.
[15] M. R. Bonyadi and Z. Michalewicz, "Locating potentially disjoint feasible regions of a search space with a particle swarm optimizer," in *Evolutionary Constrained Optimization*, K. Deb and R. Datta, Eds., ed: Springer-Verlag, 2014.
[16] X. D. Li, "Niching Without Niching Parameters: Particle Swarm Optimization Using a Ring Topology," *IEEE Transactions on Evolutionary Computation,* vol. 14, pp. 150-169, Aug 2010.
[17] M. R. Bonyadi, X. Li, and Z. Michalewicz, "A hybrid particle swarm with a time-adaptive topology for constrained optimization," *Swarm and Evolutionary Computation* vol. 18, pp. 22-37, 2014.
[18] M. Schoenauer and Z. Michalewicz, "Evolutionary computation at the edge of feasibility," *Parallel Problem Solving from Nature—PPSN IV,* pp. 245-254, 1996.
[19] M. R. Bonyadi, Z. Michalewicz, and M. Wagner, "Beyond the edge of feasibility: analysis of bottlenecks," in *International Conference on Simulated Evolution And Learning (SEAL)*, Newzealand, 2014.
[20] M. R. Bonyadi and Z. Michalewicz, "On the edge of feasibility: a case study of the particle swarm optimizer," in *Congress on Evolutionary Computation*, Beijing, China, 2014, pp. 3059-3066.
[21] M. R. Bonyadi and Z. Michalewicz, "A fast particle swarm optimization algorithm for the multidimensional knapsack problem," in *Congress on Evolutionary Computation*, 2012, pp. 1-8.

2 Literature review

The article included in this chapter is a survey paper that reviews existing approaches related to the particle swarm optimization.

M. R. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective continuous space problems: a review," *Evolutionary Computation*, Under review

Particle swarm optimization for single objective continuous space problems: a review

Mohammad Reza Bonyadi mrbonyadi@cs.adelaide.edu.au, rezabny@gmail.com
Department of Computer Science, The University of Adelaide, Adelaide, 5005, Australia

Zbigniew Michalewicz zbyszek@cs.adelaide.edu.au
Department of Computer Science, The University of Adelaide, Adelaide, 5005, Australia. He is also with Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland, Polish-Japanese Institute of Information Technology, Warsaw, Poland, and Chief of Science, Complexica, Adelaide, Australia

Abstract

This paper reviews recent studies on Particle Swarm Optimization (PSO) algorithm. The review has concentrated on high impact recent articles which have analyzed and/or modified PSO algorithm. Some potential areas for future studies related to PSO are presented in this paper.

Keywords

Particle swarm optimization, stability analysis, local convergence, rotation invariance

1 Motivation

Particle swarm optimization (PSO) is a stochastic population-based optimization method that was proposed by Kennedy and Eberhart (1995). It has been successfully applied to many problems such as artificial neural network training, function optimization, fuzzy control, and pattern classification (Engelbrecht, 2005; Poli, 2008), to name a few. Because of the ease of implementation and fast convergence to acceptable solutions, PSO has received attention in recent years (Poli, 2008). Since 1995, different aspects of the original version of PSO have been modified and many variants have been proposed. Although a few review articles on PSO (see section 2 for details) have been published already (Banks et al., 2007, 2008; Hu et al., 2004; Parsopoulos and Vrahatis, 2002b; Poli, 2008; Poli et al., 2007b; Song and Gu, 2004), there are two important reasons for an additional review paper:

1. The latest comprehensive review paper on PSO was published in 2008; however, many new articles on PSO have been published since then. To estimate the growth of the number of publications in this field, we conducted a search for the exact match of "particle swarm optimization" in the title of documents in five most recognized scientific databases (Scopus, Google scholar, Springer, Web of science, and IEEE Xplore, see Fig. 1). The ratio of the number of articles published since 2008 to the ones published before 2008 in Scopus, Google scholar, Springer, Web of science, and IEEE Xplore data bases was 3.71, 2.68, 2.71, 2.42, and 3.12, respectively. In total, almost 75% of articles on PSO have been published since 2008. As the

©200X by the Massachusetts Institute of Technology

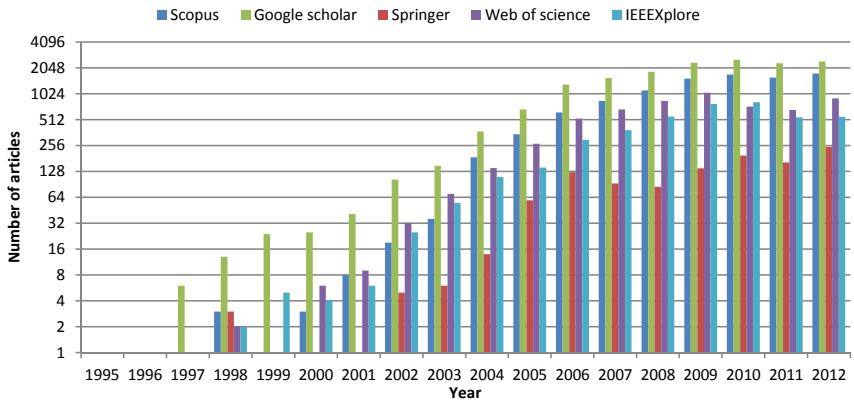


Figure 1: The number of published articles with particle swarm optimization in their title in Scopus, Google scholar, Springer, Web of science, and IEEE Xplore databases.

latest comprehensive review paper on PSO was published in 2008, the need for a new review paper seems justified.

2. Several limitations were identified in the standard PSO during past years that have not been analyzed in earlier review papers (see section 2 for more details). These limitations are related to sensitivity to transformations (rotation, translation, and scale), local convergence, stability, first hitting time, and biases (Auger et al., 2009; Clerc and Kennedy, 2002; Hansen et al., 2008; Lehre and Witt, 2013; Spears et al., 2010; Van den Bergh and Engelbrecht, 2010; Wilke et al., 2007b). Addressing these limitations is very important because, as long as these limitations exist, the good performance of PSO is not scalable to a wide variety of optimization problems (Hansen et al., 2008, 2011). Therefore, there is a need to review advances in addressing these limitations in detail.

Because of a large number of published articles related to PSO we designed two criteria to select articles for review purposes as follows:

1. Criterion $C1(yy, imp, hind, c)$ is met for an article if:
   (a) it has been published in the year *yy* or later, and
   (b) it is a journal article and the impact factor[1] of the journal (reported by Thomson reuters) is larger than *imp*, or the article has been cited more than *c* times, or
   (c) it is a conference article and the h-5[2] index of the conference (reported by the Google scholar metric measure[3]) is higher than *hind*, or the article has been cited by more than *c* times

[1]See http://wokinfo.com/essays/impact-factor/ for the definition of the impact factor. This measure is reported annually for the journals which have been indexed by the Thomson reuters web of science.

[2]h-5 index is the largest number (h) such that h articles published in the previous 5 years have at least h citations each.

[3]Available at `http://scholar.google.com.au/citations?view_op=top_venues&hl=en&vq=eng_evolutionarycomputation`.

2. Criterion $C2(yy, r)$ is met for an article if:
   (a) it has been published before the year *yy*, and
   (b) it has been cited at least *r* times.

The main source for the number of citations for the articles was the Google scholar. We used the software Harzings Publish or Perish[4] extended by some manual procedures to find the articles and check whether the criteria are met. The Criterion C1 is used to select high quality articles[5] that have been published recently (it concentrates on articles published in the year *yy* or later). Note that, because a recently published article might not have a high number of citations we have designed this criterion in such a way that highly cited articles or the articles that have been published in recognized journals/conferences are included in our review. Criterion C2 ensures to select older articles that have played a significant rule to develop the algorithm to the level that it is seen today. Note that PSO variants that belong to the family of Bare-Bones PSO (Blackwell, 2012; Kennedy, 2003) are not reviewed in this paper as this topic needs a separate review paper on its own.

The articles included in this survey are categorized into three groups:

Group 1 Articles which have analyzed PSO variants from a theoretical point of view (including proper proofs in the analyses) and have identified a limitation in a PSO variant (standard and other variants). No criteria was used in this group to filter the articles,

Group 2 Articles which have proposed new variants to improve the performance of the algorithm to solve "single objective static Unconstrained Optimization Problems", UOPs[6]. The articles for review in this group are filtered by $C1(2008, 2, 30, 30)$ or $C2(2008, 300)$,

Group 3 Articles which have modified PSO to deal with "single objective static Constrained Optimization Problems", COPs (see (Michalewicz, 1995; Michalewicz and Schoenauer, 1996) for definition of COPs). The articles for review in this group are filtered by $C1(2008, 2, 20, 20)$ or $C2(2008, 150)$.

Each of the aforementioned group is divided into several sub-groups (topics) and each of these sub-groups is discussed in one sub-section. At the end of each sub-section, a summary of the discussed topic together with challenges and potential future directions related to that topic are given. The reason behind changing the parameters for each criterion in different groups is that articles in different groups have different potential to be cited. As an example, it is probable that a paper which has analyzed coefficients of the algorithm to deal with UOPs (categorized in the Group 1) is cited in a paper that is related to COPs. Thus, higher number of citations is expected for the articles in the Group 2 in order to be included in this survey. Also, articles which have investigated/applied PSO for/to a specific application, such as solving the knapsack problem (Bonyadi and Michalewicz, 2012) or the traveling salesman problem (Chen et al., 2010), are not reviewed in this paper.

[4] Available online at `http://www.harzing.com/pop.htm`, version 4.4.5 was used.

[5] The designed criteria are of course not optimal as it is possible to find low quality papers at a high impact journal. However, due to the large number of articles in the area of PSO, we needed to design a preliminary filter to ensure the review spends more effort on most-likely higher quality articles.

[6] Although large scale optimization and niching are some types of UOPs, we will not review the papers in these topics because of the extensive number of articles, which is not fit in one single review paper.

The main aim of this survey is to review the presented ideas, categorize and link most recent high quality (based on the criteria introduced before) studies, and provide a vision for what might be valuable for future research. Some conclusions derived by the authors of the included articles are difficult to evaluate as significant additional effort would be required to fully investigate their different aspects. However, we briefly discuss some of these methods from two perspectives: theoretical (convergence to local optima, transformation invariance, and the time complexity of the methods) and experimental (validity of statistical tests (Derrac et al., 2011), number of test cases, and potential biases in test cases). Such analyses, although high level, may assists the reader to determine to what extent the findings in those articles are reliable and how they can be tested to be confirmed or disproved.

The rest of this review paper is organized as follows. Section 2 provides some information about the original form of PSO as well as earlier reviews on the PSO topic. Section 3 reviews articles which have identified limitations in PSO. Section 4 reviews articles which have modified PSO to have a better performance in solving UOPs, whereas section 5 reviews articles which have extended PSO to deal with COPs. Section concludes this paper.

2 Original PSO and earlier review papers

In this paper, a UOP (minimization) is defined as:

$$\text{find } x \in S \subseteq R^d \text{ such that } \forall y \in S, f(x) \leq f(y) \tag{1}$$

where S is the search space defined by $S = \{\vec{y} : l_i \leq y_i \leq u_i\}$, y_i is the ith dimension of the vector $\vec{y}$ and u_i and l_i are upper bound and lower bound of the ith dimension, respectively, d is the number of dimensions, and $f(.)$ is the objective function. The aim is to find x such that $f(x)$ is the minimum value in the search space.

PSO (Kennedy and Eberhart, 1995) is based on a population (referred to as swarm) of $n > 1$ particle[7]; each particle is defined by three d-dimensional vectors:

- Position ($\vec{x}_t^i$) — is the position of the ith particle in the tth iteration. This is used to evaluate the particles quality,
- Velocity ($\vec{V}_t^i$) — is the direction and length of movement of the ith particle in the tth iteration,
- Personal best ($\vec{p}_t^i$) — is the best position[8] that the ith particle has visited in its lifetime (up to the tth iteration). This vector serves as a memory to store the knowledge of quality solutions (Eberhart and Kennedy, 1995).

All of these vectors[9] are updated at every iteration t for each particle i:

$$\vec{V}_{t+1}^i = \mu\left(\vec{x}_t^i, \vec{V}_t^i, N_t^i\right) \; for\ all\ i \tag{2}$$

$$\vec{x}_{t+1}^i = \xi\left(\vec{x}_t^i, \vec{V}_{t+1}^i\right) \; for\ all\ i \tag{3}$$

[7] In theoretical studies (see section 4), the number of particles is sometimes set to 1 to simplify the analyses.

[8] Personal best can be a set of best positions, but all PSO types listed in this paper use single personal best.

[9] One should note that these vectors are in fact *position vectors*, a vector that connects a point to the center of the coordinates.

$$\vec{p}_{t+1}^{\,i} = \begin{cases} \vec{x}_{t+1}^{\,i} & f\left(\vec{x}_{t+1}^{\,i}\right) < f\left(\vec{p}_t^{\,i}\right) \ and \ \vec{x}_{t+1}^{\,i} \in S \\ \vec{p}_t^{\,i} & otherwise \end{cases} \quad for\ all\ i \tag{4}$$

In Eq. 2, N_t^i, known as the neighbor set of particle i, is a subset of personal best positions of the particles that contribute to the velocity update rule of particle i at iteration t, i.e., $N_t^i = \left\{\vec{p}_t^{\,k} | k \in \left\{T_t^i \subseteq \{1, 2, \ldots, n\}\right\}\right\}$ where T_t^i is a set of indices of particles which contribute to the velocity update rule of particle i at iteration t. Clearly, the strategy to determine T_t^i might be different for various types of PSO algorithms and it is usually referred to as the topology of the swarm. Many different topologies have been defined for PSO so far (Mendes et al., 2004), e.g., global best topology, ring topology, wheels topology, pyramid typology; each of them has some advantages and disadvantages (Clerc, 2006; Mendes et al., 2004). Topology in fact determines which other particles should a particle learn[10] from (connect to). The function $\mu\left(.\right)$ calculates the new velocity vector for particle i according to its current position, current velocity $\vec{V}_t^i$, and neighbor set N_t^i. In Eq. 3, $\xi\left(.\right)$ is a function which calculates the new position of particle i according to its previous position and its new velocity. Usually $\xi\left(\vec{x}_t^i,\ \vec{V}_{t+1}^i\right) = \vec{x}_t^i + \vec{V}_{t+1}^i$ is accepted for updating the position of particle i. In Eq. 4, the new personal best position for particle i is updated according to the objective value of its previous personal best position and the current position. In PSO, three updating rules (Eq. 2, 3, and 4) are applied to all particles iteratively until a predefined criterion is met (e.g., the maximum number of iterations is achieved or a sufficiently good objective value is found). Also, $\vec{x}_0^i$ and $\vec{V}_0^i$ are generated either randomly or by using a heuristic method and $\vec{p}_0^{\,i}$ is initialized to $\vec{x}_0^i$ for all particles.

In the first version of PSO (Kennedy and Eberhart, 1995), called "Original Particle Swarm Optimization", OPSO, the set N_t^i contained only two vectors that were the personal best position of the ith particle ($\vec{p}_t^{\,i}$) and the best position found over the whole swarm (known as $\vec{g}_t$), i.e., $T_t^i = \{i, \tau_t\}$ where $\tau_t = \underset{l=\{1,\ldots,n\}}{argmin}\left(F(\vec{p}_t^{\,l})\right)$ (the particle τ_t is referred to as the global best particle throughout the paper). This topology is called global best topology for PSO. Also, the function $\mu\left(.\right)$ in Eq. 3 was defined (Kennedy and Eberhart, 1995) as:

$$\vec{V}_{t+1}^i = \vec{V}_t^i + \varphi_1 R_{1t}^i\left(\vec{p}_t^{\,i} - \vec{x}_t^i\right) + \varphi_2 R_{2t}^i\left(\vec{g}_t - \vec{x}_t^i\right) \tag{5}$$

In this equation, φ_1 and φ_2 are two real numbers called cognitive and social weights, also known as *acceleration* coefficients[11], and $\vec{p}_t^{\,i}$ and $\vec{g}_t$ are the personal best (of particle i) and the global best vectors, respectively, at iteration t. Also, the role of vectors $CI = \vec{p}_t^{\,i} - \vec{x}_t^i$ (Cognitive Influence) and $SI = \vec{g}_t - \vec{x}_t^i$ (Social Influence) is to attract the particles to move towards known quality solutions, i.e., personal and global best. Moreover, R_{1t}^i and R_{2t}^i are two $d \times d$ diagonal matrices[12] (Clerc, 2006; Montes de Oca et al., 2009), where their elements are random numbers distributed uniformly ($U\left(0,\ 1\right)$) in [0, 1]. Note that matrices R_{1t}^i and R_{2t}^i are generated at each iteration for each particle separately. Although there is no general rule for initializing velocity, it was experimentally shown that it is usually better to initialize velocity to zero (Engelbrecht, 2012).

[10] Learning from a particle refers to gathering information from that particle (either personal best of that particle or its current position).

[11] These two coefficients control the effect of personal and global best vectors on the movement of particles and they play an important role in the convergence of the algorithm. They are usually determined by a practitioner or by analyzing the dynamic of particles movement.

[12] Alternatively, these two random matrices are often considered as two random vectors. In this case, the multiplication of these random vectors by CI and SI is element-wise

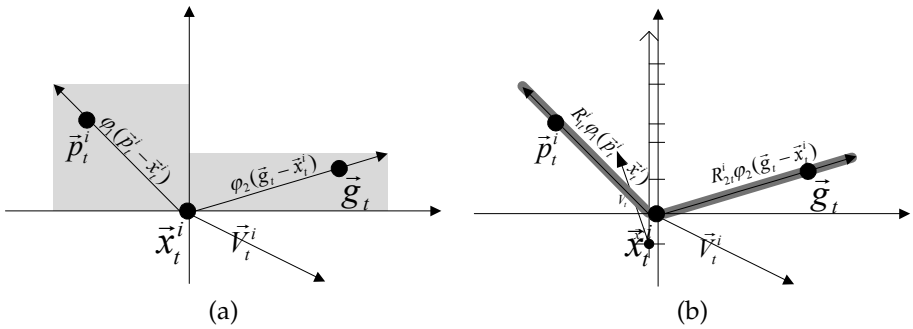


Figure 2: The grey areas are the areas in which $\varphi_1 R_{1t}^i \left(\vec{p}_t^i - \vec{x}_t^i\right)$ and $\varphi_2 R_{2t}^i \left(\vec{g}_t - \vec{x}_t^i\right)$ may be found, (a) SPSO, (b) LPSO.

In 1998, Shi and Eberhart (1998a) introduced a new coefficient ω, called inertia weight, to control the influence of the previous velocity value on the updated velocity. Indeed, Eq. 5 was rewritten as:

$$\vec{V}_{t+1}^i = \omega \vec{V}_t^i + \varphi_1 R_{1t}^i \left(\vec{p}_t^i - \vec{x}_t^i\right) + \varphi_2 R_{2t}^i \left(\vec{g}_t - \vec{x}_t^i\right) \quad (6)$$

The coefficients ω, φ_1, and φ_2 control influences of the previous velocity, cognitive influence, and social influence on the movement. This variant is called "Standard Particle Swarm Optimization", SPSO, throughout the paper. As in OPSO, R_{1t}^i and R_{2t}^i are random diagonal matrices generated for each particle at each iteration. If the random numbers on the diagonal of matrices R_{1t}^i and R_{2t}^i are set to equal values then these matrices only scale the vectors $\left(\vec{p}_t^i - \vec{x}_t^i\right)$ and $\left(\vec{g}_t - \vec{x}_t^i\right)$, i.e., no rotation takes place by these matrices. This setup (considering similar random numbers on the diagonal of the random matrices) has been introduced as a common error in the implementation of PSO in (Clerc, 2006), chapter 3, section 2. However, some studies have considered this setup as another variant of PSO that was called "Linear Particle Swarm Optimization", LPSO (Bonyadi et al., 2013) that is reviewed later in this paper. The grey rectangles in Fig. 2 represent the areas in which $\varphi_1 R_{1t}^i \left(\vec{p}_t^i - \vec{x}_t^i\right)$ and $\varphi_2 R_{2t}^i \left(\vec{g}_t - \vec{x}_t^i\right)$ may be found. Clearly, by applying R_{1t}^i and R_{2t}^i to CI and SI, the resulting vectors may be different from the original ones.

A few review articles on PSO have been published so far (Banks et al., 2007, 2008; Hu et al., 2004; Parsopoulos and Vrahatis, 2002b; Poli, 2008; Poli et al., 2007b; Song and Gu, 2004). In 2002, seven years after the original PSO was introduced, the first review paper was published (Parsopoulos and Vrahatis, 2002b). The performance of several PSO variants in locating global optimum, jumping out of local optima, dealing with noise, solving multi-objective optimization problems, etc. were reviewed. In addition to the review part of that article, a PSO variant was proposed that used a function stretching approach (Vrahatis et al., 1996) to jump out of local optima.

In 2004 another review article (Hu et al., 2004) on PSO was published that consisted of four main parts: basic PSO algorithm, discrete PSO, multi-objective, and applications. The first part contained different strategies for determining parameters of the velocity update rule (inertia, cognitive, and social weights). In the second part, several PSO variants for solving combinatorial optimization problems were overviewed. In the third part, some existing studies related to multi-objective PSO variants were outlined. Some applications of the algorithm in different disciplines (e.g., feature se-

lection (Agrafiotis and Cedeno, 2002), neural networks (Eberhart and Shi, 1998)) were discussed in the last part.

Another review study was published (Song and Gu, 2004) also in 2004. The paper had two main parts: performance improvements and applications. In the first part, several papers which aimed at modifying parameters, increasing diversity, or faster convergence were reviewed. In the second part, applications of PSO to multi-objective optimization, electronics, and training neural networks were discussed.

In 2007, the fourth review paper on PSO was published (Banks et al., 2007). The paper was organized into several sections: a section on the history of the standard PSO was followed by a discussion on discrete and continuous space PSO variants. The authors then outlined the PSO variants which addressed premature convergence, dealt with dynamic environments, analyzed the swarm behavior with different parameters values, and investigated parallelization of the algorithm. The second part of (Banks et al., 2007) was published as a separate article (Banks et al., 2008) and it reviewed hybridization in PSO, combinatorial PSO, constrained optimization using PSO, and applications of PSO. In the same year (2007), another survey paper was published (Poli et al., 2007b). The paper was organized into six parts: population dynamics focused on original version of PSO, parameters, and analysis of the dynamics of the particles; topology which was divided into static and dynamic topologies; specialization including discrete space PSO; dynamic environment, etc; theoretical analysis that contained deterministic and stochastic models of PSO; applications of PSO; and open questions.

In 2008, the last survey article was published (Poli, 2008) that discussed a number of publications of PSO. The aim of that paper was not to review advances in PSO but to show the growth of the publications in the field of PSO. Results from investigating 700 articles from IEEE Xplore database showed an expressive growth of publications in the field of PSO at that time. This growth raised a question "What makes PSO so attractive to practitioners?" It was argued that one of the main reasons behind popularity of PSO was simplicity of the algorithm that makes it easier to adapt to different applications. Also, another reason is that the algorithm delivers reasonable results for many different applications. In fact, the main request of practitioners from an optimizer is to be simple, reliable, and at the same time deliver reasonable (and not necessarily the best) results. All these characteristics are found in PSO, that makes the algorithm attractive.

There have been some other review papers that are on specific application of PSO, e.g., applications of PSO in data clustering, applications of PSO in dispatching problem (Mahor et al., 2009), applications of PSO in solar photovoltaic systems to name a few. However, they have not been included in this paper because their focus was on a specific area only.

3 Identified limitations in PSO

Several limitations in SPSO have been identified so far. The term "limitation" refers to an issue that has been proven to prevent the algorithm from performing well from different aspects such as locating high quality solutions or being stable. There are two main areas related to the limitations in PSO that are *convergence* and *transformation invariance* (Wilke, 2005) (see also (Bonyadi and Michalewicz, 2014a) about the details of these limitations in SPSO). In the following two sub-sections, articles that have analyzed limitations related to these two topics in PSO are discussed.

3.1 Limitations related to convergence

We classify limitations related to convergence in PSO into three groups: *convergence to a point*, *convergence to a local optimum*, and *expected first hitting time* (EFHT), that are defined in the next paragraphs.

One of the earliest convergence analyses of stochastic optimization algorithms took place by Matyas (1965) that was followed by (Baba, 1981; Solis and Wets, 1981) almost two decades later. An iterative stochastic optimization algorithm (optimization algorithm in short) is said to converge to a point X in the search space in probability (to converge in short) if

$$\forall \varepsilon > 0, \lim_{t \to \infty} P\left(|x_t - \mathrm{X}| < \varepsilon\right) = 1 \tag{7}$$

where P is the probability measure, x_t is a generated solution by the optimization algorithm (a point in the search space) at iteration t, and ε is a small positive value. According to this definition, two different types of convergence[13] can be defined in optimization algorithms:

1. Convergence to a point (also known as stability analysis): X is any point in the search space (including local optima),
2. Convergence to a local optimum: X is a local optimum of the objective function in the search space[14].

Analysis of convergence to a point is usually conducted for an iterative stochastic optimization algorithm to understand whether the sequence of the generated solutions by the algorithm is convergent. Analysis of convergence to a local optimum is conducted to understand whether the final solution found by the algorithm is at least a local optimum. These two types of convergence have been investigated in detail for different optimization algorithms (Birbil et al., 2004; Dasgupta et al., 2009; Matyas, 1965; Solis and Wets, 1981).

Although analysis of convergence to a local optimum is a powerful tool to understand whether the algorithm is able to locate high quality solutions (local optima), it is also important to estimate how long does it take for the algorithm to locate that solution (Dorea, 1983). Thus, apart from convergence to a local optimum, the expected number of function evaluations until a point within an arbitrary vicinity[15] of a local optimum is visited is also analyzed theoretically. This type of analysis, known as expected first hitting time (EFHT) or runtime analysis, has been conducted for evolutionary algorithms (He and Yao, 2002; Rudolph, 1997). Note that local convergence serves as a prerequisite for EFHT analysis in an optimization algorithm.

In the next three sub-sections limitations in PSO related to these three topics (convergence to a point, convergence to a local optimum, and EFHT) are reviewed.

3.1.1 Convergence to a point

The velocity vector of particles in SPSO grows to infinity for some values of acceleration and inertia coefficients (Clerc and Kennedy, 2002) (this issue is known as swarm

[13] There are also other types of stochastic convergence such as convergence in n^{th} mean and almost surely convergence. Interested readers are referred to (Rudolph, 2013) for more detail on different types of convergence.

[14] The point c is a local minimum of an objective function f over the search space S if there exists an open interval $I \subseteq S$ such that $\forall x \in I \; f(c) \leq f(x)$.

[15] An arbitrary vicinity of a point is defined by a d-dimensional ball around that point with an arbitrary radius.

explosion), which causes particles to leave the search space and move to infinity. As the search space is bounded (as mentioned in section 2), moving outside of the boundaries is not desirable even if there is a better solution (in terms of the objective value) outside of the boundaries. One of the early solutions for this issue was to restrict the value of each dimension of the velocity in a particular interval $[-V_{max},\ V_{max}]$ (Helwig et al., 2013; Helwig and Wanka, 2007, 2008; Shi and Eberhart, 1998b). However, this strategy is not comprehensive enough to prevent the swarm explosion effectively in general case (see (Helwig et al., 2013; Helwig and Wanka, 2007, 2008) for details). The reason is that this strategy restricts only the velocity of particles and not the position of particles, thus, position can still move unboundedly with restricted steps (velocity) (Helwig et al., 2013). One way to prevent particles from moving unboundedly is to restrict the position of the particles as well (Helwig and Wanka, 2007). However, even this strategy is not effective because preventing particles from leaving the boundaries of the search space may keep the particles on the boundaries of the search space and prevent them from effective search. A more fundamental solution to the swarm explosion issue can be achieved through analysis of particles behavior to find out why the sequence of generated solutions might not be convergent (Clerc and Kennedy, 2002; Trelea, 2003; Van Den Bergh, 2002; Poli, 2009; Bonyadi and Michalewicz, 2014d; Cleghorn and Engelbrecht, 2014a), this is known as stability analysis. The aim of this analysis is to define boundaries for the inertia weight and acceleration coefficients in such a way that the positions of the particles converge to a point in the search space, i.e., the position is not divergent. These boundaries are called convergence boundaries throughout the paper[16].

Some simplifications are usually considered to analyze the stability of particles. Usually velocity and position updating rules are analyzed for an arbitrary particle i in a one dimensional space. Also, it is usually considered that $\vec{p}_t^{\,i}$ and $\vec{g}_t$ are not updated during the run (analysis of a particle until it finds its next personal best). Further, the topology of the swarm is ignored in the analysis.

One might argue that the found convergence boundaries under these simplifications are not valid as the simplified system might not exactly reflect the behavior of the original system. However, it should be noted that, in these analyses, the behavior of particles is investigated while they are searching for a new personal best. Considering the one dimensional space for analysis of SPSO is reasonable because all calculations (including generation of the random values on the diagonal of R_{1t}^i) are done in each dimension independently, hence, all analyses in one dimensional case is generalizable to multi-dimensional cases as well (Poli, 2009). Also, validity of the convergence boundaries found when global and personal best are steady and the topology is ignored were studied by Cleghorn and Engelbrecht (2014b,c) experimentally. It was found that, although these conditions simplify the update rules, the calculated convergence boundaries under these conditions is in agreement with that of found under general conditions through experiments.

Three different types of stability analyses can be found in literature that are deterministic model stability analysis, first order stability analysis, and second order stability analysis. The deterministic model stability analysis excludes the stochastic component in particles and assumes that the particles are moved through a deter-

[16]One should note that determining such boundaries is very helpful for parameter setting purposes of SPSO. The reason is that the coefficients that are outside of the convergence boundaries are usually not appropriate for optimization purposes as they cause particles to move unboundedly (recall that the search space is bounded). Hence, such analyses can exclude a huge number of combination of coefficients and limit the search for good combinations to a smaller boundary.

ministic formulation–see (Clerc, 1999; Clerc and Kennedy, 2002) for example. First order stability analysis studies the expected value of the position of particles–see (Trelea, 2003; Van den Bergh and Engelbrecht, 2006; Cleghorn and Engelbrecht, 2014a) for examples–while the second order stability analysis studies the variance of the position of particles–see (Jiang et al., 2007b; Poli, 2009; Liu, 2014) for examples.

Perhaps the first study that considered theoretical analysis of the trajectory of particles in PSO was conducted by Ozcan and Mohan (1999). It was shown that the particles trajectory is of the form of a random sinus wave. Although no parameter analysis or convergence boundary was introduced in that paper, the finding of that paper set a base for further analysis.

One of the earliest attempts to analyze the convergence behavior of SPSO to find convergence boundary[17] was done by Clerc and Kennedy (2002). In that paper, in order to simplify the formulation of update rules, stochastic components (r_{1t}^i and r_{2t}^i) from the system were omitted (deterministic model analysis). It was proven that particles positions generated by SPSO converge to a stable point if

$$\chi = \frac{2k}{\left|2 - c - \sqrt{c^2 - 4c}\right|} \tag{8}$$

where χ (this coefficient is known as constriction factor) is equal to ω (inertia weight), $c_1 + c_2 = c > 4$, $c_1 = \frac{\varphi_1}{\chi}$, $c_2 = \frac{\varphi_2}{\chi}$, and k is a value in the interval $(0, 1]$ (usually set to 1 (Clerc and Kennedy, 2002)). With these settings the value of χ is in the interval $(0, 1]$. The value of k controls the speed of convergence to a fixed point, i.e., the larger the value of k is, the slower convergence to the fixed point will be. According to these coefficients, the velocity update rule is written as

$$\vec{V}_{t+1}^i = \chi\left(\vec{V}_t^i + c_1 R_{1t}\left(\vec{p}_t^i - \vec{x}_t^i\right) + c_2 R_{2t}\left(\vec{g}_t - \vec{x}_t^i\right)\right) \tag{9}$$

where χ is set by Eq. 8, $c_1 + c_2 > 4$, and usually $c_1 = c_2 = 2.05$ that results in $\chi = 0.7298$ (Clerc and Kennedy, 2002). By using these settings each dimension of the velocity vector converges to zero and, consequently, velocity does not grow to infinity. The PSO variant that uses velocity update rule in Eq. 9 is called "Constriction Coefficient Particle Swarm Optimization", CCPSO. The stable point where each particle i converges to is a point between p and g (namely $\frac{c_2 g + c_1 p}{c_1 + c_2}$).

Convergence property of SPSO was also analyzed in (Trelea, 2003). In that study the random components were replaced by their expected values ($r_{1t}^i = r_{2t}^i = 0.5$), that leads to the first order stability analysis. The parameters of the velocity update rule were analyzed to find out how they should be set to guarantee particles settle in their equilibrium. Obviously, $\vec{V}_t^i = 0$ and $\vec{x}_t^i = \vec{p}_t^i = \vec{g}_t$ at the equilibrium point. Also, an equilibrium point for the particles can be any point in the search space. It was proven that the expected value of the position of each particle settles in its equilibrium (that is $\frac{\varphi_2 g + \varphi_1 p}{\varphi_1 + \varphi_1}$) if and only if $\omega < 1$, $\varphi > 0$, and $2\omega - \varphi + 2 > 0$ where $\varphi = \frac{\varphi_1 + \varphi_2}{2}$. Fig 3 shows this relationship between ω and φ.

Particles might behave in two different ways before they converge to their equilibrium: harmonic oscillation and zigzagging. Harmonic oscillation takes place if $\omega^2 + \varphi^2 - 2\omega\varphi - 2(\omega + \varphi) + 1 < 0$ and causes particles to smoothly oscillate around the equilibrium point until they completely settle on that. Experiments showed that

[17] Van Den Bergh (2002) also analyzed the convergence behavior of SPSO in his thesis, however, the related paper to that thesis was published in 2006, that is reviewed later in this survey.

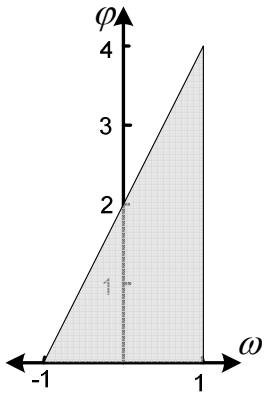


Figure 3: Particles converge to their equilibrium if and only if the value of the parameters and are set in such a way that they lie inside the gray triangle (convergence boundaries).

the oscillation is fast (causes quick convergence) if the values of ω and φ are closer to the "center" (close to the point (0, 1)) of the triangle in Fig. 3. Instead, this oscillation is slower (slower convergence) if the values of ω and φ are closer to the boundaries of the triangle. Thus, with the former setting particles spend more iteration for exploitation, while with the latter setting particles spend more iteration on exploration (see (Črepinšek et al., 2013) for the definition of exploration and exploitation). The zigzagging behavior is observed if $\omega < 0$ or $\omega - \varphi + 1 < 0$. A manual procedure was proposed in (Trelea, 2003) to calculate good settings for these parameters (ω and φ) for a problem at hand. A similar analysis for the behavior of particles before convergence was conducted in (Campana et al., 2010) for SPSO where these findings were confirmed. The parameter setting procedure by Trelea (2003) showed that $\omega = 0.6$ and $\phi_1 = \phi_2 = 1.7$ result in a good performance of the algorithm.

Results reported in (Trelea, 2003), including convergence criteria for the coefficients and convergence to a point between p and g, were also confirmed by Van den Bergh and Engelbrecht (2006) (another first order stability analysis). However, the speed of convergence was investigated in more depth in (Van den Bergh and Engelbrecht, 2006). It was proven that the speed of convergence of the position of particles is directly related to $c = \max\{||\gamma_1||, ||\gamma_2||\}$ where $\gamma_1 = \frac{1+\omega-\varphi_1-\varphi_2+\alpha}{2}$, $\gamma_2 = \frac{1+\omega-\varphi_1-\varphi_2-\alpha}{2}$ and $\alpha = \sqrt{(1+\omega-\varphi_1-\varphi_2)^2 - 4\omega}$, i.e., the larger the value of c is, the faster the algorithm converges. It was also proven that the expected value of the particle's positions converge to a fixed point (namely $\frac{\varphi_2 g+\varphi_1 p}{\varphi_1+\varphi_2}$) if and only if $c < 1$.

The analysis of convergence to a fixed point was also conducted in (Jiang et al., 2007b) from a different point of view. It was proven that the expectation of the position of each particle converges to a point (first order stability analysis) between personal and global best vector ($\frac{\varphi_2 g+\varphi_1 p}{\varphi_1+\varphi_2}$) if and only if $0 \leq \omega < 1$ and $0 < \varphi < 4(1+\omega)$ where $\varphi = \varphi_1+\varphi_2$ (note that this is exactly the same as what was found by Trelea (2003)). However, it was pointed out that the first order stability is not enough to guarantee convergence of particles and second order stability should be also guaranteed. In fact, not only

$lim_{t\to\infty} Ex\left[x_t^i\right]$ should be equal to a fixed value but also $lim_{t\to\infty} Std\left[x_t^i\right]$ should be 0 (see also (Poli, 2009; Poli et al., 2007a)). Through second order stability analysis, it was proven that setting the parameters to $0 \leq \omega < 1$, $0 < \varphi$, and $0 < \beta < \frac{\varphi_2^2(1+\omega)}{6}$, where $\beta = -\varphi\omega^2 + \left(\frac{1}{6}\varphi^2 + \frac{1}{2}\varphi_1\varphi_2\right)\omega + \varphi - \frac{1}{3}\varphi^2 - \frac{1}{2}\varphi_1\varphi_2$, guarantees second order stability of particles positions. As mentioned by Jiang et al. (2007b), this holds even if the global best vector is updated during the run. The analysis showed that the variance of the sequence of positions converges if $\frac{5\phi-\sqrt{25\phi^2-336\phi+576}}{24} < \omega < \frac{5\phi+\sqrt{25\phi^2-336\phi+576}}{24}$, where this time $\varphi = \varphi_1 = \varphi_2$. This analysis was slightly modified due to a small error and the corrected version was presented in (Jiang et al., 2007a). This study also recommended $\omega = 0.715$ and $\phi_1 = \phi_2 = 1.7$ for experimental purposes.

The expected value and the standard deviation of the sequence of generated positions by SPSO were also analyzed in (Poli, 2009; Poli et al., 2007a). It was proven that $lim_{t\to\infty} Std\left[x_t^i\right] = 0$ only if $g = p$ in SPSO. Also, there exists coefficients for SPSO that guarantee second order stability. Hence, a particle does not stop moving unless its personal best settles on the global best of the swarm. Thus, even if the personal best of a particle is updated, it does not stop moving unless its personal best is the global best of the swarm. Indeed, $\vec{x}_t^i$ converges to $\vec{p}_t^i$, and both of them converge to $\vec{g}_t$ and $\vec{V}_t^i$ converges to zero as t grows to infinity. If a particle i has converged to a point and not moved any more, it is said that the particle has stagnated. It was also proven that $lim_{t\to\infty} Std\left[x_t^i\right] = 0$ requires $\phi < \frac{12(\omega^2-1)}{5\omega-7}$ where $\varphi = \varphi_1 = \varphi_2$. This in fact is aligned with what was found by Jiang et al. (2007b) to ensure convergence of the variance.

The first and second order stability were investigated by Garca-Gonzalo and Fernndez-Martnez (2014) for SPSO when a generic distribution for the inertia weight and acceleration coefficients was considered, i.e., inertia weight is multiplied by a random number from an arbitrary probability distribution with expected value μ_ω and variance σ_ω and the random matrices are also generated by an arbitrary probability distribution with expected value μ_ϕ and variance σ_ϕ where $\phi = \phi_1 + \phi_2$. It was proven that SPSO is first order stable if $-1 < \mu_\omega < 1$ and $0 < \mu_\phi < 2(\mu_\omega + 1)$. Also, the algorithm is second order stable if $-a < \mu_\omega < a$ and $0 < \mu_\phi < b$ where $a = \frac{1}{\sqrt{c^2+1}}$, $b = \frac{2(1-(1+c)\mu_\omega^2)}{1+d^2+(d^2-1)\mu_\omega}$, $c = \sigma_\omega/\mu_\omega$, and $d = \sigma_\phi/\mu_\phi$. It was shown that the regions found to guarantee second order stability are embedded within the regions that guarantee first order stability, i.e., if a SPSO is second order stable then it is also first order stable.

Recently, some studies have tried to weaken the assumptions in previous literature to find the convergence regions under more general conditions. As an example, Cleghorn and Engelbrecht (2014a) assumed that the personal best of particles and the global best of the swarm are allowed to move and can occupy an arbitrarily large finite number of unique positions. The main finding of that study was that SPSO is first order stable if $-1 < \omega < 1$ and $0 < \varphi < 2\,(1+\omega)$ where $\varphi = \varphi_1 + \varphi_2 < 4$ under this more general condition. This is indeed the same as what was found by Trelea (2003) and other previous studies through first order stability analysis. Cleghorn and Engelbrecht (2014a) also showed that the topology does not affect the convergence boundary, but it might affect the speed of convergence/divergence.

The second order stability of SPSO was also investigated by Liu (2014). The assumption in that study was that all personal bests can move but not the global best vector. The theoretical study in that paper proved that the convergence boundaries found by Jiang et al. (2007b) as well as Poli (2009) are valid under this assumption. This study recommended $\omega = 0.45$ and $\phi_1 = \phi_2 = 1.55$ based on an extensive experiment

on some benchmark functions.

Analysis of stability was also done for a variant of PSO called "Standard PSO 2011", SPSO2011 (Bonyadi and Michalewicz, 2014d). In that variant, the velocity update rule was revised to:

$$\vec{V}_{t+1}^{i} = \omega \vec{V}_{t}^{i} + \mathcal{H}_i\left(\vec{G}_t^i,\ \vec{G}_t^i - \vec{x}_t^i\right) - \vec{x}_t^i \tag{10}$$

where $\mathcal{H}_i\left(\vec{G}_t^i,\ \vec{G}_t^i - \vec{x}_t^i\right)$ is a spherical distribution with the center $\vec{G}_t^i$ and radius $\vec{G}_t^i - \vec{x}_t^i$, $\vec{G}_t^i = \frac{\vec{L}_t^i + \vec{P}_t^i + \vec{x}_t^i}{3}$, $\vec{P}_t^i = \vec{x}_t^i + \varphi_1 R_{1t}\left(\vec{p}_t^i - \vec{x}_t^i\right)$, and $\vec{L}_t^i = \vec{x}_t^i + \varphi_2 R_{2t}\left(\vec{l}_t^i - \vec{x}_t^i\right)$ ($\vec{l}_t^i$ is the best personal best among the particles in T_t^i[18]) (Zambrano-Bigiarini et al., 2013). The analyses of convergence conducted for previous PSO variants are not applicable for SPSO2011. The reason is that updating velocity and positions in all previous PSO variants were done dimension by dimension that enabled researchers to study particles in a one dimensional space. However, calculation of $\vec{V}_t^i$ in SPSO2011 involves generating random points using a spherical distribution with a variance that is dependent on the distance between $\vec{G}_t^i$ and $\vec{x}_t^i$. In order to repeat the dimension-wise analysis (similar to that of done for SPSO) for SPSO2011 one need to decompose the random point generation procedure done by the spherical distribution into dimension-wise calculations that might need further effort. Hence, the stability analysis for SPSO2011 was done experimentally in (Bonyadi and Michalewicz, 2014d) where it was shown that convergence boundaries for particles in SPSO2011 are dependent on the number of dimensions and they are different from that of other PSO variants (e.g., SPSO). Thus, good choices of coefficient values in previous PSO variants do not necessarily lead to a good performance of SPSO2011.

Summary: Experiments showed that velocity vector grows to infinity for all particles in the swarm for some values of acceleration coefficients and inertia weight (known as swarm explosion). Some studies investigated this issue from the theoretical point of view to find the reasons behind this divergence. Some studies considered deterministic model for this analysis (e.g., Clerc and Kennedy (2002)), some others studied the first order stability of particles (Trelea, 2003; Van den Bergh and Engelbrecht, 2006; Cleghorn and Engelbrecht, 2014a), and the rest considered second order stability (Jiang et al., 2007a; Poli, 2009; Liu, 2014; Garca-Gonzalo and Fernndez-Martnez, 2014). Although some of these analyses were conducted under simplified conditions (e.g., personal best an global best are not updated), further studies (Cleghorn and Engelbrecht, 2014b,c) experimentally showed that the convergence boundaries found under these simplified conditions are in a good agreement with the cases with no simplification, especially for the studies that considered second order stability. Articles that studied the first order stability of particles found that the point each particle converges to is a point on the line segment that connects its personal best and the global best vectors (Trelea, 2003; Van den Bergh and Engelbrecht, 2006). Similar analyses for the first order stability were also conducted under more general conditions Cleghorn and Engelbrecht (2014a) (personal best and global best are updated with some assumptions). In addition to the first order stability, analyses done by Jiang et al. (2007b); Poli (2009) showed that particles do not stop oscillating (convergence of the variance of the particles positions, second order stability) on this line segment until their personal bests fall on the global best of the swarm. Also, second order stability was studied later by Liu (2014) under a more general conditions. As reported by Liu (2014), the combination ($\omega = 0.45,\ \phi = 1.55$)

[18]Note that this variant also follows the general form in Eq. 2 for velocity updating.

works better than other previously used coefficient values on a large benchmark of test functions. Further, as previous studies only considered uniform distribution for random components, Garca-Gonzalo and Fernndez-Martnez (2014) studied the convergence boundaries for both first and second order stability for an arbitrary distribution. Also, first order stability analyses in (Trelea, 2003; Van den Bergh and Engelbrecht, 2006) showed that particles behave differently and converge with different speeds for different coefficient values before convergence. Although there have been many studies that have analyzed the stability of particles, these analyses have been limited to SPSO and have been only recently done for SPSO2011. There are many other PSO variants that are substantially different from SPSO and their convergence behavior is unknown (such as PSO variants described in (Bonyadi et al., 2014b; Wilke et al., 2007b)). Thus, it would be valuable to conduct convergence analysis for these variants as well. It would be also valuable to study benefits of different convergence behaviors, e.g., zigzagging and harmonic (see Trelea (2003)), on different types of landscapes (neutral, rugged, etc.). Such study can potentially lead to an adaptive approach that alters the convergence behavior of the particles according to the type of the landscape during the run. A similar study can be also conducted on the speed of convergence–see (Van den Bergh and Engelbrecht, 2006)). Another potential direction for further studies is to investigate the differences/similarities between PSO variants and Evolutionary Strategy (ES) from theoretical point of view, e.g., how the dynamic of the two methods are different and how these differences, if any, impact the search ability of these methods? Angeline (1998) conducted such comparison from philosophical and experimental points of view; however, there have been many theoretical investigations on both algorithms (ES and PSO) since then that enable researchers to conduct more-in-depth theoretical comparisons between these two algorithms.

3.1.2 Convergence to a local optimum

Convergence to a point ensures that a particle settles on its equilibrium and do not move to infinity. It was proven that all particles converge to $\vec{x}_t^i = \vec{p}_t^i = \vec{g}_t$ and $\vec{V}_t^i = 0$ (stagnation) if coefficients (acceleration and inertia) are in the convergence boundaries (equilibrium point). However, whatever the values of coefficients are, there is no guarantee that the equilibrium point is a local optimum. A particle is said to be *locally convergent* if

$$\forall \varepsilon > 0, \lim_{t\to\infty} P\left(\left|\vec{p}_t^i - X\right| < \varepsilon\right) = 1$$

where X is a local optimum (Bonyadi and Michalewicz, 2014a). An alternative for the above definition (Van den Bergh and Engelbrecht, 2010) is, a PSO algorithm is said to be locally convergent if

$$\forall \varepsilon > 0, \lim_{t\to\infty} P\left(\left|\vec{g}_t - X\right| < \varepsilon\right) = 1$$

Local convergence was first analyzed by Van den Bergh and Engelbrecht (2002) for SPSO. It was observed that SPSO does not perform well when the swarm size is too small. This is called swarm size issue throughout the paper. The explanation given in (Van den Bergh and Engelbrecht, 2002) was that particles in a SPSO with small swarm size (e.g., 2) have larger probability to be stagnated. A variant of SPSO called "Guaranteed Convergent Particle Swarm Optimization", GCPSO (Van den Bergh and Engelbrecht, 2002), was proposed in which the local convergence issue was addressed. In GCPSO, a new velocity update rule was introduced for the global best particle τ_t.

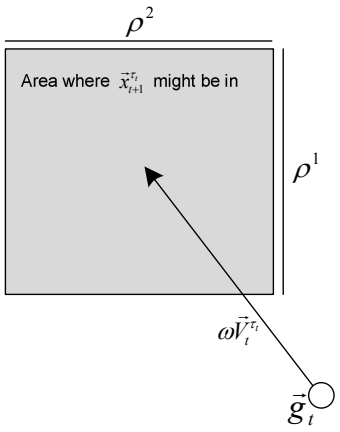


Figure 4: In GCPSO $\vec{x}_{t+1}^{\tau_t}$ is taken randomly from the gray area while all other particles follow the original position and velocity update rule.

Eq. 11 shows the velocity update rule for GCPSO.

$$\vec{V}_{t+1}^{i} = \begin{cases} -\vec{x}_t^i + \vec{g}_t + \omega\vec{V}_t^i + \vec{\rho} & if\ i = \tau_t \\ \omega\vec{V}_t^i + \varphi_1 R_{1t}^i \left(\vec{p}_t^i - \vec{x}_t^i\right) + \varphi_2 R_{2t}^i \left(\vec{g}_t - \vec{x}_t^i\right) & otherwise \end{cases} \tag{11}$$

where ρ^j (the value of the jth dimension of $\vec{\rho}$) is a random value, determined through an adaptive approach, which applies perturbation to the velocity vector. In each iteration of GCPSO, a random location in a hyper-rectangle with the jth side length equal to ρ^j is generated. This random location is moved from $\vec{g}_t$ by the vector $\omega\vec{V}_t^i$ to form the new position for the particle τ_t (global best particle). Fig. 4 shows the hyper-rectangle and the area that $\vec{x}_{t+1}^{\tau_t}$ might be in.

The velocity update rule for the particle τ_t in GCPSO forces that particle to always move (non-zero value for $\vec{V}_{t+1}^{\tau_t}$ for any t). It was proven (Van den Bergh and Engelbrecht, 2010) that SPSO is not locally convergent while GCPSO is locally convergent. Also, GCPSO could provide acceptable results with small swarm size (in reported experiments, the number of particles was set to 2). Stagnation and its relation to small swarm size was also investigated in (Bonyadi et al., 2013; Bonyadi and Michalewicz, 2014b) that are reviewed in section 5 as these methods were used to deal with COPs.

The local convergence issue for SPSO was also discussed in (Schmitt and Wanka, 2013) where it was proven that SPSO is locally convergent for one dimensional problems; however, it is not locally convergent when the number of dimensions is larger than one. The main idea to address the local convergence issue was to detect whether particles are becoming stagnated and then prevent them from stagnation by a perturbation operator. The value $\beta_t^i = ||\vec{V}_t^i|| + ||\vec{g}_t - \vec{x}_t^i||$ was used to detect if the particle i is stagnating. If $\beta_t^i < \delta$ (where δ is a small positive value) then the particle i is expected to approach stagnation situation. In such case the velocity vector of particle i was regenerated randomly within a hyper-rectangle with the side length δ. It was proven that by using this approach for the velocity vector, the algorithm is locally convergent.

The issue of local convergence was investigated by Bonyadi and Michalewicz

(2014a) for SPSO. The formulation for SPSO was revised in that paper as follows:

$$\vec{V}_{t+1}^{i} = \omega \vec{V}_{t}^{i} + \sum_{k \in T_t^i} \varphi_k r_{kt}^i \left(m_{kt}^i \left(\vec{p}_t^k \right) - \vec{x}_t^i \right) \quad (12)$$

where $m_{kt}^i : R^d \rightarrow R^d$ is a function (stochastic or non-stochastic) which applies perturbations on its inputs and $r_{kt}^i \in R$ is a random scalar (rather than a random diagonal matrix) in the interval [0, 1]. This variant was called "Locally convergent Rotationally invariant Particle Swarm Optimization", LcRiPSO (see Section 3.2 for details about rotation invariant property of the algorithm). It was proven that, in LcRiPSO, if the function m is designed in such a way that

$$\forall \vec{y} \in S, \exists A_y \subseteq S, \forall \vec{z} \in A_y, \forall \delta > 0, P\left(|m\left(\vec{y}\right) - \vec{z}| < \delta\right) > 0$$

then the algorithm is locally convergent, where P is the probability measure and A is an open set, i.e., for any input vector $\vec{y}$ in the search space, there exists an open set A which contains $\vec{y}$ and $m(\vec{y})$ can be located anywhere in A. It was proven that a normal distribution with mean equal to $\vec{y}$ and a non-zero variance is an instance of the function m. It was proposed to use a proportion of the last non-zero distance between $\vec{x}_i$ and $\vec{p}_i$ for this variance. The algorithm was applied to 20 benchmark test functions (CEC05 and CEC08 benchmarks) and its results were compared with other PSO variants. Experiments (based on the Wilcoxon test) showed that LcRiPSO outperforms some other PSO variants in most cases (e.g., GCPSO) and it provides acceptable results with small swarm sizes. Note that the time complexity of LcRiPSO depends on the function m. However, the designed instance of the function m (the normal distribution) results in the time complexity of $O(d)$ to update the position of the particles. Also, as all particles in this algorithm are locally convergent and it performs well with small swarm size, this algorithm can be a good candidate to locate niches in a search space–see (Bonyadi and Michalewicz, 2014b) for details.

Summary: Selecting values of coefficients in the convergence boundary prevents particles from moving unboundedly. However, there is no guarantee that the point a particle converges to is a local optimum. If the personal best of a particle is guaranteed to converge to a local optimum, it is said that the particle is locally convergent. If a PSO method guarantees convergence of $\vec{g}_t$ to a local optimum then that PSO method is said to be locally convergent. It has been proven that SPSO is not locally convergent. Also, convergence to a non-optimal solution takes place more frequently if the swarm size is small. To overcome the local convergence issue in SPSO, a mutation operator (e.g., replacing global/personal best vectors by a randomly selected point around the global best vector (Bonyadi and Michalewicz, 2014a; Van den Bergh and Engelbrecht, 2010) or regenerating velocity vector (Schmitt and Wanka, 2013)) may be applied to prevent particles from stagnation. Also, it was proven that these methods are locally convergent (Bonyadi and Michalewicz, 2014a; Schmitt and Wanka, 2013; Van den Bergh and Engelbrecht, 2010). Although a mutation operator addresses local convergence issue, it might slow down the search process (see section 3.1.3). One of the issues in these mutation operators is that the known information from the search space, such as potentially good directions towards better solutions, is completely ignored. In fact, rather than considering a uniform distribution in a hyper-rectangle to pick the random point from (mutation operator), one can consider a normal distribution, preferably with a covariance matrix calculated based on potentially good directions that actually speeds-up convergence to local optima (similar to approaches for ES (Beyer and Schwefel,

2002; Hansen, 2000, 2006)). Note that a simple version of adaptive mutation based on normal distribution was proposed in (Bonyadi and Michalewicz, 2014b)–this paper is discussed later in section 5. However, that simple adaptive approach did not consider covariance matrix calculations.

3.1.3 Expected first hitting time (EFHT)

Recently, the lack of EFHT (runtime) analysis for PSO was pointed out by Witt (2009). The authors argued that most convergence analyses for SPSO (including (Clerc and Kennedy, 2002; Jiang et al., 2007b; Trelea, 2003)) have been limited to the concept of convergence to an attractor and not particularly to an optimum. As GCPSO was the only PSO variant that has been proven to be locally convergent by that time, the EFHT analysis was conducted for this variant. Analysis showed that the convergence speed of GCPSO is asymptotically the same as a simple (1+1) ES–see (Jägersküpper, 2008) for analysis of the EFHT for randomized direct search methods with an isotropic sampling, including (1+1) ES–to optimize the sphere function. However, GCPSO is not rotation invariant while (1+1) ES is. The author concluded that more theoretical analysis on PSO is needed to justify its successes in practice.

A detailed analysis of EFHT for SPSO was also conducted in (Lehre and Witt, 2013) where the algorithm was applied to a sphere function. It was proven that EFHT for SPSO is potentially infinite even when it is applied to a one dimensional sphere. It was shown that in some situations that occur with non-zero probability the algorithm cannot locate the optimal solution even for a one dimensional sphere. Note that the setting of the parameters in that paper was different from that of (Schmitt and Wanka, 2013), where it was proven that SPSO is locally convergent for one dimensional problems. A variant of PSO, called "Noisy Particle Swarm Optimization", NPSO, was introduced (Lehre and Witt, 2013) in which a uniform randomly generated vector in a hyper-rectangle with the side length $\delta > 0$ was added to the velocity vector of each particle. The authors proved that EFHT for NPSO is finite, i.e., NPSO converges to a local optimum in a finite number of function evaluations. It was concluded that more efforts are needed on the EFHT of SPSO to understand how the algorithm works. The same analysis was also done by Bonyadi and Michalewicz (2014d) for SPSO2011, where it was proven that, with specific setting of parameters, EFHT of the algorithm for any one dimensional optimization function is infinite.

Summary: Although addressing local convergence issue (discussed in section 3.1.2) is of a great significance, it is also important to understand what is the expected time (average number of function evaluations) to hit a point within a vicinity of the local optimum (Dorea, 1983). There have not been many attempts to analyze/improve EFHT for PSO variants. A potential reason behind the lack of EFHT analysis in PSO variants is that there are not many PSO variants that have been proven to be locally convergent while local convergence is served as a prerequisite for EFHT analysis in an algorithm. Also, it has been shown that the locally convergent PSO variants perform almost the same as very simple ES–see (Jägersküpper, 2008)–methods in terms of EFHT on a simple problem (i.e., sphere function). Thus, it would be valuable if the local convergence issue in PSO is addressed in a way that EFHT of the algorithm becomes better than other competitive methods, e.g., Differential Evolution (DE) and ES.

3.2 Limitations related to transformation invariance

It is important to investigate whether an algorithm is "transformation invariant" (Wilke et al., 2007b). Usually performance of optimization algorithms is studied under scale,

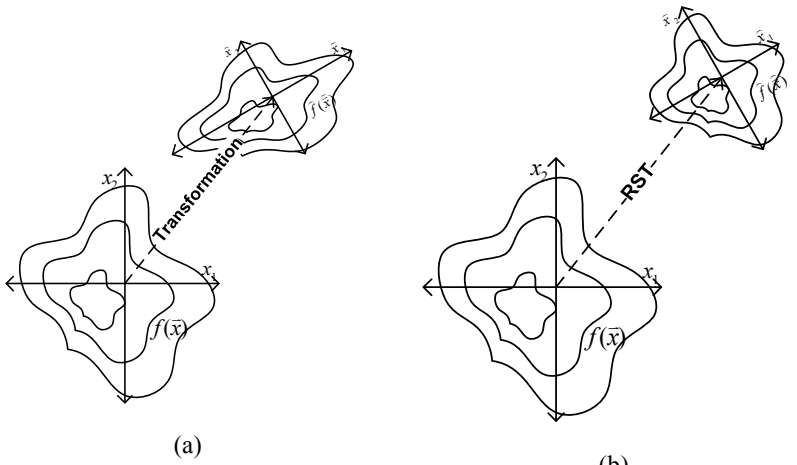


Figure 5: a sample rotated, scaled, and translated search space.

rotation, and translation transformations (we call it RST transformation in this paper). A formal definition for RST invariance is as follows.

Let $S' = \{\vec{y} : sQ\vec{x} + \vec{b}, for\ all\ \vec{x} \in S\}$ where $s \in R$ is a scale factor, $Q \in R^d \times R^d$ is a rotation matrix, and $\vec{b} \in R^d$ is a translation vector. Also, let for any $\vec{x} \in S$, $f(x) = f'(x')$ where $f' : S' \rightarrow R$ and $x' = sQ\vec{x} + \vec{b}$, i.e., the objective value of a point in S is equal to the transformed objective value of the transformed point by s, Q, and $\vec{b}$. Assume that $\hat{x}_t$ is an arbitrary point in S' and $\hat{x}_t = sQ\vec{x}_t + \vec{b}$ where $\vec{x}_t \in S$. An optimization algorithm is RST invariant if:

$$\forall t > 0\ \forall s \in R\ \forall Q \in R^d \times R^d\ \forall \vec{b} \in R^d\ \hat{x}_{t+1} = sQ\vec{x}_{t+1} + \vec{b} \tag{13}$$

where $\hat{x}_{t+1}$ and $\vec{x}_{t+1}$ are the generated points by the algorithm at the iteration $t+1$ in $\hat{S}$ and S, respectively.

In other words, generating a new point in the rotated, scaled, and translated search space (S') is the same as generating a new point in the search space and then rotate, scale, and translate the result (see Fig. 5). For a stochastic optimization algorithm, this condition should be met asymptotically, i.e., the probability that this condition is met increases to 1 during iterations.

The transformations invariance has been investigated for many optimization algorithms (Hansen, 2000; Salomon, 1995). Among rotation, scale, and translation, rotation has received more attention by researchers in the field of PSO. Probably the main reason for this special attention is that rotation potentially makes objective function non-separable (Hansen et al., 2011). Also, dealing with non-separable optimization functions is of more interest as there are many optimization problems with this characteristic. Translation invariance is probably the most basic characteristic among these transformations that an optimization algorithm must have. The reason is that, if an algorithm is sensitive to translation, the algorithm in fact is making an assumption about the location of the optimal solution of the problem at hand. Hence, if the optimal solution is shifted, that assumption is not valid anymore that causes a change on the performance of the algorithm.

Transformation invariance of SPSO was first analyzed in 2007, where it was shown (Wilke, 2005; Wilke et al., 2007b) that SPSO is scale and translation invariant

while it is rotation variant. The reason that SPSO is rotation variant is that the random diagonal matrices in the velocity update rule are rotation variant. In contrast, it was proven that LPSO (see section 2) is rotation, scale, and translation invariant. However, LPSO suffers from a limitation called line search. In LPSO, if $(\vec{p}_t^i - \vec{x}_t^i) \mid\mid (\vec{g}_t - \vec{x}_t^i)$ and $\vec{V}_t^i \mid\mid (\vec{p}_t^i - \vec{x}_t^i)$, particle i oscillates between its personal best and the global best (line search, see also (Bonyadi et al., 2013)) and it is not able to search in other directions. A new PSO variant was proposed in (Wilke et al., 2007b) that was rotation invariant while it did not have the line search limitation. Such algorithm (called "Rotation invariant Particle Swarm Optimization", RPSO) used random rotation matrices rather than random diagonal matrices to rotate the velocity vector. An approximation idea (called exponential map) was used to generate the rotation matrices to reduce the computational expenses arising from matrix multiplication. In the exponential map, the rotation matrix M is generated by:

$$M = I + \sum_{i=1}^{maxC} \frac{1}{i!}\Big(\frac{\alpha\pi}{180}\left(A - A^T\right)\Big)^i \tag{14}$$

where A is a $d \times d$ matrix with elements generated randomly in the interval [-0.5, 0.5], α is the angle that serves as a real scale factor, I is the identity matrix, and $maxC$ is the maximum number of coefficients for the accuracy of the rotation matrix calculation. The generated matrix M is a random rotation matrix with the angle α (set to 3 in all experiments in (Wilke et al., 2007b)). It is clear in Eq. 14 that the matrix M becomes a more accurate estimation of a rotation matrix with the angle α when $maxC$ is larger. In that article the value of $maxC$ was set to 1, limiting the approximation to one term of the summation only. Thus, the approximation error of the random rotation matrix method grows with the rotation angle (α). The time complexity for generating and multiplying the approximated rotation matrix (with $maxC = 1$) into a vector is in $O(d^2)$, including generating the approximated matrix ($O(d^2)$) plus vector-matrix multiplication ($O(d^2)$). Note that for larger values of $maxC$, the order of complexity of generating M grows.

The idea of replacing random diagonal matrices by rotation matrices was further investigated by Bonyadi et al. (2014b). A method for generating accurate random rotation matrices (not approximation like exponential map) was proposed. The method was based on Euclidean rotation matrices that could be used to rotate vectors in any combination of the hyper planes. Two approaches for rotation of the velocity vector were investigated in that paper: rotation in all possible planes (complexity in $O(d^2)$) and rotation in one randomly selected plane (complexity in $O(d)$). Experiments showed that rotation in one plane is slightly worse (in terms of the objective value) than rotation in all planes, however, it is much faster. The random diagonal matrices were replaced by rotation matrices in all planes (generated by the proposed method) in several PSO variants proposed in (Clerc and Kennedy, 2002; Eberhart and Shi, 2001; Nickabadi et al., 2011; PSO, 2006; Shi and Eberhart, 1998b; Van den Bergh and Engelbrecht, 2010; Zheng et al., 2003). Results (based on the Wilcoxon test) showed that PSO variants usually benefit from random rotation matrices.

Potentially, rotation of the search space of a problem makes the problem non-separable (Salomon, 1995). Thus, in order to test the performance of SPSO in dealing with non-separable optimization problems, the algorithm was applied to several problems when their search space were rotated (Hansen et al., 2011, 2008). Results of applying SPSO to these problems were compared with "Covariance Matrix Adaptation ES" (Hansen and Ostermeier, 1996), CMA-ES, and "Differential Evolution", DE (Storn

and Price, 1997). Experiments showed that, although the performance of SPSO in separable optimization problems is acceptable, the algorithm performs poorly when it is applied to non-separable optimization problems. Rotation of the landscape causes the algorithm to perform differently, the algorithm is rotation variant. Although the test cases in that study were quite limited (only four optimization problems were used for the tests), the authors claimed that the poor performance of SPSO in non-separable functions can be generalized to solving other optimization problems. The reason was that as no free lunch theorem (Wolpert and Macready, 1997) is not applicable in the continuous space, it is expected that SPSO performs better in most cases even if the benchmark size is small. However, results showed that SPSO is easily outperformed by CMA-ES and DE in most cases. The authors concluded that the invariance issue in SPSO needs to be investigated in more detail. The same comparison was also conducted in (Auger et al., 2009). This study confirmed the findings presented in (Hansen et al., 2011, 2008), including good performance of SPSO for separable optimization problems and poor performance for non-separable optimization problems. Also, the authors argued that the main reason behind this poor performance is that all calculations in SPSO are done for each dimension separately (Auger et al., 2009), which makes the algorithm rotationally variant.

It was shown (Spears et al., 2010) that not only SPSO is rotation variant, but also particles in SPSO are biased towards the lines parallel to the coordinate axes. In fact, by tracking the positions of particles in the search space during the optimization process, it was found that particles tend to sample more points along the lines parallel to coordinate axes rather than all points in the search space. The authors used these two issues (rotation variant and the bias) to design test problems that are hard or easy for SPSO to optimize.

In order to address the rotation variance issue in SPSO, a PSO variant called "Standard Particle Swarm Optimization 2011", SPSO2011, was proposed (Clerc, 2011; Zambrano-Bigiarini et al., 2013) (see Eq. 10). It was claimed that SPSO2011 is rotation invariant; however, no proper proof for this claim was provided until recently (Bonyadi and Michalewicz, 2014d). A test reported in (Spears et al., 2010) was also conducted to experimentally confirm the theoretical finding about rotation invariance of SPSO2011.

Perhaps (Bonyadi and Michalewicz, 2014a) is the only study that integrates rotation invariance and local convergence in PSO at the same time (see Eq. 12 for the variant proposed in that stucy, called LcRiPSO). It was proven that LcRiPSO is rotationally invariant if the function m is invariant under any affine transformation, i.e.,

$$\forall s \in R, \forall Orth\left(Q\right), \forall \vec{b}, \vec{y} \in R^d sQm\left(\vec{y}\right) + \vec{b} = \hat{m}\left(sQ\vec{y} + \vec{b}\right)$$

where $Orth(Q)$ indicates that the matrix Q is an orthogonal matrix. Also, hatted ($\hat{}$) variables, vectors, and operators are the ones in the transformed space. The time complexity of this variant is in $O(d)$, that is faster than the variant proposed by Wilke et al. (2007b) and Bonyadi et al. (2014b). Note that an instance for the function m is a normal distribution (see section 3.1.1 for details of this instance of m).

Summary: Importance of transformation invariance in iterative optimization algorithms has been emphasized in many articles (Bonyadi and Michalewicz, 2014a; Hansen et al., 2011, 2008; Wilke, 2005; Wilke et al., 2007b). Among all transformations (shear, stretch, rotation, translation, and scale), rotation has received special attention in PSO–related articles because SPSO is sensitive to rotation of the search space, i.e., if the search space is rotated, its performance changes (Wilke et al., 2007b). Note that rotation of search space potentially makes the objective function non-separable. It has been

shown that the poor performance of SPSO in dealing with non-separable search spaces stems from this limitation (being rotation variant) as well (Hansen et al., 2008). The rotation variant is independent from the distribution of the random values on the diagonal of the random matrices (Bonyadi et al., 2014b; Wilke et al., 2007a,b). One way to address rotation variance issue in SPSO is to use random rotation matrices rather than random diagonal matrices in the velocity update rule that was done in (Wilke et al., 2007b) and (Bonyadi et al., 2014b). However, the time complexity of generating random rotation matrices grows in a quadratic rate by the number of dimensions, which makes this approach inefficient to deal with large scale problems. Thus, it would be valuable to investigate other ideas to make the algorithm rotationally invariant while keeping the time complexity of the algorithm linear–see for example rotation in one plane proposed in (Bonyadi et al., 2014b) which is in $O(d)$ or the idea in (Bonyadi and Michalewicz, 2014a). Also, Bonyadi et al. (2014b) showed that the usage of rotation matrices not only makes the algorithm rotationally invariant but also it improves its performance when it is applied to standard benchmarks. Hence, it might be valuable to compare the results of rotationally invariant PSO variants to other optimization algorithms such as CMA-ES and DE. Another potential direction would be to analyze stability of PSO variants that are rotationally invariant (e.g., SPSO2011, RPSO). Note that, this analyses might be different from that of SPSO because all convergence analyses conducted on SPSO considered that the velocity for each particle is updated for each dimension independently (see section 3.1.1). However, this is not the case when rotation matrices are used because the procedure of rotation is not done dimension by dimension–see (Bonyadi and Michalewicz, 2014d) for details. These analyses might lead researchers towards identification of the convergence boundaries for the coefficients for these algorithms. Moreover, one can use the principal components to adjust the search directions to improve the algorithm in non-separable search spaces (see also (Voss, 2005)). As the particles move in the search space, they collect samples from the search space that can be used to estimate regularities such as local gradients and correlations, including multiple correlations or principle components. This information can be used to verify/modify the particles velocities so that the swarm can find better solutions.

4 Modifications of PSO

According to the No Free Lunch (NFL) theorem (Wolpert and Macready, 1997), if an algorithm A performs better than an algorithm B on a subset of optimization problems, it will compensate on some other subsets of optimization problems, i.e., it will perform worse than B on some other subset of problems. However, attempts to improve continuous space optimization methods (e.g., PSO) were intensified as soon as Auger and Teytaud (2007) showed that NFL does not hold for continuous spaces, that was also investigated in (Everitt et al., 2014). Three different approaches were considered for improving PSO: setting parameters, modifying components of the algorithm, and combining the algorithm with other algorithms. Setting parameters refers to setting the topology, coefficients (acceleration coefficients or inertia weight), and population size, modifying components refers to changes of the velocity or position updating rule (including adding new components; modifying the way they are calculated); and combining the algorithm with other algorithms refers to hybridization of PSO with other methods. In the following subsections, PSO variants which have improved the performance of SPSO through setting its parameters (section 4.1), modification of the velocity/position updating rules (section 4.2), and hybridizing the algorithm (section 4.3)

are reviewed.

4.1 Parameters setting

Parameter setting has been a challenge in iterative optimization methods during past several years (Eiben et al., 1999). Two different approaches for setting parameters of an evolutionary algorithm might be considered: parameter tuning and parameter control (Eiben et al., 1999). Parameter tuning refers to setting parameters of an algorithm through experiments to some constant values. Parameter control, however, refers to design of a strategy which changes the value of parameters during the run. Parameter control approaches are categorized into three groups: deterministic, adaptive, and self-adaptive. In deterministic parameter control approaches a rule (called time-varying rule) is designed to calculate the value of a parameter based on the iteration number. In adaptive parameter control approaches a function is designed that maps some feedback from the search into the value of the parameter. In a self-adaptive parameter control approach the parameters are encoded into individuals and are modified during the run by the optimization algorithm.

In this sub-section, articles that have studied different parameters for PSO (topology, coefficients, and population size) are reviewed.

4.1.1 Topology

The concept of topology in PSO was taken from social psychology, the science of how people behave in a society. In societies, each individual communicates with other individuals which it is connected to. This connection is determined by the network of the society that suggests the same concept as the topology in PSO variants does. Probably the earliest attempt tp modify the topology (see section 2 for the definition of topology) in PSO was conducted in (Kennedy, 1999)[19]. The aim of that was that, as different topologies impose different speed of propagation of information among particles, they should affect the explorative and exploitative behavior of the swarm. Four different topologies were tested:

1. Circle (local best or ring) topology: $T_t^i = \{i, \psi_t\}$ where $\psi_t = \underset{l=\{i-1,i,i+1\}}{argmin} \left\{F\left(\vec{p}_t^l\right)\right\}$
2. Wheels topology: $T_t^1 = \{1, \tau_t\}$ where $\tau_t = \underset{l=\{1,\dots,n\}}{argmin} \left\{F\left(\vec{p}_t^l\right)\right\}$ and $T_t^{i\neq 1} = \{1\}$
3. Star (global best) topology: $T_t^i = \{i, \tau_t\}$ where $\tau_t = \underset{l=\{1,\dots,n\}}{argmin} \left\{F\left(\vec{p}_t^l\right)\right\}$
4. Random edge topology: $T_t^i = \{i, j\}$ where j is randomly selected among other individuals.

All these topologies were tested on a benchmark of four functions (Kennedy, 1999). Experiments showed that the global best topology is the fastest communication topology while the ring topology is the slowest in the tested benchmark set. This results in highest exploitative behavior in the former while highest explorative behavior in the latter topology. This finding was also investigated in (Kennedy and Mendes, 2002) and later in (Mendes et al., 2004). Mendes et al. (2004) proposed a more general formulation for CCPSO, called "Fully Informed Particle Swarm", FIPS, and different topologies were tested under that formulation too. All of these studies, however, suffer from the

[19]Two simple topologies were actually introduced earlier in (Eberhart and Kennedy, 1995), however, that study was not focused on the topic of topology

lack of proper statistical analysis of the results or the adequate number test cases that affect the generality of the drown conclusions.

To address the lack of statistical analysis and the number of test cases to draw conclusions about the performance of different topologies, an extensive comparison (60 benchmark test functions) between the global best and local best topologies was conducted by Engelbrecht (2013). Experiments on 60 test functions showed that these two topologies perform almost the same on separable, non-separable, unimodal, and multimodal problems with a slight favor towards the global best topology in terms of the quality of the final solutions. Also, it was experimentally shown that the diversity of the swarm is not consistently higher or lower when either of these two topologies are used. Hence, it appears that the performance of these topologies and the diversity they impose to the swarm also dependents on the structure of the landscape of the problem at hand.

Apart from well-known topologies such as global best and local best, some other topologies (in the static or adaptive form) were also proposed in PSO literature. As an example, a "Hierarchical Particle Swarm Optimization", HPSO, was proposed by Janson and Middendorf (2005). The main idea in that study was that it is usually better to make high quality particles contribute more in updating position of other particles. Thus, particles in HPSO were arranged in a tree structure with a predefined branching factor (a hierarchy). The tree was updated at each iteration based on the quality of the personal best of particles in a way that better particles were placed in higher levels. The velocity of particle i was then updated according to its own personal best together with the personal best of the particle in its parent node in the tree. It was shown that larger values for the branching factor are more appropriate for the beginning of the optimization process while a smaller branching factor performs better at the later stage of the optimization process. Although the idea behind this paper was somewhat new and interesting, as experiments were conducted on a limited number of test functions (6 test functions) and results were not supported by appropriate statistical tests, the good performance of the method is not generalizable.

Selection of a good topology during the run is a natural extension that can be considered in SPSO. This idea was proposed in (Elsayed et al., 2012) where two PSO variants with different topologies, global best topology and random tournament topology, were used in two independent populations. According to the performance of these populations, particles were taken from one and added to the other to make more use of the population that shows better performance. Also, a local search was added to improve the results found by the algorithm. The proposed method was applied to 13 standard benchmark problems and results were compared with other methods such as an ES and a DE using the Wilcoxon test. However, as the results were not compared with other topology adaptation methods, it is not possible to conclude whether the proposed topology adaptation was responsible for the good performance of the algorithm or it was the effect of the local search that improved the performance of the algorithm to a significant level.

As the topology in PSO has its roots in social networks, one might borrow the structure of such networks and apply the idea to communication between particles in PSO. As an example, it has been shown (Albert and Barabsi, 2002) that in real society networks there are a few nodes that are connected to a large number of nodes in the network. Also, most nodes have a small degree, meaning that most nodes are connected to a few nodes in the network. When a new node is added to a real network it usually prefers to attach to the nodes that have more connections, i.e., riches get richer

(a networks with these properties is called a scale-free network). These analogies were adopted to be implemented as a topology for PSO (Zhang and Yi, 2011). Two types of particles (in SPSO) were defined in a swarm: active and inactive. The active particles move around the search space and look for optima, while inactive particles are not updated until they become activated. At each iteration, all inactive particles are activated by connecting to one of the existing active particles. This method was called "Scale-free Fully Informed Particle Swarm Optimization", SFIPSO. Some experiments were conducted to compare the diversity of particles when the proposed topology, ring topology, or global best topology is used. It was found that the proposed topology offers a better diversity for particles in the search space comparing to other tested topologies. However, the experiment related to diversity was simple (only based on the average values) and no statistical test was conducted to make sure the results are generalizable.

Another example of the usage of real networks in PSO literature was the idea of "six degrees of separation" (Newman et al., 2006) in social psychology (any two arbitrary persons are connected to each other by at most six level of nodes that connect their friends) that was used to construct the connection between the particles in the swarm (Gong and Zhang, 2013). Every certain number of generations, k other particles are selected randomly to contribute into update rules of each dimension j of a particle i in the swarm. This in fact enables particles to update their different dimensions using different network of particles, that is a generalization of the concept of topology. The algorithm was applied to 13 standard test functions and the results were compared with some other PSO based methods.

There have been some other good articles related to the topology (Bratton and Kennedy, 2007; Mendes, 2004). These articles addressed analysis of the influence of topology on the performance of SPSO (Mendes, 2004) (an extensive study related to topology), setting parameters (including topology) experimentally (Bratton and Kennedy, 2007), time-varying topologies (Bonyadi et al., 2014a; Emara, 2009), and an adaptive topology PSO (Cooren et al., 2009). However, these studies have been excluded from this review because of the selection criteria (see section 1).

Summary: Communication among individuals in a society is of a great importance for development. This is also true in swarm-based optimization algorithms where topology defines the communication lines among individuals. It was shown (on a limited number of test functions) that the topology influences the speed of convergence and diversity of the swarm (Kennedy, 1999; Kennedy and Mendes, 2002) in SPSO. As an example, global best topology favors exploitative (over explorative) behavior of the swarm and fast convergence to the global best vector while the ring topology favors explorative (over exploitative) behavior and slow convergence to the global best vector (Kennedy and Mendes, 2002). Although these claims appear intuitive, they were disapproved through an extensive experiment conducted by Engelbrecht (2013). That study showed that the diversity of particles and the performance of the algorithm does not only depend on topology, but also depends on the problem at hand and such general claims about topology is not correct. Also, adaptive topologies (e.g., see (Janson and Middendorf, 2005; Zhang and Yi, 2011)) have been shown to be effective on some benchmark test functions to improve the search ability of the algorithm. It seems that topology affects the performance and diversity of particles, however, such measures are also affected by the characteristics of the landscape. Hence, it would be valuable if all of these topologies (ring, global best, hierarchical, etc) are implemented and classified based on their performances or diversities they impose to the swarm according to the problem characteristics (using a decision tree for example) to find out which topol-

ogy can handle the search more effective in which search space. This direction can lead to design an adaptive topology according to the landscape characteristics detected during the run. In addition, there are not many articles on the theoretical aspects of topologies and how they influence the search process—such studies can be considered as a promising future research direction. In fact, this was one of the challenges in PSO introduced in 2007 (Poli et al., 2007b): "How do the details of the equations of motion or the communications topology affect the dynamics of the PSO?", which has not been addressed properly yet!

4.1.2 Coefficients

Different coefficient values in SPSO, i.e., acceleration coefficients and inertia weight, affect the performance of SPSO (see section 3.1.1). Thus, several attempts were made (Clerc, 1999; Shi and Eberhart, 1998a; Eberhart and Shi, 2000; Carlisle and Dozier, 2001) to tune the values of these coefficients for a set of benchmark functions. The best values these studies found for the inertia weight and acceleration coefficients on most cases falls in [0.4, 1.1] and [1.5, 3], respectively. Also, these studies experimentally showed that the coefficients affect the exploration and exploitation ability of the algorithm. In particular, it was shown that larger values for the inertia weight (relative to the acceleration coefficients) result in a better exploration and smaller values for the inertia weight (relative to the acceleration coefficients) result in a better exploitation. Thus, a linear decreasing inertia weight, ω decreased linearly from 0.9 to 0.4 during the run, was tested by Shi and Eberhart (1998b) (this variant was called "Decreasing Inertia Particle Swarm Optimization", DIPSO) on SPSO. However, the lack of accurate experimental analyses can be clearly observed in all of mentioned articles. There were no statistical analyses in these articles and the number of test cases were very limited that makes the drawn conclusions specific to a small number of cases. The reason behind such shortcoming could be the lack of appropriate benchmark functions or the hardware/software in computers at that time. The consequence of such limited experiments could be seen more apparently when it was shown (Zheng et al., 2003) that increasing the value of ω from 0.4 to 0.9 actually works better than decreasing it on some test cases. Hence, the conclusion drawn by Shi and Eberhart (1998b) is not general and the time-varying strategy is actually problem dependent.

The idea of decreasing inertia weight was extended by Chatterjee and Siarry (2006) in a way that a non-linear function was used to decrease the value of inertia weight: $\omega_i = \frac{(maxi-i)^n}{maxi^n}(\omega_s - \omega_e) + \omega_e$, where ω_s and ω_e were starting and final inertia weights, and $maxi$ was the maximum allowed iterations. With $n = 1$, this formulation is exactly the same as that in DIPSO. After experimenting with the sphere function, the parameters of this variant were set to: $\omega_s = 0.2$, $\omega_e = -0.3$, $n = 1.2$, and $maxi = 2,000$. This method with mentioned settings was compared with other optimization methods such as a genetic algorithm and a deferential evolution on 17 benchmark problems. Although experiments showed that the proposed method achieves high quality solutions in less number of function evaluations, the results were compared based on the average values that makes the final conclusions unreliable.

Although inertia weight influences exploration and exploitation behavior of the algorithm, acceleration coefficients also play important role in this regard. A time-varying approach was proposed (Ratnaweera et al., 2004) where the acceleration coefficients were changed during the run (the method was called "Time Varying Acceleration Coefficient Particle Swarm Optimization", PSO-TVAC). In that method the value of φ_1 (cognitive weight) decreased while the value of φ_2 (social weight) increased during the

run of the algorithm. Thus, particles tend to conduct exploration at the beginning and exploitation at the latter stages of the optimization process. The authors further modified PSO-TVAC by considering an adaptation rule to reinitialize the velocity of particles and another variant (Ratnaweera et al., 2004) called "self-organizing Hierarchical Particle Swarm Optimization", HPSO-TVAC, emerged. It was also proposed to reinitialize the velocity of particles when its value becomes zero (with some precision). The authors concluded that, based on the results, HPSO-TVAC is the best choice among tested methods. This conclusion was supported by only 5 test functions and results were analyzed with regard to the mean and standard deviation values. Because the number of test problems were very limited, the derived conclusions can not be generalized and a more through experiment might be needed to confirm these findings. The idea of reinitializing each dimension of velocity when it is zero is very similar to that of proposed in (Schmitt and Wanka, 2013) (see section 3.1.2) where velocity is reinitialized whenever the particle was stagnating. Hence, although a proper proof might be needed for the local convergence of an algorithm, it seems that HPSO-TVAC overcomes the stagnation and, most likely, is locally convergent.

As experiments confirmed potential benefits of time-varying approaches on some test functions, an adaptive approach for setting the value of coefficients during the run can be also beneficial as it adjusts the coefficients values according to the current situation. An example of such idea was proposed by Arumugam et al. (2008) where the values of inertia weight and acceleration coefficients were related to the objective value of the personal best and global best vectors, i.e., smaller acceleration coefficients and larger inertia weight if the personal best of a particle is not as high quality as the global best vector. Also, the authors experimented with mutation and crossover operators to understand if these operators can improve SPSO. The experiments were conducted on 13 standard test functions and mean, variance, median, minimum, and maximum of found solutions were compared with that of other methods. It was concluded that the tested PSO variants most likely benefit from crossover and mutation operators according to the experiments. Also, the proposed adaptation showed a good potential to enhance the algorithm to find better solutions in the tested problems.

Another adaptive approach, called "Adaptive Particle Swarm Optimization", APSO, was proposed by Zhan et al. (2009) in which three modifications took place: adaptive changes of the inertia weight according to the distribution of particles, adaptive changes of the acceleration coefficients according to the evolution state, and new updating rules in specific evolution states. Evolution state was defined in that paper as the states that a particle might be in, including exploration, exploitation, convergence, and jumping out. The distance of the position of particles from the global best vector was used as a measure to identify the evolution state of particles. In addition to changing the values of acceleration coefficients, a perturbation procedure was applied to the global best particle when all particles are close to each other. The concept of "closeness" was defined by a fuzzy membership function. Although this strategy might help the algorithm to converge to a local optimum, no theoretical discussion was provided to prove this statement. APSO was compared with some other PSO variants when they were applied to 12 test functions. Results showed that APSO is significantly (based on a t-test[20] better than other tested PSO variants in most cases. The selection of the test

[20]The t-test is not the best choice to be used to compare these results because t-test assumes that the distribution of the data is normal (or close to normal). However, this assumption might not be true for the results of an algorithm. A better alternative statistical test for this data is the Wilcoxon test that does not consider such assumption for the distribution of the data (see Derrac et al. (2011) for more information).

functions was, however, not convincing as the optimal solution of most selected functions were in the center of the coordinate axes[21]. Also, because the Euclidean distance between particles was used as a measure of evolution state, the time complexity of calculations was in $O(d^2)$ that is slower than that of SPSO. Such higher complexity might be negligible for problems with small number of dimensions (30 dimensions for example) but causes significant slowdown of the algorithm when the number of dimensions is higher (1,000 dimensions for example).

In order to balance exploration and exploitation behavior of the particles, another adaptive approach, called "Adaptive Inertia Weight Particle Swarm Optimization", AIWPSO, was proposed (Nickabadi et al., 2011). In AIWPSO the value of inertia weight was adaptively changed during the search process. The authors proposed a linear relation between the inertia weight in the current iteration and the number of particles which were improved in the previous iteration. The idea behind this approach was that if success rate is high then it can be concluded that the current direction is promising and, hence, it is better to have a higher value for the inertia weight so that particles are encouraged to move with larger steps towards the current direction. However, when the success rate is low, the current direction is not promising and hence the inertia weight is decreased. Although the idea was simple, experimental results on 15 benchmark test functions showed a considerable improvement (in terms of mean and standard deviation) in solution qualities in comparison to many other adaptive/time-varying strategies for controlling the value of inertia weight.

Recently, another adaptive approach was proposed in which both inertia weight and acceleration coefficients were modified during the search (Leu and Yeh, 2012). The method used a similarity measure called grey relation (Deng, 1989) (the PSO variant was thus called "Grey Particle Swarm Optimization", GrPSO). The idea was based on the following observation: if a particle is less similar to the global best vector (relative to the other particles) then it is in the exploration phase. Thus, larger values for inertia and cognitive weights for that particle are preferred. Also, as the value of social and cognitive weight were related to each other by: $\varphi_1 + \varphi_2 = 4.0$ (Clerc and Kennedy, 2002), changing the value of cognitive weight enforces changes the social weight as well. Experiments were done for 12 different standard test functions and results were compared with other PSO variants, such as HPSO-TVAC and APSO. As the results were compared only based on the mean values, the validity of conclusions (e.g., GrPSO outperforms other methods on the tested functions) is not acceptable.

There have been some other quality studies that have proposed different approaches in setting the coefficients of SPSO such as (Bratton and Kennedy, 2007; Ghosh et al., 2010; Shi and Eberhart, 2001). For example, in (Bratton and Kennedy, 2007) a set of parameters for SPSO were suggested which resulted in a good performance of the algorithm on some benchmark problems. Also, two inertia adaptive PSOs were proposed in (Ghosh et al., 2010; Shi and Eberhart, 2001). However, all these articles have been excluded from this review because of the selection criteria (see section 1).

Summary: There have been many studies to set the inertia weight (ω) and acceleration coefficients (φ_1 and φ_2) of SPSO. Some studies tried to tune these parameters (Carlisle and Dozier, 2001). However, tuning is done on a limited number of test functions and the results are not generalizable. Hence, a parameter control approach

[21] Recall that the position of particles in SPSO is biased towards the solutions on the coordinate axes (see section 3.2) that enables the algorithm to perform better when such solutions are optimum. Hence, a benchmark of problems that have their optimum solutions on the coordinate axes, e.g., center of the coordinate, can not test the performance of the algorithm thoroughly.

is usually preferable. For instance, several time-varying approaches were introduced to change the values of coefficients of SPSO during the run. The main idea for time-varying approaches was based on the fact that usually encouraging exploration at the beginning and encouraging exploitation at the latter stages of the optimization process is beneficial. Also, a large value of inertia and cognitive weights (φ_1) relative to the social weight usually encourages exploration (Ratnaweera et al., 2004; Shi and Eberhart, 1998b) and a large value for social weight (φ_2) relative to the inertia and cognitive weights usually encourages exploitation around the best ever-found solutions. Thus, one can change the values of inertia weight and acceleration coefficients during the run accordingly (Ratnaweera et al., 2004; Shi and Eberhart, 1998b; Zheng et al., 2003). It is, however, not easy to determine how fast these values should be changed during the run. Indeed, the speed of these changes depends on the search space as if the search space is very smooth then radical changes of the coefficients might not be desirable. An adaptive approach to control these variables can be also beneficial as it is independent from the landscape of the problem (see (Arumugam et al., 2008; Leu and Yeh, 2012; Nickabadi et al., 2011)). The main challenge in adaptive approaches is to recognize which behavior (explorative or exploitative) is preferable at the current stage of the search (so called evolution state in (Zhan et al., 2009)). The distance between particles can be used as a measure to determine the evolution state (used in (Zhan et al., 2009)), however, this calculation increases the time complexity of the algorithm. Similarity measure (Leu and Yeh, 2012) and success rate (Nickabadi et al., 2011) are two other candidate measures to determine the state of the search, however, there is no concrete evidence (apart from experiments) to show that these strategies are effective in all situations. It seems that there have been many advances on the parameter setting of SPSO and different approaches have been introduced including time varying, adaptive, and tuning. However, most findings in this topic were either supported by a limited number of experiments or were not discussed through proper statistical/theoretical analysis. One potential future direction related to the parameter tuning would be to study the effects of parameters on a large benchmark of test functions (e.g., CEC or BOBB benchmark sets) using some statistical tests. Another interesting topic related to the parameter tuning would be to study the performance of the algorithm according to the landscape characterization and then classify the results (see Caamaño et al. (2010) for example) based on the achieved performance. Also, comparison (philosophical discussion, theoretical, or experimental) between different adaptive approaches and measures to determine the evolution stage can be very beneficial for further enhancements of the algorithm. Another possibility would be to use analysis presented in (Trelea, 2003) (section 3.1.1) to adaptively set behavior of the particles (oscillating, zigzagging, and convergence) according to the current state of the particle (evolution state).

4.1.3 Population size

Generally, most iterative search methods encourage exploration during the early stages of the iterative process while encourage exploitation in the latter stages. It is also well-known that concentrating individuals only on the area of the highest potential during the exploitation phase is usually beneficial. Thus, as this area is small relative to the whole search space, a small swarm performs almost as effective as a large swarm to find better solutions. Hence, it might be better to reduce the population size at the latter stages of the optimization process to save on function evaluations. Probably the first attempt to design an adaptive population size in an evolutionary algorithm was in 1994 where a genetic algorithm with a varying population size (Arabas et al., 1994) was

proposed. The idea of adapting population size was adopted (Chen and Zhao, 2009) for PSO later in 2009 (the method was called "Ladder Particle Swarm Optimization", LDPSO). In LDPSO, at every predefined period of time, the diversity of the swarm was evaluated and the swarm size was adjusted accordingly. The diversity measure was based on the average Hamming distance between particles. Comparisons showed that the algorithm outperforms some other PSO variants in terms of the solution quality when it was applied to 5 standard benchmark problems. However, due to the lack of statistical analysis and the limitation of the number of test problems, the validity of the comparisons and conclusions are questionable.

A population manager, a solution sharing technique, and a search range sharing strategy were proposed in (Hsieh et al., 2009) (this variant was called "Efficient Population Utilization Strategy Particle Swarm Optimization", EPUS-PSO). The population manager in EPUS-PSO adjusted the population size by:

1. removing a particle from the swarm when global best vector was improved at least once in the previous k consecutive iterations,
2. adding a new particle to the swarm if the global best vector was not changed in the previous k consecutive iterations,
3. replacing an existing one when adding is not possible (the upper bound of the number of particles in the swarm has been reached).

EPUS-PSO also used the global best vector in the velocity update rule of each particle with the probability P and used the personal best of another particle (selected by a tournament selection approach) with probability $1 - P$ (this approach was called solution sharing strategy). The aim of the solution sharing strategy was to give a chance to particles to learn from other particles rather than always using the global best vector in the velocity update rule. In addition, a solution range sharing strategy was presented to prevent the particles from premature convergence (Hsieh et al., 2009). The algorithm was applied to 15 benchmark test functions, together with their rotated versions in 10 and 30 dimensional problems. Results (based on the average and standard deviation) showed that the algorithm is comparable with other PSO variants.

A strategy called "Incremental Social Learning", ISL (Montes de Oca and Stützle, 2008), was used to set the population size of SPSO during the run (Mondes de Oca et al., 2011). ISL suggests an increasing population-size approach that in some cases facilitates the scalability of systems composed of multiple learning agents. Two variants of PSO were proposed based on ISL idea. In the first variant, called IPSO, whenever the algorithm could not find a satisfactory solution, a new particle was added to the population. In the second variant (called IPSOLS, IPSO with a local search), a local search approach was run to gather local information around the current position. If the local search procedure was not successful (no better solution was found around the current position), it meant that the particle was already in a local optimum. In this situation, a new particle was added to the swarm. To set the position of this new particle, a social learning approach was used in which the personal best location of one of the existing particles (called "model" particle) was considered for the location of the new particle. The algorithm applied to 12 standard optimization test functions and its results were compared with other PSO methods. The median and mean of the results were compared in the paper. Results showed that the proposed method is capable of finding high quality solutions and it is comparable with other methods on the tested problems.

Summary: It is clear that majority of particles are very close to each other at the exploitation phase to exploit better solutions close to the best found solution so far. In contrast, during the exploration phase, particles are far from each other to explore the search space to locate as many basins of attractions as possible. Thus, a large population size during the exploration phase and small population size during the exploitation phase seems to constitute a reasonable choice (Chen and Zhao, 2009). However, there are two challenges in setting the population size during the run. The first challenge is to determine which of these behaviors (exploration/exploitation) are preferred at the current situation (iteration) and set the population size accordingly, which can be considered as a direction for further research. For example, in (Chen and Zhao, 2009) the diversity of the swarm was considered as an indication of the exploration/exploitation tendency of the swarm. The second challenge is that adding/removing a new particle to the swarm is not a trivial task. For adding for example one need to determine the position, velocity, and personal best of the particle that is going to be added to make sure the new particle can improve the search ability of the swarm. Also, theoretical studies along the population size, including dynamic approaches, is completely missing in the literature related to PSO. One can study the effect of population size on the search ability or EFHT of the algorithm. Further, diversity of the swarm is another related topic that can be investigated under different test functions or theoretically.

4.2 Modification of the velocity/position updating rules

Introduction of inertia weight in OPSO was probably the first modification of the velocity in the original algorithm (Shi and Eberhart, 1998a) (this algorithm, called SPSO, was reviewed in section 2). The impact of inertia weight was analyzed in (Shi and Eberhart, 1998b) together with other studies reviewed in previous sections.

An issue called "two steps forward one step back" was identified (Van den Bergh and Engelbrecht, 2004) in SPSO: as all dimensions of the position of particles are updated at every iteration, there is a possibility that some components in this vector are moved closer to a better solution, while others actually are moved away from that good solution. As long as the objective value of the new position vector is better than the objective value of the previous position, SPSO assumes that the solution has been improved, no matter if some components have moved away from the good solution. To address this issue, it was proposed to divide the search space to d sub-problems (each variable becomes a sub-problem) and different variables are updated by different sub-swarm—this is called "Cooperative Particle Swarm Optimization", CPSO (Van den Bergh and Engelbrecht, 2004). Although CPSO performed better than many other variants of PSO, it had two potential issues (Van den Bergh and Engelbrecht, 2004) as follows:

- It might converge to a pseudominima, i.e., minima that were generated because of the partitioning of the search space,
- Its performance depends on the correlation of the sub-problems.

As SPSO does not suffer from these issues, a hybrid method was also proposed (Van den Bergh and Engelbrecht, 2004). Results in that paper also was based on a small benchmark (5 test functions) and no statistical test was presented. Note that CPSO is not guaranteed to converge to a local optimum as it is constructed from SPSO that is not locally convergent by itself (an stagnation scenario was also presented in that paper that is another support to show that the algorithm is not locally convergent).

Further, as the sub-swarms are indeed instances of SPSO, the algorithm is sensitive to rotation. This can be also concluded from the results reported in that paper where the performance of CPSO is tested under rotation of the search space.

Global best vector in SPSO is considered as an indication of a potentially high-quality area in the search space. Thus, it would be worthwhile to use this location together with the current location of a particle to generate a new location for the particle. This idea was used in a PSO variant called "extrapolation Particle Swarm Optimization", ePSO (Arumugam et al., 2009), in which the position update rule was revised as follows:

$$\vec{x}_{t+1}^{i} = \vec{g}_t + \alpha\vec{g}_t + \gamma\left(\vec{g}_t - \vec{x}_t^{i}\right) \tag{15}$$

where $k_1 = k_2 = e^{\frac{-current\ iteration}{\max number\ of\ iterations}}$, $\alpha = k_1 r_t^i$, $\gamma = k_1 e^{k_2\beta}$, r_t^i is a random value between 0 and 1, and $\beta = \frac{f(\vec{g}_t) - f\left(\vec{x}_t^{i}\right)}{f(\vec{g}_t)}$ (there is no velocity vector in this variant). The term $\vec{g}_t + k_1 r_t^i \vec{g}_t$ generates a random point around the global best vector. Note that, as k_1 becomes smaller over time (i.e., with growth of the number of iterations), the generated points by $\vec{g}_t + k_1 r_t^i \vec{g}_t$ become closer to $\vec{g}_t$. This results in better exploitation around global best vector at the latter stage of the optimization. The last term $k_1 e^{k_2\beta}\left(\vec{g}_t - \vec{x}_t^{i}\right)$ determines the step size that the current position ($\vec{x}_t^{i}$) takes to go towards the global best vector. This step size is controlled by the objective value of the current location and the objective value of the global best vector. Experiments on 13 standard optimization test functions showed that ePSO performs better than SPSO on almost all of the tested functions. This conclusion was supported by the Wilcoxon test. Although experiments were designed very carefully, the position update rule of ePSO is sensitive to the translation of the search space. The reason is, if we calculate the generated points by the algorithm in a translated space we get: $\hat{x}_{t+1} = \hat{g}_t + \alpha\hat{g}_t + \gamma\left(\hat{g}_t - \hat{x}_t^{i}\right)$ that is equal to $\vec{g}_t + \vec{b} + \alpha(\vec{g}_t + \vec{b}) + \gamma\left(\vec{g}_t - \vec{x}_t^{i}\right)$. Also, $\vec{x}_{t+1} + \vec{b} = \vec{g}_t + \alpha\vec{g}_t + \gamma\left(\vec{g}_t - \vec{x}_t^{i}\right) + \vec{b}$. This means that $\hat{x}_{t+1} \neq \vec{x}_{t+1} + \vec{b}$, hence the algorithm is not translation invariant[22] in the general case (see definition of transformation invariance in Section 3.2). Note also that α is not zero, nor reduces to zero (the minimum value of $alpha$ is $\frac{1}{e}$) during iterations. As translation invariance is a very important property of an optimization algorithm (see Section3.2 and also (Hansen et al., 2011) for the importance of translation invariance), ePSO is not an effective optimization method and its good results on the benchmark set cannot be generalized in any way.

According to the velocity update rule of SPSO, attraction towards personal or global best vectors does not depend only on the acceleration coefficients, but also depends on the average values of $\vec{p}_t^{i} - \vec{x}_t^{i}$ and $\vec{g}_t - \vec{x}_t^{i}$. It was observed that, in SPSO, the value of $\vec{p}_t^{i} - \vec{x}_t^{i}$ is usually smaller than $\vec{g}_t - \vec{x}_t^{i}$ for any particle i and any iteration t that results in larger attraction towards $\vec{g}_t$ than $\vec{p}_t^{i}$. Hence, particles tend to concentrate around $\vec{g}_t$ that reduces the diversity of the swarm (Liang et al., 2006). To overcome this issue, a variant of PSO called "Comprehensive Learning Particle Swarm Optimization", CLPSO, was proposed (Liang et al., 2006). In this variant, the velocity update rule was modified to the following form:

$$\vec{V}_{t+1}^{i} = \omega\vec{V}_t^{i} + \varphi R_t^{i}\left(\vec{P}_t^{i} - \vec{x}_t^{i}\right) \tag{16}$$

[22]The translation variance of this algorithm can be also shown in a more intuitive way. In this algorithm, $\vec{g}$ is the vector that connects the center of coordinates to the point g (position vector g). This vector is added to another position vector ($\alpha\vec{g}$). If the coordinate system is shifted, the vectors that represent the position of g is also changed, and, hence, this addition does not result in the same result. Note however that if $\alpha\vec{g}$ was replaced by $\alpha(\vec{g} - \vec{x}_i^t)$ then this algorithm was invariant from translation.

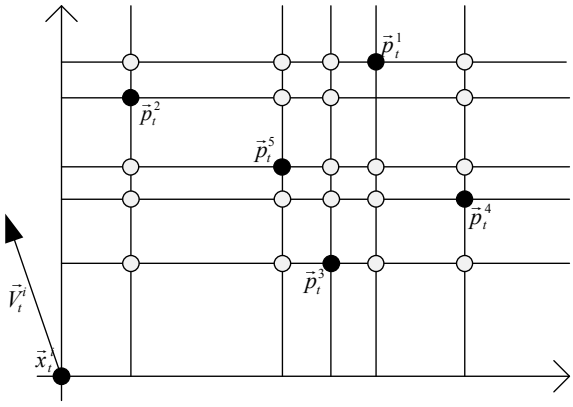


Figure 6: potential locations for $\vec{P}_t^i$ in CLPSO have been represented by grey circles and personal bests of other particles have been presented by black circles.

where the value of the jth dimension in $\vec{P}_t^i$ (shown by $\vec{P}_t^{i,j}$) is calculated by $P_t^{i,j} = p_t^{l_i(j),j}$ where $l_i(j) \in \{1, 2, \ldots, n\}$, which refers to the index of a particle. In fact, each dimension of $\vec{P}_t^i$ is potentially taken from the personal best of different particles. Fig. 6 shows potential locations where $\vec{P}_t^i$ might be selected from for a particle i in a swarm with the size 5 as an example. The value of $l_i(j)$ was set to i with a probability, otherwise, a tournament selection was applied to select the value of $l_i(j)$. Results of application of the algorithm to 16 benchmark test functions showed that the algorithm outperforms several other PSO variants on multimodal functions, while it performed almost the same as the others on unimodal functions. The discussions over the results were supported by the mean and variance of the found solutions by the algorithms. It is apparent that the algorithm can be stagnated and, hence, it is not locally convergent (if all particles are at the same point in the search space). Moreover, the algorithm is not rotationally invariant as it uses the random diagonal matrices the way it was used in SPSO. This can be also observed by comparing the results reported in that paper for rotated and original optimization test functions.

The good performance of CLPSO to find high quality solutions on multimodal optimization problems stems from its good ability in exploration. Also, as it was mentioned, CLPSO does not perform well in unimodal optimization problems, which reflects weak ability of the algorithm in exploitation. To improve the ability of the algorithm for exploitation, CLPSO was revised by Huang et al. (2010) in a way that it emphasized learning from elite particles (the particles with the best personal best vectors in the swarm). The new variant was called "Example-based Learning Particle Swarm Optimizer", ELPSO. Eq. 16 was revised as follows:

$$\vec{V}_{t+1}^i = \omega \vec{V}_t^i + \varphi_1 R_{1t}^i \left(\vec{P}_t^i - \vec{x}_t^i\right) + \varphi_2 R_{2t}^i \left(\vec{G}_t - \vec{x}_t^i\right) \tag{17}$$

where $\vec{P}_t^i$ was defined similar to the one in CLPSO (except the strategy of setting $l_i(j)$ which was selected randomly by a uniform distribution over all particles) and each

dimension j of $\vec{G}_t$ (shown by $\vec{G}_t^j$) is randomly taken from a set of previously sampled positions by $\vec{g}_t$. It was shown that the searching interval of each dimension[23] by each particle is larger than what it was in CCPSO and CLPSO, which indicates a better diversification in the swarm. Experiments were conducted on 16 benchmark test functions and t-test was used to compare the results. These experiments showed that ELPSO is more effective than CLPSO in multimodal and unimodal optimization problems. However, the issue of local convergence and rotation variance still exist in ELPSO as well. A very similar approach to ELPSO was also proposed in (Nasir et al., 2012).

To prevent SPSO from premature convergence[24], the velocity update rule of SPSO was revised (Xinchao, 2010) and a method called "perturbed Particle Swarm Algorithm", pPSA, was proposed. In pPSA, the vector $\vec{g}_t$ in the velocity update rule of the standard PSO was substituted by $N\left(\vec{g}_t, \sigma^2 I\right)$, where N is the normal distribution and σ is the standard deviation. The value of σ was set through a very simple time-varying strategy. The algorithm was applied to some standard unimodal/multimodal optimization benchmarks. Results showed that the algorithm performs better than SPSO in terms of the quality of solutions and robustness. In comparison to GCPSO, results showed that both methods are almost the same on multimodal functions while GCPSO performs better than pPSA in unimodal functions. Results were compared based on mean, median, and standard deviation, that looks enough to show the advantages. No theoretical analyses were included in that article, however, it seems the algorithm can escape stagnation and guarantee local convergence as it uses a mutated global best vector with non-zero variance. In addition, the algorithm is the same as SPSO in terms of transformation invariance, i.e., rotation variance while scale and translation invariance–see (Bonyadi and Michalewicz, 2014a) for the proof. The main reason is the usage of random diagonal matrices that forces the algorithm to conduct the search dimension by dimension.

Diversity of particles is directly related to the explorative and exploitative behavior of the particles. An algorithm called "Diversity enhanced with Neighborhood Search Particle Swarm Optimization", DNSPSO, was proposed (Wang et al., 2013) in which the explorative behavior was controlled by enhancing the diversity of the particles in the swarm. At each iteration, the position and velocity of each particle is updated by the rules in SPSO. The generated position $\vec{x}_{t+1}^i$ is then combined with $\vec{x}_t^i$ to generate a trial particle. The combination is done by taking either the value of $x_t^{i,j}$ or $x_{t+1}^{i,j}$ for each $j \in \{1, \ldots, d\}$ with probability Pr. The personal best of the particle i is updated according to the trial particle. Also, two local searches were used to search the neighborhood of the personal best and global best vectors, which actually improved the exploitation ability of the algorithm. The usage of these neighborhood search strategies was shown to be beneficial for accelerating the convergence rate (experimentally tested). The algorithm was applied to 30 benchmark functions, 10 of them with up to 50 dimensions and 20 of them with 1000 dimensions, and the results were compared with other methods such as CLPSO, APSO, and CPSO using Friedman test. Comparisons showed that the algorithm is comparable with others for both group of test functions.

[23]The term "searching interval of a dimension" refers to the average of size of the interval that is sampled by the optimization algorithm in a dimension. The larger the value of searching interval is, the more diverse the individuals have been.

[24]The term "premature convergence" is usually misused in literature and different researchers define it differently. It sometimes refer to converging to a local optimum and addressing stagnation, while sometimes refer to escaping from a local optimum and finding better solutions that might not be necessarily local optimum, but better than what was found previously. In this context, it refers to escaping stagnation and converging to a local optimum.

There have been some other modifications proposed for update rules of SPSO (Chang-Huang and Jia-Shing, 2009; Kaucic, 2013; Krink et al., 2002). For example, in (Krink et al., 2002) a strategy was proposed to maintain the level of diversity of the particles, in (Chang-Huang and Jia-Shing, 2009) SPSO without CI component was compared with SPSO without SI component, and in (Kaucic, 2013) a multi-start method was experimented with. However, these studies have been excluded from this review because of the selection criteria defined in section 2.

Summary: Many different approaches have been proposed for the velocity update rule, e.g., attraction by global best and personal best vectors while considering momentum (Shi and Eberhart, 1998a), cooperative co-evolution (Van den Bergh and Engelbrecht, 2004), comprehensive learning approaches (Liang et al., 2006), to name a few. Recently, comprehensive learning approaches in PSO have gained a lot of interests (Changhe et al., 2012; Huang et al., 2010; Tang et al., 2011). The reason behind this interest is that these approaches result in a better exploration behavior by the particles (Liang et al., 2006). It is thus worthwhile to follow this direction in more details from convergence and invariance point of views to design a PSO variant with enhanced and scalable performance. Also, the cooperative co-evolution idea for PSO used SPSO as sub-swarms to find optimum solutions. However, this choice causes the algorithm to have the local convergence and transformation variance issues. Hence, the cooperative co-evolution can benefit from a locally convergent and transformation invariant algorithm (e.g., LcRiPSO (Bonyadi and Michalewicz, 2014b)). This idea can be combined with problem decomposition proposed by Omidvar et al. (2014) to improve the performance of the optimization algorithm. Such decompositions ensure preventing the emergence of pseudominima that was one of the issues in (Van den Bergh and Engelbrecht, 2004). Another potential direction for further studies is to investigate differences between PSO, DE, and ES methods. Once these differences were identified, it would be possible to borrow ideas from ES and DE methods to improve the velocity/position update rules in PSO. In addition, because both updating rules in PSO contain vector calculations, it would be valuable to do these calculations in the hyper-spherical coordinates rather than Cartesian coordinates. One benefit in hyper-spherical coordinates is that calculations of rotation of the vectors are easier, which potentially enables the algorithm to deal with non-separable problems (see section 3.1.3) more efficiently.

4.3 Hybridization

Usually optimization methods have their merits and drawbacks to deal with different optimization problems. For example, the Newton's method is locally convergent (under some assumptions, see Kelley (1999)) and is able to find a local optimum usually fast (i.e., effective exploitation). Also, stochastic search methods are usually effective in locating many basins of attractions (effective exploration). In hybrid methods, the aim is to combine different optimization methods in a way that the resulted method has the merits of the both. In this sub-section, articles that have combined PSO with other methods are reviewed. Many combinations are possible, for instance using one method to adjust parameters of the other (see (Parsopoulos and Vrahatis, 2002b) and (Alatas et al., 2009) where DE and a chaotic map were used to set the coefficients of SPSO during the run, respectively), or running different methods iteratively to improve the outcome of one anothe–see (Kao and Zahara, 2008) where GA was combined with SPSO.

Two issues, namely "oscillation" of particles (as each particle is attracted by two vectors, i.e., personal best and global best vectors, it might keep oscillating between

the two) (Parsopoulos and Vrahatis, 2004) and "two steps forward, one step back"(see section 4.2 for details on this issue) were investigated in (Zhan et al., 2011)[25]. To address these two issues it was proposed to combine global and personal best and use the combined vector in the velocity update rule instead. In order to combine these two vectors, the "Orthogonal Experimental Design", OED, approach was applied (Montgomery et al., 1984) (the new PSO variant was called "Orthogonal Learning Particle Swarm Optimization", OLPSO). The combined position was then used in the velocity update rule, exactly the same as the one in CLPSO (see Eq. 16) where $\vec{P}_i^t$ was replaced by the vector generated by OED approach. This strategy was tested on SPSO and LPSO where results on 16 test cases showed significant improvement (based on the t-test) on the performance of these algorithms when OED is used. However, the selected benchmark had a bias towards the test functions that their optimal solutions are at the center of the coordinate (11 over 16). Given that particles positions in SPSO and LPSO bias towards the coordinate axes Spears et al. (2010), it would be better to test the performance of the algorithm on problems that their solutions are not on the coordinate axes.

Different studies on PSO have been proposed to improve different components of the algorithm (e.g., a better topology and adaptive coefficients). Thus, it might be beneficial if best strategies for updating each of the components are taken from previous studies and combined to generate a new PSO variant. This was done in (Montes de Oca et al., 2009) and the method was called "Frankensteins Particle Swarm Optimization", FPSO. The main aim of the method was to combine different PSO variants together and create a new method that enables them to overcome each other's deficiencies. FPSO components were taken from: the inertia weight setting taken from DIPSO, the velocity was updated by FIPS formulation, acceleration coefficients and setting of V_{max} were set by the method used in HPSO-TVAC, and the topology was set by the method proposed in (Janson and Middendorf, 2005). Experimental results showed that FPSO is effective and outperforms the methods it has been composed of. Also, the same idea was tested in (Tang et al., 2011) and a variant called "Feedback Learning Particle Swarm Optimization", FLSPO, was proposed. FLPSO used DIPSO for the inertia weight, an adaptive approach for φ_1 and φ_2 (similar to HPSO-TVAC with some modification), an adaptive strategy to use personal best or global best vector in the velocity update rule, and a mutation operator applied to a randomly selected dimension of the global best vector. As such mutation operator prevents stagnation, it is very likely that FLPSO is locally convergent. However, the algorithm is not rotationally invariant as it ignores correlation between dimensions. This can be also observed from the reported experimental results in that paper.

As there are many different position/velocity update rules and each of them contain their pros and cons in different situations, it would be beneficial to detect which update rule is more beneficial at the current situation (iteration) and use that strategy for further iterations. As an example, a self-adaptive method was proposed in which the most effective PSO variant was selected during the run for the problem at hand (Wang et al., 2011). This variant was called "Self-Adaptive Learning Particle Swarm Optimization", SALPSO, where four velocity update rules (taken from different PSO variants) were considered and the best update rule was selected to be used in each iteration. As each velocity update rule had its own capabilities in different situations (e.g., exploration, exploitation, multi modal, or unimodal search spaces), it was expected that the overall performance of the algorithm is improved. For each particle and every G

[25]Note that these two issues were not considered in section 3 as they have not been properly supported from theoretical point of view in the related articles.

iterations (a constant set to 10 experimentally), the probability of using each updating rule was updated based on the improvement it offered during the last G iterations. The velocity update rules were selected for each particle at each iteration to update the particles according to these probabilities. Experiments (based on average and standard deviation) showed that SLPSO is effective to deal with both unimodal and multimodal problems. A very similar idea was also presented by Changhe et al. (2012) where the method was called "Self-Learning Particle Swarm Optimization", SLPSO. The only difference between SLPSO and SALPSO was in the implementation details such as how to update probabilities and which velocity update rules are used. However, all used velocity update rules in both methods were rotationally variant, that makes these methods also rotationally variant. In addition, there was no strategy in these methods to deal with stagnation that shows these methods suffer from the local convergence issue.

The ideas used in SALPSO and SLPSO triggered the emergence of another PSO variant called "Multiple Adaptive Methods for Particle Swarm Optimization", PSO-MAM (Hu et al., 2012). In PSO-MAM, all particles were updated by the update rules in SPSO except the global best particle. To update the global best particle, two approaches, a mutation and a gradient descent approach, were designed and one of them was selected randomly (based on a probability distribution that was updated according to the current performance) at each iteration. An extensive experiment was conducted (31 functions and comparison with many different variants of PSO such as CPSO and CLPSO) that showed PSO-MAM is comparable with other PSO variants. It seems that the local convergence issue does not exist in this variant because of the mutation operators. However, the rotation invariance issue does exist in the algorithm that can be observed by comparing its results in that article when the search space has been rotated. The idea of hybridization of SPSO with gradient descent method was also used in (Noel, 2012).

There are some other high quality articles related to the hybridization of PSO such as (Muller et al., 2009), where SPSO was combined with CMA-ES, however, these articles have been excluded from this review because of the selection criteria defined in section 2.

Summary: It can be beneficial to combine different methods in such a way that they overcome each others shortcomings. For example, each particle in SPSO uses information from the global best vector together with its own personal best vector for updating its velocity. However, it is possible to first combine personal best vector with global best vector and then use this new vector as an attractor for the particle. One candidate approach for this combination procedure is OED, which was used in OLPSO (Zhan et al., 2011). It might be valuable to investigate other approaches for combining these vectors (e.g., crossover) and determine the best combination approach in this line. Also, implementing several algorithms and selecting the best algorithm during the run (using some adaptive selection probability) can improve the performance. Many different models can be considered for this selection. As an example, in SLPSO (Wang et al., 2011), the methods were selected randomly while the probability of selection was updated during the run. One potential direction related to this kind of hybrid methods is to investigate the selection method to select appropriate variant during the run. An example can be the usage of a method to recognize the landscape characteristics (rugged, smooth, etc, see (Malan and Engelbrecht, 2009) for example) and use the most appropriate update rule accordingly. Another approach for hybridization is to design one method while taking its different components (e.g., topology adaptation and coefficients adaptation) from different PSO variants, example

is FPSO (Montes de Oca et al., 2009). One of the challenges in these methods is that we need to first determine the best candidate to be used in each component.

5 Modification of PSO to deal with COPs

A single objective COP is defined by

$$\text{find } x \in \mathcal{F} \subseteq S \subseteq R^d \text{ such that } \forall y \in \mathcal{F}, f(x) \leq f(y) \tag{18}$$

where $\mathcal{F} = \{\vec{y} : g_j(\vec{y}) \leq 0 \text{ and } h_k(\vec{y}) = 0 \text{ for all } \vec{y}, j \in \{1, ..., p\}, k \in \{1, ..., q\}\}$, g_j are inequality constraints and h_k are equality constraints, both of them are $R^d \rightarrow R$ for all j and k, and S is defined the same as the one for UOPs (see Eq. 1).

Algorithms that are applied to COPs usually contain two parts: an optimizer and a constraint handling method. The responsibility of the optimizer is to generate new solutions that are potentially better than what was generated before. The constraint handling method is responsible to compare two solutions and to determine which solution is better than the other. Although such comparison is very straightforward in UOPs (two solutions can be compared according to their objective value), it is not that easy to compare solutions when it comes to COPs. The reason is that comparing two infeasible solutions or a feasible and an infeasible solution depends on many parameters such as the shape of the objective function and constraints. Some constraint handling methods involve modification of the solutions so that they become comparable. As an example, in repair (a constraint handling method, see Michalewicz and Schoenauer (1996)), solutions are first modified according to a predefined procedure to make them feasible and then the feasible solutions are compared based on their objective values. Bonyadi and Michalewicz (2014b) provided for a more detailed discussion on the complexity of comparisons in COPs.

There have been a few attempts to extend PSO to use constraint handling methods to deal with COPs. One of the first studies in this area was conducted in (Parsopoulos and Vrahatis, 2002a) where a penalty function was incorporated into SPSO to deal with constraints. The method was applied to several benchmark functions of COP and its results were compared with a variant of ES introduced in (Yang et al., 1997). Results showed that SPSO outperforms ES in many benchmark COPs. This result encouraged researchers to test other PSO variants with different constraint handling methods, e.g., preserving feasibility of solutions (Hu and Eberhart, 2002), closeness to the feasible region (based on the constraint violation value) (Pulido and Coello, 2004), and penalty functions (Coath and Halgamuge, 2003), on more complex COPs.

LPSO was modified (Paquet and Engelbrecht, 2007) and applied to COPs with linear equality constraints (recall that LPSO was a variant of PSO in which the random matrices, R_{1t} and R_{2t}, were replaced by random values, r_{1t} and r_{2t}). Linear equality constraints are formulated as $A\vec{x} = \vec{b}$ where A is a m by d matrix and $\vec{x}$ and $\vec{b}$ are two d dimensional vectors. It was proven that if all personal best and velocities of particles are initialized such that $Ap_i^0 = b$ and $Av_i^0 = 0$ then all further generated solutions by LPSO are feasible. Hence, LPSO is ideal to search through linear equality constraints. Although results showed that LPSO performs well in dealing with linear equality constraints, the algorithm suffered from the line search and stagnation issues. To overcome these issues, the velocity update rule for the global best vector was modified and another type of LPSO called "Guaranteed Converging Linear Particle Swarm Optimization", GCLPSO, was proposed (Paquet and Engelbrecht, 2003, 2007). The idea was similar to what was discussed for GCPSO with some modifications to guarantee

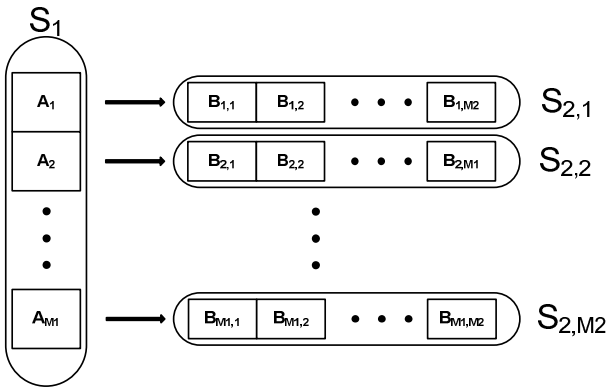


Figure 7: A_i is the ith particle of the swarm S_1, which provides penalty factors for swarm $S_{2,i}$. M1 is the size of the swarm type one (S_1) and M2 is the size of each swarm of type two (each $S_{2,i}$).

feasibility after mutation. Experiments with a small number of benchmark functions showed that GCLPSO on some benchmark problems were comparable with that of Genocop II (Michalewicz and Janikow, 1996) (a GA-based optimization approach for COPs).

In 2007 a "Co-evolutionary Particle Swarm Optimization for Constraint Optimization Problems", CPSO-COP, approach was proposed (He and Wang, 2007). SPSO was used as an optimizer in CPSO-COP while the constraint handling method was somewhat new. Two swarms were considered in CPSO-COP, one to store penalty coefficients (swarm S_1 with the size n_1) and the other to store solutions (swarm S_2). Each particle in S_1 provided penalty coefficients to evaluate particles in S_2. S_2 contained n_1 sub-swarms, the sub-swarm i was evaluated based on the coefficients provided by the particle i in S_1 (see Fig. 7). At each generation of co-evolution process, every $S_{2,i}$ (sub-swarm i of the S_2) was evolved using SPSO for a certain number of generations and it used particle i from S_1 for evaluation. After all $S_{2,i}$ were evolved, S_1 was evolved using SPSO for one iteration. This process was re-run until a termination criterion was met. The authors compared their results with GA-based optimization methods for solving several engineering COP benchmarks. Experimental results showed that CPSO-COP converges to better solutions (in terms of the mean, min, max, standard deviation of the results) in less number of iterations in comparison to GA (Coello Coello, 2000).

A "Cooperation Comprehensive Learning Particle Swarm Optimization", Co-CLPSO, was implemented and applied to COPs (Liang et al., 2010). The comprehensive learning approach in PSO (e.g., see CLPSO description in section 4.2) was used as an optimization method to solve COPs. Also, the idea of the constraint handling method in that paper was to assign particles adaptively to either satisfy different constraints or optimize the objective value. A local search based on "Sequential Quadratic Programming", SQP, was used to improve the solutions during the run. This approach received the fourth place in the competition of CEC2010 in solving COPs (Mallipeddi and Suganthan, 2010). However, no results were reported for the algorithm without the usage of the local search SQP. Hence, it is not clear whether it is the local search that

is responsible for the good performance of the algorithm or other components of the algorithm are the reasons behind such good results.

A PSO variant was proposed by Sun et al. (2011) in which a feasibility preserving constraint handling method was used. In that method, it was suggested to update the position of the particles by $x^i_{t+1} = x^i_t + Mv^i_{t+1}$ where M is a $d \times d$ diagonal matrix where each element on the diagonal of M belongs to [0, 1]. If x^i_{t+1} for $M = I$ (I is the identity matrix) was feasible then the new position is accepted, otherwise, another SPSO instance is run to find a matrix M in a way that x^i_{t+1} is a high quality point (feasible and minimizes the objective value). This means that the latter instance of SPSO searches through a hyper-cube where x^i_t is a vertex of it and v^i_{t+1} is its diagonal. The comparison results (based on best, worst, and average of the found solutions on a standard benchmark of COPs with 13 test cases) showed that this method is comparable with other PSO variants.

A multi-start PSO was proposed (Bonyadi et al., 2013) where a mutation operator was added to LPSO to address the stagnation, line search, and swarm size issues (this variant was called "Mutation Linear Particle Swarm Optimization", MLPSO). MLPSO was then used as an optimizer to solve COPs while a simple approach was used to compare solutions. It was assumed that any feasible solution is better than any infeasible solution, feasible solutions are compared based on their objective value, and infeasible solutions are compared based on their constraint violation value (calculated by $\sum_{i=1}^{p} max\{0, g_i(x)\} + \sum_{i=1}^{q} max\{0, h_i(x)\}$). This constraint handling method was indeed a simplified version of "Epsilon Level Constraint Handling", ELCH (Takahama and Sakai, 2005). Also, a method based on the CMA-ES was proposed that used the same technique to handle constraints. Experiments showed that MLPSO has a better performance in finding feasible solutions while CMA-ES performs better in optimizing the objective function. Thus, a hybrid method was proposed that run PSO to find the first feasible solution and then that solution was improved by CMA-ES. However, to prevent MLPSO from finding feasible solutions in a poor-quality feasible region, a multi-start strategy was incorporated in the algorithm so that several instances of MLPSO were run after each other to generate different feasible solutions. This idea potentially leads to locating disjoint feasible regions that increases the possibility of finding the region which includes the optimal solution. Then the best among those solutions were fed into CMA-ES for further improvement. MLPSO was further enhanced in (Bonyadi et al., 2014a) with a time-adaptive topology rather than a multi-swarm strategy. Also, the idea of locating disjoint feasible regions in a COP was further discussed in (Bonyadi and Michalewicz, 2014b) where the importance of the topic was emphasized and challenges were reviewed.

A PSO algorithm for solving COPs called "Automatic Particle Injection PSO", API-PSO, was proposed (Elsayed et al., 2013) in which the balance between exploration and exploitation was controlled during the run by using LPSO or SPSO in each iteration. The choice between the two methods was made based on a random value, which favored SPSO at the beginning and LPSO at the end of the run. The idea was to emphasize exploration in the earlier and exploitation in the latter iterations. Also, a mutation operator and an automatic injection of new particles were added to prevent the algorithm to get stuck in local optima. Further, the inertia weight and acceleration coefficients were considered as variables and they were fed into the position of particles ($\vec{x}$) so that they are set during the run by the algorithm (self-adaptive). In order to handle constraints, API-PSO was extended by ELCH. Results (based on the average values) showed that the algorithm is comparable with other methods to deal with COPs.

In many real-world COPs it is highly probable that some constraints are active at optimum points (Schoenauer and Michalewicz, 1996), i.e., some optimum points are on the boundary of feasible area of the search space. The reason is that constraints in real-world problems often represent limitations of resources, such as rate, number of resource, etc. Thus, it is usually beneficial to make use of some resources as much as possible, which means that some constraints are active at high quality solutions. Such resources that limit the optimization algorithm to achieve better solutions (i.e., forms the feasible space) are called bottlenecks Bonyadi et al. (2014c). Bottlenecks are usually important to be known in a system because the best investment from companies should be concentrated on the bottlenecks to maximize profit (see Bonyadi et al. (2014c); Chatterjee and Mukherjee (2006) for a complete discussion on bottlenecks and investment and their relations to optimization and constraints). Presence of active constraints at the optimum points causes difficulty for many optimization algorithms to locate optimal solutions (Schoenauer and Michalewicz, 1997). In order to address this issue, it was proposed to define a function (called "Constraints Boundary Narrower", CBN) that maps all feasible regions of a COP to the edges of feasibility of that COP so that search methods only focus on the boundaries of feasible and infeasible areas rather than all feasible regions (Bonyadi and Michalewicz, 2014c). In that study SPSO was used as an optimizer and CBN (together with some of its derivatives) were tested on COPs that their optimum solution were on the boundary of feasible area. Results showed that this combination is effective in finding optimal solutions when they are close to the edges of feasible area.

Summary: There have been a few attempts to extend PSO to deal with COPs, such as the usage of penalty functions (Parsopoulos and Vrahatis, 2002a; Takahama and Sakai, 2005) or preservation of feasibility (Coath and Halgamuge, 2003). However, there have not been many attempts to extend PSO to search the edge of feasible region, which can be considered as future research direction. This idea has been used in other optimization methods to improve their performance to find high quality solutions for COPs (e.g. (Leguizamon and Coello Coello, 2009)). In addition to the edges of feasibility, another potential area of research is to extend the algorithm so that it can deal with dynamic COPs (Nguyen and Yao, 2012). As constraints formulate limitations of resources and these limitations might change in time because of random events in real-world, dynamic constraints are found in real-world optimization problems. Further, the importance of locating disjoint feasible areas has been discussed in (Bonyadi et al., 2014a; Bonyadi and Michalewicz, 2014b; Smith et al., 2013). Indeed, feasible areas in COPs might be of irregular shapes or might be disjoint. Ability to locate these feasible areas might be quite important in many applications, which introduces another area of research for PSO as well as other evolutionary methods. Another potential area of research is to investigate the performance of PSO variants to deal with COPs through theoretical analysis, e.g., first hitting time, convergence, and stability. This track of research started on 1981 (Baba, 1981), but to the best of our knowledge, remained untouched for optimization algorithms in the field of evolutionary computation.

6 Conclusions

New advances in the PSO algorithm, including convergence properties, transformation invariance, modification of components, coefficients adaptation, population sizing, topology, hybridization, and dealing with constraint optimization problems, were reviewed in this paper. For each of these topics, several articles were cited and, at the end of each sub-section, a summary of the reviewed articles was given where the potential

challenges and research directions related to that area were discussed.

The first part of this article was concentrated on limitation of PSO that have been theoretically investigated in the past years. These limitations were categorized into two main groups: convergence and transformation invariance. For convergence properties of PSO it was pointed out that analyzing different behavior of particles before convergence and convergence analysis (including convergence to a fixed point, local convergence, and stagnation) of other PSO variants (e.g., SPSO2011 and RPSO) constitute potential research directions (sections 3.1 and 3.1.2). Also, there have not been many studies on the first hitting time analysis of PSO variants, which is another potential research direction (section 3.1.3). Another important area of study in PSO is transformation invariance property of the algorithm. Indeed, there are not many PSO variants that are transformation invariant, that makes this area of research an open era for new ideas and analyses (section 3.2). Note that, as it has been already mentioned in some articles (Auger et al., 2009; Hansen et al., 2008), these properties (convergence and transformation invariance) are important for an algorithm as, if an algorithm does not contain these properties, it is impossible to scale its good performance on a test set to a wide variety of problems. Moreover, many other derivative-free optimization methods (e.g., ES, CMA-ES, DE) have addressed these issues successfully (Auger et al., 2009).

Apart from theoretical studies, many experimental studies related to modifications of components and parameters of PSO were also reviewed. Topology of the swarm (section 4.1.1), coefficients (section 4.1.2), and population size (section 4.1.3) are parameters of PSO that were investigated in this paper. Also, articles that modified the velocity updating rule of PSO were analyzed in section 4.2. Some other studies that hybridized the algorithm with other optimization methods to improve the overall performance of the designed optimizer were discussed in section 4.3. Finally, applications of PSO to deal with COPs were investigated (section 5) and potentially promising directions in this topic were also summarized. One of the main potential future directions related to these topics was the usage of land scape characteristics to determine the current situation and then decide based on that (e.g., to update topology or to change the velocity update rule). Another potential direction was to investigate the theoretical aspects of the current experimental approaches to see if the good performance reported in those articles can be supported more formally (see summaries of these sections for additional possible directions).

The articles reviewed in this paper were selected from high-ranked conferences and journals, however, most of them (especially those included in the experimental part of this survey) did not provide adequate discussion (from theoretical perspective or even just general discussion) on merits of the proposed approach. Only experimental results were presented. The lack of such discussion becomes more severe as many articles did not include appropriate statistical analyses of the results, did not use standard benchmarks, or they only focused on a small number of test cases, in some articles, those test cases were biased (see section 4.1.2 and 4.1.1 for some examples). This in fact invalidates many claims by the authors and may mislead other researchers, although the ideas themselves were original and potentially beneficial. In addition, in some articles, many ideas have been experimented with and many components have been added/modified–all these may confuse the reader, as it is not clear which component is responsible for a good performance of the algorithm and why (see section 4 and 5 for some examples). Such issue is more visible in hybrid approaches where the resulted method needs to be shown (through proper statistical tests or theoretical analyses) to outperform each of the methods that it is constituted upon.

7 Acknowledgment

The authors would like to extend their great appreciation to Professor James Kennedy who provided detailed comments on earlier versions of this paper. This work was partially funded by the ARC Discovery Grant DP130104395 and by grant N N519 5788038 from the Polish Ministry of Science and Higher Education (MNiSW).

References

Agrafiotis, D. K. and Cedeno, W. (2002). Feature selection for structure-activity correlation using binary particle swarms. *Journal of Medicinal Chemistry*, 45(5):1098–1107.

Alatas, B., Akin, E., and Ozer, A. B. (2009). Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons and Fractals*, 40(4):1715–1734.

Albert, R. and Barabsi, A. L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):1–47.

Angeline, P. J. (1998). Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In *Evolutionary Programming VII*, pages 601–610. Springer.

Arabas, J., Michalewicz, Z., and Mulawka, J. (1994). Gavaps-a genetic algorithm with varying population size. In *World Congress on Computational Intelligence*, pages 73–78. IEEE.

Arumugam, M. S., Rao, M., and Chandramohan, A. (2008). A new and improved version of particle swarm optimization algorithm with globallocal best parameters. *Knowledge and Information systems*, 16(3):331–357.

Arumugam, M. S., Rao, M. V. C., and Tan, A. W. C. (2009). A novel and effective particle swarm optimization like algorithm with extrapolation technique. *Applied Soft Computing*, 9(1):308–320.

Auger, A., Hansen, N., Zerpa, J. P., Ros, R., and Schoenauer, M. (2009). Empirical comparisons of several derivative free optimization algorithms. In *Acte du 9ieme colloque national en calcul des structures (Proceedings of the 9th National Symposium in calculation of structures)*, volume 1.

Auger, A. and Teytaud, O. (2007). Continuous lunches are free! In *Genetic and evolutionary computation Conference*, pages 916–922. ACM.

Baba, N. (1981). Convergence of a random optimization method for constrained optimization problems. *Journal of Optimization Theory and Applications*, 33(4):451–461.

Banks, A., Vincent, J., and Anyakoha, C. (2007). A review of particle swarm optimization. part i: background and development. *Natural Computing*, 6(4):467–484.

Banks, A., Vincent, J., and Anyakoha, C. (2008). A review of particle swarm optimization. part ii: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Natural Computing*, 7(1):109–124.

Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies–a comprehensive introduction. *Natural computing*, 1(1):3–52.

Birbil, . l., Fang, S.-C., and Sheu, R.-L. (2004). On the convergence of a population-based global optimization algorithm. *Journal of global optimization*, 30(2-3):301–318.

Blackwell, T. (2012). A study of collapse in bare bones particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 16(3):354–372.

Bonyadi, M. R., Li, X., and Michalewicz, Z. (2013). A hybrid particle swarm with velocity mutation for constraint optimization problems. In *Genetic and Evolutionary Computation Conference*, pages 1–8. ACM.

Bonyadi, M. R., Li, X., and Michalewicz, Z. (2014a). A hybrid particle swarm with a time-adaptive topology for constrained optimization. *Swarm and Evolutionary Computation*, 18:22–37.

Bonyadi, M. R. and Michalewicz, Z. (2012). A fast particle swarm optimization algorithm for the multidimensional knapsack problem. In *Congress on Evolutionary Computation*, pages 1–8. IEEE.

Bonyadi, M. R. and Michalewicz, Z. (2014a). A locally convergent rotationally invariant particle swarm optimization algorithm. *Swarm Intelligence*, 8(3):159–198.

Bonyadi, M. R. and Michalewicz, Z. (2014b). Locating potentially disjoint feasible regions of a search space with a particle swarm optimizer. In Deb, K. and Datta, R., editors, *Evolutionary Constrained Optimization*, book section 8, pages 205–230. Springer-Verlag.

Bonyadi, M. R. and Michalewicz, Z. (2014c). On the edge of feasibility: a case study of the particle swarm optimizer. In *Congress on Evolutionary Computation*, pages 3059–3066. IEEE.

Bonyadi, M. R. and Michalewicz, Z. (2014d). Spso2011 analysis of stability, local convergence, and rotation sensitivity. In *Genetic and Evolutionary Computation Conference*, pages 9–16. ACM.

Bonyadi, M. R., Michalewicz, Z., and Li, X. (2014b). An analysis of the velocity updating rule of the particle swarm optimization algorithm. *Journal of Heuristics*, 20(4):417–452.

Bonyadi, M. R., Michalewicz, Z., and Wagner, M. (2014c). Beyond the edge of feasibility: analysis of bottlenecks. volume LNCS 8886, pages 431–442. Springer.

Bratton, D. and Kennedy, J. (2007). Defining a standard for particle swarm optimization. In *Swarm Intelligence Symposium*, pages 120–127. IEEE.

Caamaño, P., Prieto, A., Becerra, J. A., Bellas, F., and Duro, R. J. (2010). Real-valued multimodal fitness landscape characterization for evolution. In *Neural Information Processing. Theory and Algorithms*, pages 567–574. Springer.

Campana, E., Fasano, G., and Pinto, A. (2010). Dynamic analysis for the selection of parameters and initial population, in particle swarm optimization. *Journal of Global Optimization*, 48(3):347–397.

Carlisle, A. and Dozier, G. (2001). An off-the-shelf PSO. In *PSO Workshop*.

Chang-Huang, C. and Jia-Shing, S. (2009). Simple particle swarm optimization. In *International Conference on Machine Learning and Cybernetics*, volume 1, pages 460–466.

Changhe, L., Shengxiang, Y., and Trung Thanh, N. (2012). A self-learning particle swarm optimizer for global optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(3):627–646.

Chatterjee, A. and Mukherjee, S. (2006). Unified concept of bottleneck. Technical report, Indian Institute of Management Ahmedabad, Research and Publication Department.

Chatterjee, A. and Siarry, P. (2006). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers and Operations Research*, 33(3):859–871.

Chen, D. B. and Zhao, C. X. (2009). Particle swarm optimization with adaptive population size and its application. *Applied Soft Computing*, 9(1):39–48.

Chen, W. N., Zhang, J., Chung, H. S. H., Zhong, W. L., Wu, W. G., and Shi, Y. H. (2010). A novel set-based particle swarm optimization method for discrete optimization problems. *IEEE Transactions on Evolutionary Computation*, 14(2):278–300.

Cleghorn, C. W. and Engelbrecht, A. P. (2014a). A generalized theoretical deterministic particle swarm model. *Swarm Intelligence*, 8(1):35–59.

Cleghorn, C. W. and Engelbrecht, A. P. (2014b). Particle swarm convergence: An empirical investigation. In *Congress on Evolutionary Computation (CEC)*, pages 2524–2530. IEEE.

Cleghorn, C. W. and Engelbrecht, A. P. (2014c). Particle swarm convergence: Standardized analysis and topological influence. In *Swarm Intelligence Symposium*, pages 134–145. Springer.

Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Congress on Evolutionary Computation*, volume 3. IEEE.

Clerc, M. (2006). *Particle swarm optimization*. Wiley-ISTE.

Clerc, M. (2011). Standard particle swarm optimisation.

Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73.

Coath, G. and Halgamuge, S. K. (2003). A comparison of constraint-handling methods for the application of particle swarm optimization to constrained nonlinear optimization problems. In *Congress on Evolutionary Computation*, volume 4, pages 2419–2425 Vol.4. IEEE.

Coello Coello, C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41(2):113–127.

Cooren, Y., Clerc, M., and Siarry, P. (2009). Performance evaluation of tribes, an adaptive particle swarm optimization algorithm. *Swarm Intelligence*, 3(2):149–178.

Črepinšek, M., Liu, S.-H., and Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: a survey. *ACM Computing Surveys (CSUR)*, 45(3):35.

Dasgupta, S., Das, S., Biswas, A., and Abraham, A. (2009). On stability and convergence of the population-dynamics in differential evolution. *Ai Communications*, 22(1):1–20.

Deng, J.-L. (1989). Introduction to grey system theory. *The Journal of grey system*, 1(1):1–24.

Derrac, J., García, S., Molina, D., and Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18.

Dorea, C. (1983). Expected number of steps of a random optimization method. *Journal of Optimization Theory and Applications*, 39(2):165–171.

Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *International Symposium on Micro Machine and Human Science*, pages 39–43. IEEE.

Eberhart, R. and Shi, Y. (2001). Tracking and optimizing dynamic systems with particle swarms. In *Congress on Evolutionary Computation*, volume 1, pages 94–97. IEEE.

Eberhart, R. C. and Shi, Y. (1998). Evolving artificial neural networks. In *Proceedings of the International Conference on Neural Networks and Brain*, pages 84–89.

Eberhart, R. C. and Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Congress on Evolutionary Computation*, volume 1, pages 84–88. IEEE.

Eiben, A. E., Hinterding, R., and Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141.

Elsayed, S. M., Sarker, R. A., and Essam, D. L. (2012). Memetic multi-topology particle swarm optimizer for constrained optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE.

Elsayed, S. M., Sarker, R. A., and Mezura-Montes, E. (2013). Particle swarm optimizer for constrained optimization. In *Congress on Evolutionary Computation*, pages 2703–2711. IEEE.

Emara, H. M. (2009). Adaptive clubs-based particle swarm optimization. In *American Control Conference*, pages 5628–5634. IEEE.

Engelbrecht, A. (2005). *Fundamentals of computational swarm intelligence*. Wiley, Hoboken, NJ.

Engelbrecht, A. (2012). Particle swarm optimization: Velocity initialization. In *Congress on Evolutionary Computation*, pages 1–8. IEEE.

Engelbrecht, A. (2013). Particle swarm optimization: Global best or local best. In *Congress on Computational Intelligence*, pages 124–135. IEEE.

Everitt, T., Lattimore, T., and Hutter, M. (2014). Free lunch for optimisation under the universal distribution. In *Congress on Evolutionary Computation*, pages 167–174. IEEE.

Garca-Gonzalo, E. and Fernndez-Martnez, J. L. (2014). Convergence and stochastic stability analysis of particle swarm optimization variants with generic parameter distributions. *Applied Mathematics and Computation*, 249(0):286–302.

Ghosh, S., Das, S., Kundu, D., Suresh, K., Panigrahi, B. K., and Cui, Z. (2010). An inertia-adaptive particle swarm system with particle mobility factor for improved global optimization. *Neural Computing and Applications*, 21(2):237–250.

Gong, Y. j. and Zhang, J. (2013). Small-world particle swarm optimization with topology adaptation. In *Conference on Genetic and evolutionary computation*, pages 25–32. ACM.

Hansen, N. (2000). Invariance, self-adaptation and correlated mutations in evolution strategies. In *Parallel Problem Solving from Nature PPSN VI*, pages 355–364. Springer.

Hansen, N. (2006). The cma evolution strategy: a comparing review. *Towards a new evolutionary computation*, pages 75–102.

Hansen, N. and Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Congress on Evolutionary Computation*, pages 312–317. IEEE.

Hansen, N., Ros, R., Mauny, N., Schoenauer, M., and Auger, A. (2011). Impacts of invariance in search: When cma-es and pso face ill-conditioned and non-separable problems. *Applied Soft Computing*, 11(8):5755–5769.

Hansen, N., Ros, R., Mauny, N., Schoenausr, M., and Auger, A. (2008). Pso facing non-separable and ill-conditioned problems. Technical Report 6447, INRIA, France.

He, J. and Yao, X. (2002). From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5):495–511.

He, Q. and Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, 20(1):89–99.

Helwig, S., Branke, J., and Mostaghim, S. (2013). Experimental analysis of bound handling techniques in particle swarm ptimization. *IEEE Trans. Evolutionary Computation*, 17(2):259–271.

Helwig, S. and Wanka, R. (2007). Particle swarm optimization in high-dimensional bounded search spaces. In *Swarm Intelligence Symposium*, pages 198–205. IEEE.

Helwig, S. and Wanka, R. (2008). Theoretical analysis of initial particle swarm behavior. In *International Conference on Parallel Problem Solving from Nature*, pages 889–898. Springer-Verlag Berlin Heidelberg.

Hsieh, S. T., Sun, T. Y., Liu, C. C., and Tsai, S. J. (2009). Efficient population utilization strategy for particle swarm optimizer. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 39(2):444–456.

Hu, M., Wu, T., and Weir, J. D. (2012). An intelligent augmentation of particle swarm optimization with multiple adaptive methods. *Information Sciences*, 213:68–83.

Hu, X. and Eberhart, R. (2002). Solving constrained nonlinear optimization problems with particle swarm optimization. In *World multiconference on systemics, cybernetics and informatics*, volume 5, page 203206. Citeseer.

Hu, X., Shi, Y., and Eberhart, R. (2004). Recent advances in particle swarm. In *Congress on Evolutionary Computation*, volume 1, pages 90–97 Vol. 1. Ieee.

Huang, H., Qin, H., Hao, Z., and Lim, A. (2010). Example-based learning particle swarm optimization for continuous optimization. *Information Sciences*, 182(1):125–138.

Jägersküpper, J. (2008). Lower bounds for randomized direct search with isotropic sampling. *Operations research letters*, 36(3):327–332.

Janson, S. and Middendorf, M. (2005). A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(6):1272–1282.

Jiang, M., Luo, Y., and Yang, S. (2007a). Particle swarm optimization-stochastic trajectory analysis and parameter selection. In Chan, F. and KumarTiwari, M., editors, *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, book section 11, pages 179–198. I-TECH Education and Publishing, Wien.

Jiang, M., Luo, Y. P., and Yang, S. Y. (2007b). Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters*, 102(1):8–16.

Kao, Y.-T. and Zahara, E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing*, 8(2):849–857.

Kaucic, M. (2013). A multi-start opposition-based particle swarm optimization algorithm with adaptive velocity for bound constrained global optimization. *Journal of Global Optimization*, 55(1):165–188.

Kelley, C. T. (1999). *Iterative methods for optimization*, volume 18. Siam.

Kennedy, J. (1999). Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Congress on Evolutionary Computation*, volume 3. IEEE.

Kennedy, J. (2003). Bare bones particle swarms. In *Swarm Intelligence Symposium*, pages 80–87.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE.

Kennedy, J. and Mendes, R. (2002). Population structure and particle swarm performance. In *Congress on Evolutionary Computation*, volume 2, pages 1671–1676. IEEE.

Krink, T., VesterstrOm, J. S., and Riget, J. (2002). Particle swarm optimisation with spatial particle extension. In *Congress on Evolutionary Computation*, volume 2, pages 1474–1479. IEEE.

Leguizamon, G. and Coello Coello, C. A. (2009). Boundary search for constrained numerical optimization problems with an algorithm inspired by the ant colony metaphor. *IEEE Transactions on Evolutionary Computation*, 13(2):350–368.

Lehre, P. K. and Witt, C. (2013). Finite first hitting time versus stochastic convergence in particle swarm optimisation. In *Advances in Metaheuristics*, pages 1–20. Springer.

Leu, M.-S. and Yeh, M.-F. (2012). Grey particle swarm optimization. *Applied Soft Computing*, 12(9):2985–2996.

Liang, J., Zhigang, S., and Zhihui, L. (2010). Coevolutionary comprehensive learning particle swarm optimizer. In *Congress on Evolutionary Computation*, pages 1–8. IEEE.

Liang, J. J., Qin, A. K., Suganthan, P. N., and Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3):281–295.

Liu, Q. (2014). Order-2 stability analysis of particle swarm optimization. *Evolutionary Computation*, pages 1–30.

Mahor, A., Prasad, V., and Rangnekar, S. (2009). Economic dispatch using particle swarm optimization: A review. *Renewable and sustainable energy reviews*, 13(8):2134–2141.

Malan, K. and Engelbrecht, A. (2009). Quantifying ruggedness of continuous landscapes using entropy. In *Congress on Evolutionary Computation*, pages 1440–1447. IEEE.

Mallipeddi, R. and Suganthan, P. (2010). Problem definitions and evaluation criteria for the cec 2010 competition on constrained real-parameter optimization. *Singapore: Nanyang Technological University. Technical report*.

Matyas, J. (1965). Random optimization. *Automation and Remote Control*, 26(2):246–253.

Mendes, R. (2004). *Population topologies and their influence in particle swarm performance*. PhD thesis.

Mendes, R., Kennedy, J., and Neves, J. (2004). The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 8(3):204–210.

Michalewicz, Z. (1995). A survey of constraint handling techniques in evolutionary computation methods. *Evolutionary Programming IV*, pages 135–155.

Michalewicz, Z. and Janikow, C. (1996). Genocop: a genetic algorithm for numerical optimization problems with linear constraints. *Communications of the ACM*, 39(12es):175.

Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1):1–32.

Mondes de Oca, M. A., Stützle, T., Van den Enden, K., and Dorigo, M. (2011). Incremental social learning in particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(2):368–384.

Montes de Oca, M. A. and Stützle, T. (2008). Towards incremental social learning in optimization and multiagent systems. In *Genetic and evolutionary computation conference*, pages 1939–1944. ACM.

Montes de Oca, M. A., Stützle, T., Birattari, M., and Dorigo, M. (2009). Frankenstein's pso: A composite particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 13(5):1120–1132.

Montgomery, D. C., Montgomery, D. C., and Montgomery, D. C. (1984). *Design and analysis of experiments*, volume 7. Wiley New York.

Muller, C., Baumgartner, B., and Sbalzarini, I. F. (2009). Particle swarm cma evolution strategy for the optimization of multi-funnel landscapes. In *Congress on Evolutionary Computation*, pages 2685–2692. IEEE.

Nasir, M., Das, S., Maity, D., Sengupta, S., Halder, U., and Suganthan, P. N. (2012). A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. *Information Sciences*, 209:16–36.

Newman, M., Barabasi, A.-L., and Watts, D. J. (2006). *The structure and dynamics of networks*. Princeton University Press.

Nguyen, T. and Yao, X. (2012). Continuous dynamic constrained optimisation-the challenges. *IEEE Transactions on Evolutionary Computation*, 16(6):769–786.

Nickabadi, A., Ebadzadeh, M. M., and Safabakhsh, R. (2011). A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing*, 11(4):3658–3670.

Noel, M. M. (2012). A new gradient based particle swarm optimization algorithm for accurate computation of global minimum. *Applied Soft Computing*, 12(1):353–359.

Omidvar, M., Li, X., Mei, Y., and Yao, X. (2014). Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):378–393.

Ozcan, E. and Mohan, C. K. (1999). Particle swarm optimization: surfing the waves. In *Congress on Evolutionary Computation*, volume 3. IEEE.

Paquet, U. and Engelbrecht, A. (2003). A new particle swarm optimiser for linearly constrained optimisation. In *Congress on Evolutionary Computation*, volume 1, pages 227–233. IEEE.

Paquet, U. and Engelbrecht, A. (2007). Particle swarms for linearly constrained optimisation. *Fundamenta Informaticae*, 76(1):147–170.

Parsopoulos, K. and Vrahatis, M. (2002a). Particle swarm optimization method for constrained optimization problems. *Intelligent TechnologiesTheory and Application: New Trends in Intelligent Technologies*, 76:214–220.

Parsopoulos, K. and Vrahatis, M. (2002b). Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1(2):235–306.

Parsopoulos, K. E. and Vrahatis, M. N. (2004). On the computation of all global minimizers through particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):211–224.

Poli, R. (2008). Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artifficial Evolution and Application*, 2008(3):1–10.

Poli, R. (2009). Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *IEEE Transactions on Evolutionary Computation*, 13(4):712–721.

Poli, R., Brattonx, D., Blackwell, T., and Kennedy, J. (2007a). Theoretical derivation, analysis and empirical evaluation of a simpler particle swarm optimiser. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 1955–1962.

Poli, R., Kennedy, J., and Blackwell, T. (2007b). Particle swarm optimization: an overview. *Swarm Intelligence*, 1(1):33–57.

PSO (2006). Pso source code (available online at pso info).

Pulido, G. T. and Coello, C. A. C. (2004). A constraint-handling mechanism for particle swarm optimization. In *Congress on Evolutionary Computation*, volume 2, pages 1396–1403. IEEE.

Ratnaweera, A., Halgamuge, S. K., and Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 8(3):240–255.

Rudolph, G. (1997). Local convergence rates of simple evolutionary algorithms with cauchy mutations. *IEEE Transactions on Evolutionary Computation*, 1(4):249–258.

Rudolph, G. (2013). Stochastic convergence. In G. Rozenberg, T. B. and Kok, J., editors, *Handbook of Natural Computing*, chapter 27, pages 847–869. Springer.

Salomon, R. (1995). Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions - a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39:263–278.

Schmitt, M. and Wanka, R. (2013). Particle swarm optimization almost surely finds local optima. In *Genetic and Evolutionary Computation Conference*, pages 1629–1636. ACM.

Schoenauer, M. and Michalewicz, Z. (1996). Evolutionary computation at the edge of feasibility. *Parallel Problem Solving from NaturePPSN IV*, pages 245–254.

Schoenauer, M. and Michalewicz, Z. (1997). Boundary operators for constrained parameter optimization problems. In *ICGA*, pages 322–329.

Shi, Y. and Eberhart, R. (1998a). A modified particle swarm optimizer. In *World Congress on Computational Intelligence*, pages 69–73. IEEE.

Shi, Y. and Eberhart, R. (1998b). Parameter selection in particle swarm optimization. In *Evolutionary Programming VII*, pages 591–600. Springer Berlin Heidelberg.

Shi, Y. and Eberhart, R. C. (2001). Fuzzy adaptive particle swarm optimization. In *Congress on Evolutionary Computation*, volume 1, pages 101–106. IEEE.

Smith, L., Chinneck, J., and Aitken, V. (2013). Constraint consensus concentration for identifying disjoint feasible regions in nonlinear programmes. *Optimization Methods and Software*, 28(2):339–363.

Solis, F. J. and Wets, R. J.-B. (1981). Minimization by random search techniques. *Mathematics of Operations Research*, 6(1):19–30.

Song, M.-P. and Gu, G.-C. (2004). Research on particle swarm optimization: a review. In *International Conference on Machine Learning and Cybernetics*, volume 4, pages 2236–2241. IEEE.

Spears, W., Green, D., and Spears, D. (2010). Biases in particle swarm optimization. *International Journal of Swarm Intelligence Research*, 1(2):34–57.

Storn, R. and Price, K. (1997). Differential evolutiona simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359.

Sun, C.-l., Zeng, J.-c., and Pan, J.-s. (2011). An improved vector particle swarm optimization for constrained optimization problems. *Information Sciences*, 181(6):1153–1163.

Takahama, T. and Sakai, S. (2005). Constrained optimization by constrained particle swarm optimizer with -level control. *Soft Computing as Transdisciplinary Science and Technology*, pages 1019–1029.

Tang, Y., Wang, Z., and Fang, J.-a. (2011). Feedback learning particle swarm optimization. *Applied Soft Computing*, 11(8):4713–4725.

Trelea, I. C. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, 85(6):317–325.

Van Den Bergh, F. (2002). *An analysis of particle swarm optimizers*. PhD thesis.

Van den Bergh, F. and Engelbrecht, A. (2002). A new locally convergent particle swarm optimiser. In *Systems, man and cybernetics*, volume 3, pages 96–101. IEEE.

Van den Bergh, F. and Engelbrecht, A. (2004). A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):225–239.

Van den Bergh, F. and Engelbrecht, A. (2010). A convergence proof for the particle swarm optimiser. *Fundamenta Informaticae*, 105(4):341–374.

Van den Bergh, F. and Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information Sciences*, 176(8):937–971.

Voss, M. S. (2005). Principal component particle swarm optimization: a step towards topological swarm intelligence. In *Congress on Evolutionary Computation*, volume 1, pages 298–305.

Vrahatis, M., Androulakis, G., and Manoussakis, G. (1996). A new unconstrained optimization method for imprecise function and gradient values. *Journal of Mathematical Analysis and Applications*, 197(2):586–607.

Wang, H., Sun, H., Li, C., Rahnamayan, S., and Pan, J.-s. (2013). Diversity enhanced particle swarm optimization with neighborhood search. *Information Sciences*, 223:119–135.

Wang, Y., Li, B., Weise, T., Wang, J., Yuan, B., and Tian, Q. (2011). Self-adaptive learning based particle swarm optimization. *Information Sciences*, 181(20):4515–4538.

Wilke, D. (2005). *Analysis of the particle swarm optimization algorithm*. PhD thesis.

Wilke, D., Kok, S., and Groenwold, A. (2007a). Comparison of linear and classical velocity update rules in particle swarm optimization: notes on diversity. *International journal for numerical methods in engineering*, 70(8):962–984.

Wilke, D., Kok, S., and Groenwold, A. (2007b). Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on scale and frame invariance. *International journal for numerical methods in engineering*, 70(8):985–1008.

Witt, C. (2009). Why standard particle swarm optimisers elude a theoretical runtime analysis. In *Foundations of genetic algorithms*, pages 13–20. ACM.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Xinchao, Z. (2010). A perturbed particle swarm algorithm for numerical optimization. *Applied Soft Computing*, 10(1):119–124.

Yang, J., Chen, Y., Horng, J., and Kao, C. (1997). Applying family competition to evolution strategies for constrained optimization. In *Evolutionary Programming VI*, volume 1213, pages 201–211. Springer.

Zambrano-Bigiarini, M., Clerc, M., and Rojas, R. (2013). Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In *Congress on Evolutionary Computation*, pages 2337–2344. IEEE.

Zhan, Z., Zhang, J., Li, Y., and Chung, H. (2009). Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1362–1381.

Zhan, Z.-H., Zhang, J., Li, Y., and Shi, Y.-h. (2011). Orthogonal learning particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 15(6):832–847.

Zhang, C. and Yi, Z. (2011). Scale-free fully informed particle swarm optimization algorithm. *Information Sciences*, 181(20):4550–4568.

Zheng, Y., Ma, L., Zhang, L., and Qian, J. (2003). Empirical study of particle swarm optimizer with an increasing inertia weight. In *Congress on Evolutionary Computation*, pages 221–226. IEEE.

3 Theoretical analysis and modifications of PSO

This chapter includes 3 subsections; each of these subsections contains one paper related to theoretical analysis of PSO.

3.1 Influence of rotation matrices on the performance of PSO variants

The article included in this subsection studies application of rotation matrices in the velocity update rule of several PSO variants.

M. R. Bonyadi, Z. Michalewicz, and X. Li, "An analysis of the velocity updating rule of the particle swarm optimization algorithm," *Journal of Heuristics,* vol. 20, pp. 417-452, 2014

Bonyadi, M.R., Michalewicz, Z. & Li, X. (2014) An analysis of the velocity updating rule of the particle swarm optimization algorithm.
Journal of Heuristics, v. 20(4), pp. 417-452

NOTE:
This publication is included on pages 59-94 in the print copy of the thesis held in the University of Adelaide Library.

It is also available online to authorised users at:

http://doi.org/10.1007/s10732-014-9245-2

3.2 Rotation invariance and local convergence

The article included in this subsection studies local convergence and rotation sensitivity properties of the original PSO.

M. R. Bonyadi and Z. Michalewicz, "A locally convergent rotationally invariant particle swarm optimization algorithm," *Swarm Intelligence*, vol. 8, pp. 159-198, 2014

Bonyadi, M.R. & Michalewicz, Z. (2014) A locally convergent rotationally invariant particle swarm optimization algorithm.
Swarm Intelligence, v. 8(3), pp. 159-198

NOTE:
This publication is included on pages 96-135 in the print copy of the thesis held in the University of Adelaide Library.

It is also available online to authorised users at:

http://doi.org/10.1007/s11721-014-0095-1

3.3 Analysis of the standard particle swarm 2011

The article included in this subsection studies stability, local convergence, and transformation sensitivity of a standard particle swarm optimization that was proposed in 2011.

M. R. Bonyadi and Z. Michalewicz, "SPSO2011 – Analysis of stability, local convergence, and transformation sensitivity," *Evolutionary Computation,* (Invited paper)

SPSO2011 - Analysis of stability, local convergence, and transformation sensitivity

Mohammad Reza Bonyadi mrbonyadi@cs.adelaide.edu.au, vardiar@gmail.com
Department of Computer Science, The University of Adelaide, Adelaide, 5005, Australia

Zbigniew Michalewicz zbyszek@cs.adelaide.edu.au
Department of Computer Science, The University of Adelaide, Adelaide, 5005, Australia
Institute of Computer Science, Polish Academy of Sciences, ul. Ordona 21, 01-237 Warsaw, Poland
Polish-Japanese Institute of Information Technology, ul. Koszykowa 86, 02-008 Warsaw, Poland

Abstract

In any particle swarm optimization (PSO) variant it is essential to guarantee convergence of particles to a point in the search space; this property is called stability of particles. It is also important that a PSO variant converges to a local optimum; this is called the local convergence property. Further, it is usually expected that the performance of a PSO variant is not affected by any linear transformation of the search space; this property is called the transformation sensitivity. In this paper, these three properties, i.e., stability of particles, local convergence, and transformation sensitivity, are investigated for a particular variant of PSO called Standard PSO2011 (SPSO2011). We experimentally identify boundaries for the coefficients in the velocity update rule of this algorithm in a way that particles converge to a point in the search space. Our experiments show that, unlike earlier variants of PSO, these boundaries are dependent on the number of dimensions of the problem. We also determine boundaries for coefficients associated with different behaviors (including non-oscillatory, harmonic, and zigzagging) of particles before convergence through some experimental analysis of particles positions. In addition to stability analysis, we investigate local convergence property of this algorithm. We show that SPSO2011 is not locally convergent in general case. We provide sufficient condition for local convergence (together with the proof of local convergence) for a general formulation which represents updating rules of a large class of PSO variants. We modify SPSO2011 in such a way that it satisfies that sufficient condition, hence, the modified SPSO2011 is locally convergent. Finally, we provide a proof and experimental evidence that the algorithm is not sensitive to rotation, scale, and translation of the search space.

Keywords

Particle swarm optimization, stability analysis, local convergence, transformation invariance

1 Introduction

Particle swarm optimization (PSO) is a stochastic population-based optimization method developed by Kennedy and Eberhart (1995). The algorithm has been successfully applied to many problems such as artificial neural network training, function optimization, and pattern classification (Engelbrecht, 2005; Poli, 2008), to name but a few.

©200X by the Massachusetts Institute of Technology

Since 1995, different aspects of the original version of PSO have been modified and many variants have been proposed. Because of the emergence of many variants, some researchers have defined standard version for the PSO algorithm (PCS, 2006; Bratton and Kennedy, 2007; Clerc, 2011). The standard version is updated every few years by incorporating new advances into the earlier standard version. The most recent standard version for PSO was defined in 2011 (Clerc, 2011); it is called Standard PSO2011 (SPSO2011). SPSO2011 was applied to some continuous space benchmark problems and its results were reported to set a baseline for further research. Although there exists some studies on application (Zambrano-Bigiarini et al., 2013) of SPSO2011 on continuous space optimization benchmarks, there have not been any attempts to investigate some aspects of SPSO2011 including stability of particles (Clerc and Kennedy, 2002; Trelea, 2003), local convergence (Lehre and Witt, 2013; Van den Bergh and Engelbrecht, 2010), and transformation sensitivity (Wilke et al., 2007).

In this paper, stability of particles, local convergence, and transformation sensitivity of SPSO2011 are analyzed. For the stability analysis, we use an algorithm (called Estimate Coefficients Boundaries, ECB) which can estimate the boundaries of coefficients for a PSO variant so that particles are stable. In order to determine if identified boundaries by ECB are aligned with the theoretical findings, ECB is applied to PSO variant for which these boundaries are known. ECB is used to estimate the coefficients of the velocity updating rule of SPSO2011. We also analyze the behavior of the position of particles before convergence. We investigate the local convergence property of SPSO2011 and we prove that this algorithm is not locally convergent in the general case. We propose a simple modification and prove that it is effective to address this issue. Finally, it is proven that the algorithm is invariant from rotation, scale, and translation. Because many researchers have emphasized the importance of investigating rotation sensitivity in optimization algorithms (Hansen et al., 2011; Bonyadi and Michalewicz, 2014), this property of the algorithm is also tested experimentally and it is compared with earlier PSO variants. Note that the experimental evidence is provided to only show the difference between rotation sensitive and rotation insensitive PSO variants.

The rest of the paper is organized as follows. Section 2 provides some background information about PSO and its variants. We describe existing approaches for stability analysis of PSO variants and we discuss stability of particles in SPSO2011 in section 3. In section 4 some analyses conducted by earlier studies for local convergence property of PSO variants are reviewed and this property for SPSO2011 is investigated. In section 5 we start with an overview of existing transformation sensitivity analyses for some PSO variants and then we provide a proof and experimental evidence that SPSO2011 is invariant from rotation, scale, and translation. Section 6 concludes the paper.

2 Particle swarm optimization

Each particle in the Original PSO (OPSO) (Eberhart and Kennedy, 1995) contains three vectors:

- Position ($\vec{x}_t^i$) is the position of the ith particle in the tth iteration. This is used to evaluate the particle quality
- Velocity ($\vec{V}_t^i$) is the direction and length of movement of the ith particle in the tth iteration
- Personal best ($\vec{p}_t^i$) is the best position[1] that the ith particle has visited in its lifetime

[1] In general, personal best can be a set of best positions, but all PSO types listed in this paper use single

(up to the tth iteration). This vector serves as a memory for keeping knowledge of best found solutions (Eberhart and Kennedy, 1995).

At every iteration, the velocity of each particle is updated by

$$\vec{V}_{t+1}^{i} = \vec{V}_{t}^{i} + \phi_1 R_{1t} \left(\vec{p}_{t}^{i} - \vec{x}_{t}^{i}\right) + \phi_2 R_{2t} \left(\vec{g}_{t} - \vec{x}_{t}^{i}\right) \tag{1}$$

where ϕ_1 and ϕ_2 are two real numbers called *acceleration coefficients*[2] and $\vec{p}_{t}^{i}$ and $\vec{g}_{t}$ are the personal best (of the particle i) and the global best vectors, respectively, at iteration t. Also, the role of vectors $PI = \vec{p}_{t}^{i} - \vec{x}_{t}^{i}$ (Personal Influence) and $SI = \vec{g}_{t} - \vec{x}_{t}^{i}$ (Social Influence) is to attract particles to move toward known quality solutions, i.e., personal and global best vectors. Moreover, R_{1t} and R_{2t} are two $d \times d$ diagonal matrices (Clerc, 2006; Montes de Oca et al., 2009), where their elements are random numbers distributed uniformly ($U(0, 1)$) in the interval [0, 1]. Note that matrices R_{1t} and R_{2t} are generated at each iteration for each particle separately. The position of the particles is updated by

$$\vec{x}_{t+1}^{i} = \vec{x}_{t}^{i} + \vec{V}_{t+1}^{i} \tag{2}$$

OPSO was studied by many researchers since 1995 and many new variants were proposed. As an example, the velocity updating rule was modified (Shi and Eberhart, 1998) where previous velocity ($\vec{V}_{t}^{i}$) was multiplied by an inertia weight (ω) to control the impact of $\vec{V}_{t}^{i}$ on the movement of particles (we call this PSO variant as Inertia PSO, IPSO, in this paper). In fact, the velocity update rule for IPSO was written as

$$\vec{V}_{t+1}^{i} = \omega\vec{V}_{t}^{i} + \phi_1 R_{1t} \left(\vec{p}_{t}^{i} - \vec{x}_{t}^{i}\right) + \phi_2 R_{2t} \left(\vec{g}_{t} - \vec{x}_{t}^{i}\right) \tag{3}$$

where ω is inertia weight. As many new PSO variants emerged (see (Poli et al., 2007) for a review paper on some variants), it became harder to point to one of those as a standard version. This was causing difficulties for researchers to compare their new PSO variants with a baseline algorithm to find out if their method performs acceptably well. Thus, some researchers designed standard version of PSO that was updated frequently (every couple of years)[3]. Standard PSOs were updated according to new findings related to the algorithm, including new values for coefficients, population size, topology, velocity updating rule, and adaptations (see for example (Bratton and Kennedy, 2007)).

The most recent standard PSO algorithm (SPSO2011) was proposed in (Clerc, 2011) and later was applied to some standard benchmarks by Zambrano-Bigiarini et al. (2013). In SPSO2011, the velocity updating rule was written as:

$$\vec{V}_{t+1}^{i} = \omega\vec{V}_{t}^{i} + H\left(\vec{G}_{t}^{i}, \|\vec{G}_{t}^{i} - \vec{x}_{t}^{i}\|\right) - \vec{x}_{t}^{i} \tag{4}$$

where $H\left(\vec{G}_{t}^{i}, \|\vec{G}_{t}^{i} - \vec{x}_{t}^{i}\|\right)$ is a spherical distribution with the center $\vec{G}_{t}^{i}$ and radius $\|\vec{G}_{t}^{i} - \vec{x}_{t}^{i}\|$, $\vec{G}_{t}^{i} = (\vec{L}_{t}^{i} + \vec{P}_{t}^{i} + \vec{x}_{t}^{i})/3$, where $\vec{P}_{t}^{i} = \vec{x}_{t}^{i} + c_1(\vec{p}_{t}^{i} - \vec{x}_{t}^{i})$, $\vec{L}_{t}^{i} = \vec{x}_{t}^{i} + c_2(\vec{l}_{t}^{i} - \vec{x}_{t}^{i})$, and $\vec{l}_{t}^{i}$ is the best personal best among all particles connected to particle i (known as local best). In (Zambrano-Bigiarini et al., 2013), the values of c_1 and c_2 were set to $\phi_1 R_{1t}$ and $\phi_2 R_{2t}$, respectively, where $R_{?t}$ are random diagonal matrices and $\phi_?$ are constant values. We

personal best

[2]These two coefficients control the effect of personal and global best vectors on the movement of particles and they play an important role in the convergence of the algorithm. They are usually determined by a practitioner or by the dynamic of the particles movement.

[3]Some source codes and documentation for these standard PSOs can be found in http://www.particleswarm.info/Programs.html

refer to the SPSO2011 variant which was described in (Zambrano-Bigiarini et al., 2013) throughout this paper unless specified. Note that, in (Clerc, 2011), it was suggested that if $\vec{l}_t^i = \vec{p}_t^i$ then calculate $\vec{G}_t^i = (\vec{P}_t^i + \vec{x}_t^i)/2$, which was not mentioned in (Zambrano-Bigiarini et al., 2013). It was claimed that SPSO2011 is rotation invariant; however, no proper proof for this claim was provided.

3 Stability analysis

One of the earliest convergence analysis of stochastic optimization algorithms was conducted by Matyas (1965), which was later followed by Baba (1981); Solis and Wets (1981). An iterative stochastic optimization algorithm (optimization algorithm in short) is said to converge in probability (to converge in short) if

$$\forall \epsilon > 0, \lim_{t \to \infty} P(||\vec{x}_t - \vec{X}|| < \epsilon) = 1 \quad (5)$$

where P is the probability measure, $\vec{x}_t$ is a generated solution by the optimization algorithm (a point in the search space) at iteration t, ϵ is a small positive value, and $\vec{X}$ is a point in the search space. Analysis of convergence to a point (also known as stability analysis) is usually conducted for an iterative stochastic optimization algorithm to understand whether the sequence of the generated solutions by the algorithm is convergent.

In this section we analyze the stability and behavior of particles in SPSO2011 and IPSO.

3.1 Stability analysis for IPSO

Stability analysis for IPSO was performed by many researchers (Clerc and Kennedy, 2002; Van den Bergh and Engelbrecht, 2006; Jiang et al., 2007; Poli, 2009), to name a few. One of the aims of the stability analysis for IPSO was to find boundaries for the coefficients of velocity update rule (i.e. inertia weight and acceleration) so that positions of particles converge to a point in the search space. The set of all boundaries for all coefficients of a PSO variant that guarantee convergence to a point is called *convergence boundary* in this paper. As an example, Fig. 1 shows the convergence boundary for IPSO identified in (Jiang et al., 2007).

The position of a particle in IPSO is called a stable position, also known as equilibrium point (Trelea, 2003), if $\vec{x}_{t+1}^i = \vec{x}_t^i$ (particle does not move anymore) for all $t > t'$. It is obvious that $\vec{V}_{t+1}^i = 0$ is a necessary condition for the stability of particles because $\vec{x}_{t+1}^i = \vec{x}_t^i + \vec{V}_{t+1}^i$ (Poli, 2009; Trelea, 2003). Also, the necessary condition for $\vec{V}_{t+1}^i = 0$ is that $\vec{p}_t^i = \vec{g}_t$ ($\vec{p}_t^i = \vec{l}_t^i$) for all i (see also (Poli, 2009) for detailed analysis).

Because the aim of stability analysis is to find convergence boundary for a PSO variant, it is considered that $\vec{p}_t^i$ and $\vec{g}_t$ are not updated during the run. Also, usually velocity and position updating rules are analyzed for an arbitrary particle i in a one dimensional space. The reason for considering a one dimensional space is that in IPSO all calculations (including generation of the random values on the diagonal of R_{1t} and R_{2t}) are done in each dimension independently, hence, all analyses in one dimensional case is generalizable to the multi-dimensional case as well (Poli, 2009).

3.2 Stability analysis for SPSO2011

All stability analyses for IPSO were done for a one dimensional particle and, as each dimension was updated separately in IPSO, the results were generalizable to any number of dimensions. This assumption is not applicable for stability analysis of SPSO2011

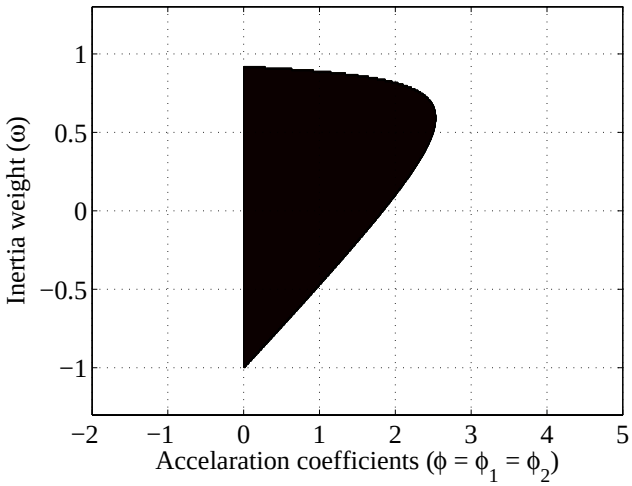


Figure 1: The dark area represents the convergence boundary for IPSO that was reported in (Jiang et al., 2007) through some theoretical analysis

because the velocity vector in SPSO2011 is not updated for each dimension separately. In SPSO2011 the operator $\vec{c} = H(\vec{a}, b)$ (for an arbitrary vector $\vec{a}$ and scalar value b) generates a random point in the space with center $\vec{a}$ and radius b. Hence, the process of generating $\vec{c}$ is not decomposable to a dimension-wise process. Therefore, any analysis of stability of particles for SPSO2011 need to consider the general case of d-dimensional space, i.e., $\|\vec{V}_t^i\| = 0$ at the equilibrium point, rather than considering each dimension separately. Whereas repeating previous analyses that were done for IPSO is not applicable in a general d-dimensional case, we conduct an experimental study to estimate the convergence boundary for SPSO2011. For that purpose we assume that

1. the personal best and global best vectors are equal for all time from the current time ($\vec{p}_t^i = \vec{g}_t$ for all $t > t'$),
2. personal best is not updated ($\vec{p}_{t+1}^i = \vec{p}_t^i$ for all $t > t'$, i.e., personal best is not updated after t'), as it was assumed in earlier stability analysis for IPSO,
3. $\vec{x}_{t'}^i = \vec{p}_{t'}^i$ (the point $\vec{p}_{t'}^i$ is sampled by $\vec{x}_{t'}^i$),
4. there are arbitrary number of particles in the swarm,
5. $\phi = \phi_1 = \phi_2$ for the sake of simplicity (this was also assumed in (Poli, 2009) for analyses),
6. $t' = 0$.

With these settings, we propose an experiment (called Estimate Convergence Boundary, ECB) to estimate the convergence boundary for a PSO variant.

As $\vec{g}_t$ is not updated for any $t > t'$, we can safely analyze one particle (particle i) in the swarm. The presence of other particles in the swarm does not influence the stability of particle i because the only information that is propagated through particles

Algorithm 1 ECB (Estimate Convergence Boundaries)

1: Input $\phi_a, \phi_s, \phi_e, \omega_a, \omega_s, \omega_e, d, r, maxIter$
2: **for** $\phi = \phi_a : \phi_s : \phi_e$ **do**
3: $\phi_1 = \phi_2 = \phi$
4: **for** $\omega = \omega_a : \omega_s : \omega_e$ **do**
5: Run the PSO algorithm for a predefined number of iterations (maxIter) and number of runs (r) for the given $\phi_1 = \phi_2 = \phi$ and ω
6: $\vec{K}_{avr}$=average of the final velocity vectors over r runs (separately for each dimension)
7: $\vec{U}_{avr}$=average of the initial velocity vectors over r runs (separately for each dimension)
8: $K_\omega = log_{10}\left(\frac{\|\vec{K}_{avr}\|}{\|\vec{U}_{avr}\|}\right)$
9: **end for**
10: $Y^d_{\phi,\omega} = K_\omega$ for all ω in $\omega_a : \omega_s : \omega_e$
11: **end for**

is the position of $\vec{g}_t$, which is not updated anymore. This assumption is also correct for other topologies ($\vec{l}_t^i$ for each particle rather than $\vec{g}_t$) as the personal best for other particles can be assumed to be steady as well.

In ECB, we change the value of coefficients (ϕ and ω) and, for each pair, we run the PSO algorithm for $maxIter$ iterations r times. The final velocities for these r runs are averaged for each dimension and the norm of the resulted vector (this vector is called $\vec{K}_{avr}$) is divided by the norm of the average of $\vec{V}_0^i$s (initialized velocities) for all runs (this vector is called $\vec{U}_{avr}$). The value of $log_{10}(\|\vec{K}_{avr}\|/\|\vec{U}_{avr}\|)$ is stored in $Y^d_{\phi,\omega}$ for each ω and ϕ (see Algorithm ECB). Thus, $Y^d_{\phi,\omega}$ represents the logarithm of the overall rate of changes in the norm of the velocity vector ($\|\vec{V}_t^i\|$). Also, we initialize $\vec{x}_0^i = \vec{p}_0^i$ as well as $\vec{V}_0^i$ to random values in the d dimensional space in each run separately. It is expected that if the setting of coefficients (ω and ϕ) for a run is in the convergence boundary, the value of $\|\vec{V}_t^i\|$ shrinks with regard to $\|\vec{V}_0^i\|$. Thus, we expect that the value of $Y^d_{\phi,\omega}$ is small if ω and ϕ are in the convergence boundary.

To test the validity of this experimental approach (ECB algorithm) for identifying convergence boundary, we applied this approach to find the convergence boundary for IPSO with $\phi_a = -2$, $\phi_s = 0.1$, $\phi_e = 8.5$, $\omega_a = -1.3$, $\omega_s = 0.01$, $\omega_e = 1.3$, $r = 10$, $maxIter = 2000$ for $d = 1$ and $d = 10$ (see Fig. 2 (a) and Fig. 2 (b)).

The color spectrum in Fig. 2(a and b) represents the value of $Y^d_{\phi,\omega}$ for IPSO for different values of inertia weight and acceleration coefficients. In the boundaries where $Y^d_{\phi,\omega}$ is relatively small (dark areas), the particles have became stable. Also, as it is clear in Fig. 2(a and b), when $\omega = 0$ the stability of the particles is very probable for all values of ϕ and d. This was in fact expected as if $\omega = 0$, and because $\vec{x}_0^i = \vec{p}_0^i$ and $\vec{p}_t^i = \vec{g}_t$ for all t, the value of $\vec{V}_t^i$ becomes zero for all t, which imposes stability. Of course this case is not interesting from application point of view. Figs. 2(a and b) show that the convergence boundary for IPSO is similar for different number of dimensions (the value of $Y^d_{\phi,\omega}$ for $d = 1$ and 10 is very close for the tested ω and ϕ). This indeed confirms the assumption by earlier analyses, e.g. (Trelea, 2003), for IPSO that the analysis for 1 dimensional space is generalizable to analysis for d-dimensional space. Also, these results are very similar to what was reported in (Jiang et al., 2007; Poli, 2009) for the

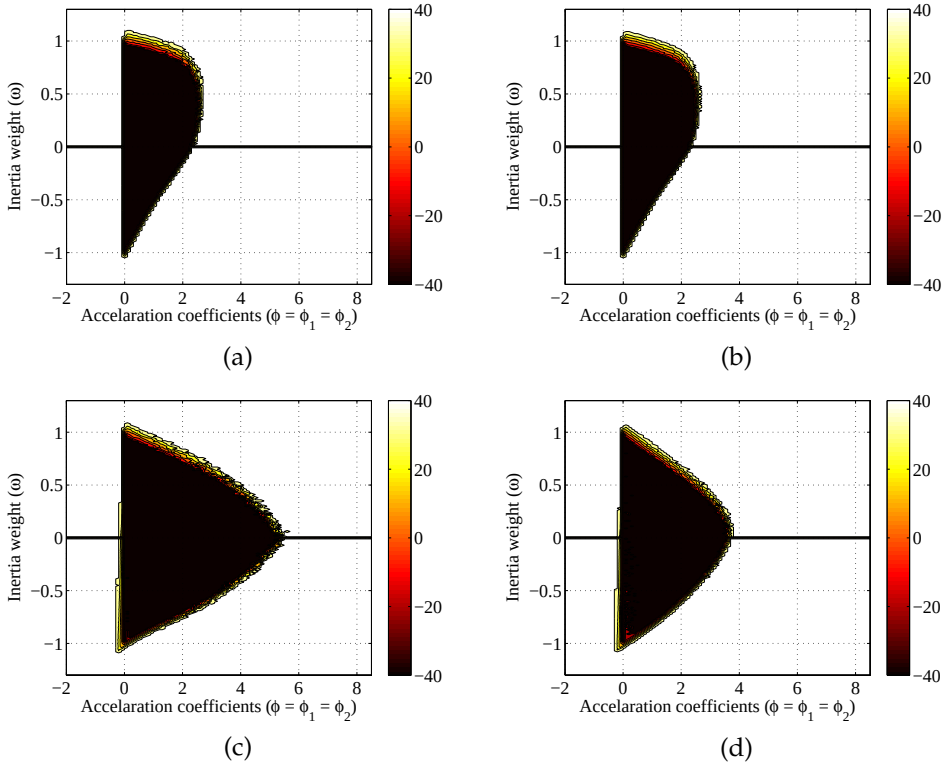


Figure 2: The boundaries for ω and ϕ so that IPSO (a and b) and SPSO2011 (c and d) are convergent. The darker the area is, the smaller the value of $Y^d_{\phi,\omega}$. Thus, the darker the area is, the more probable that the algorithm is convergent with those coefficients. The color spectrum shows the value of the matrix $Y^d_{\phi,\omega}$. Experiments were conducted for $d = 1$ (a and c) and $d = 10$ (b and d)

stability of particles (see also Fig. 1 and compare it with Fig. 2(a)). This in fact confirms that the experimental study proposed in this paper (algorithm ECB) can estimate the convergence boundary of IPSO.

We now apply ECB to find the convergence boundary for SPSO2011. We used the following parameters for ECB: $\phi_a = -2$, $\phi_s = 0.1$, $\phi_e = 8.5$, $\omega_a = -1.3$, $\omega_s = 0.01$, $\omega_e = 1.3$, $r = 10$, $maxIter = 1,000$ for $d = 1$ and $d = 10$ (see Fig. 2 (c and d)). Figs. 2 (c and d) show that the convergence boundary for SPSO2011 is changed by changing the number of dimensions. Also, the boundaries look very different from the one for IPSO.

To test the effects of convergence boundary on SPSO2011, we applied SPSO2011 to a sphere function ($f(x) = \sum_{i=1}^{d} x_i^2$) with $d = 1$ and $d = 10$ with two different settings for (ω, ϕ)—the first setting (0.73, 1.5) that is frequently used in PSO literature for IPSO (see (Clerc and Kennedy, 2002; Montes de Oca et al., 2009) for example) and the second setting (0.72, 1.1) selected because SPSO2011 is convergent for both $d = 1$ and $d = 10$ with this setting according to the results from ECB algorithm. The value of $Y^d_{\phi,\omega}$ for these different settings are:

- $Y^1_{0.72,1.1} = -40$

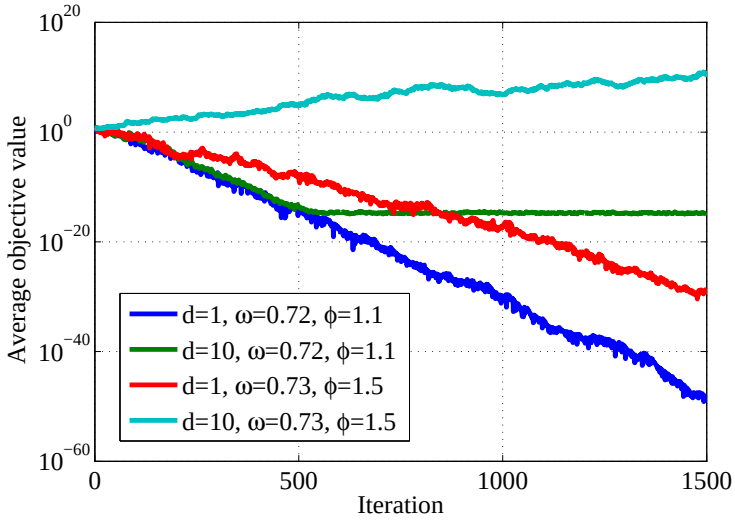


Figure 3: Average value of the norm of the velocity for a randomly selected particle over 100 runs for different settings

- $Y^{10}_{0.72,1.1} = -40$
- $Y^{1}_{0.73,1.5} = -40$
- $Y^{10}_{0.73,1.5} = 17.04$

These values suggest that the particles should become stable in all cases except $d = 10$ and $(\omega, \phi) = (0.73, 1.5)$. We applied SPSO2011 to the sphere function with the two aforementioned settings and for $d = 1$ and $d = 10$, 100 times each, and recorded the norm value of the velocity vector of a randomly selected particle at each run. The average of this norm value over 100 runs for each case is shown in Fig. 3.

It is clear in Fig. 3 that for the first setting of coefficients ($\omega = 0.73$ and $\phi = 1.5$), the velocity of the randomly selected particles decreased over iterations for the sphere function with $d = 1$, while it increased to 1E+12 for this function with $d = 10$. However, this value was decreased for the sphere function with $d = 1$ or 10 when the second setting of coefficients ($\omega = 0.72$ and $\phi = 1.1$) was considered. This means that appropriate values of coefficients for SPSO2011 is different from that of IPSO and it needs more investigations.

3.3 Convergence behavior

If the value of coefficients in a PSO variant are selected in the convergence boundary then particles converge to their equilibrium. During the run, however, particles oscillate in different ways (e.g. harmonic, zigzagging) around their equilibrium until they collapse on it (converge). These different oscillations are in fact a consequence of different values of coefficients of the velocity update rule (Trelea, 2003). Fig. 4 shows examples of these oscillations for IPSO.

These different behaviors potentially affect the quality of final solutions found by the particles. As an example, a non-oscillatory behavior (Fig. 4 (a)) causes particles

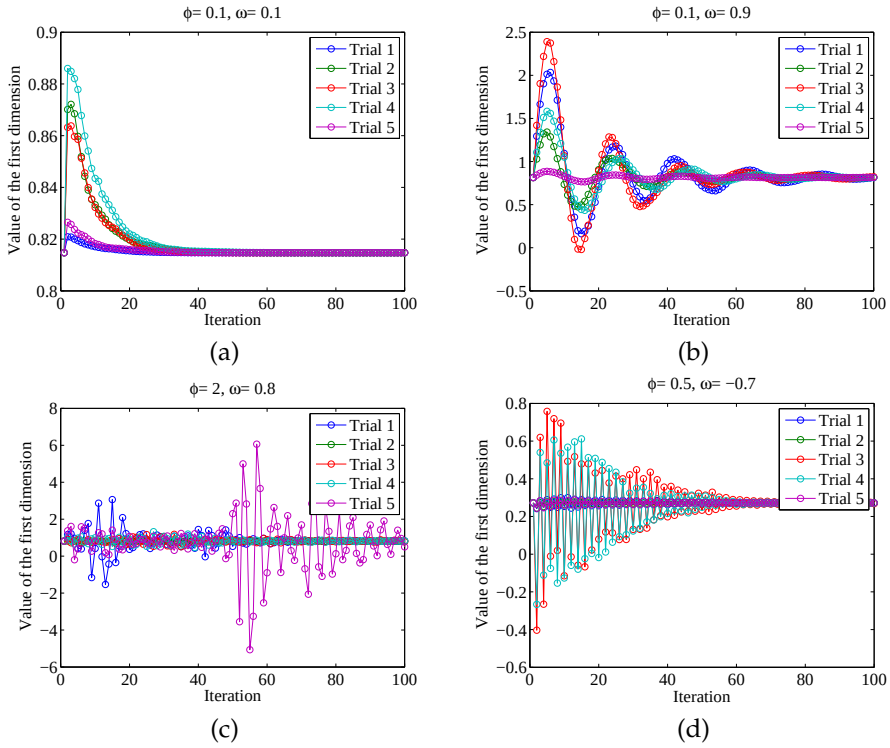


Figure 4: Examples for (a) non-oscillatory convergence, (b) harmonic oscillation, (c) harmonic combined with zigzagging, and (d) zigzagging, oscillations of the position of particles

to search only one side of each dimension of the equilibrium point. This behavior can be beneficial when searching boundaries of the search space (feasible space) are required. Harmonic behavior (Fig. 4 (b)) is beneficial in the exploitation phase as particles smoothly oscillate around their equilibrium point and higher quality solutions around the current best found points might be found. This behavior can be also beneficial when the search space is smooth (not rugged). The zigzagging behavior (Fig. 4 (d)) is more beneficial for the exploitation phase as particles jump all around the space to look for better basins of attraction. This behavior can be more useful in rugged search spaces. The combined harmonic with zigzagging behavior (Fig. 4 (c)) can be beneficial for the transition from exploration to exploitation phase. If the boundaries of coefficients associated with these behavior are known then one can design an adaptive approach that changes the values of coefficients according to the most beneficial behavior. Note however that designing such adaptive approach is not a trivial task as it need strategies to identify most beneficial behavior from the first place.

These behaviors were studied by Trelea (2003) and later by Campana et al. (2010) for IPSO. In those studies, the position update rule of the particles in IPSO was written as a recursion equation, $x_{t+1} = \omega(x_t - x_{t-1}) + \phi_1 R_{1,t}(p_t - x_t) + \phi_2 R_{2,t}(l_t - x_t)$, and was investigated by ordinary differential equations techniques (this recursion is correspondence with the following second order differential equation: $x'' = \omega(x' - x) + \phi_1 R_{1,t}(p_t - x') + \phi_2 R_{2,t}(l_t - x')$). In Trelea (2003), the characteristic equation of

this differential equation was used to calculate the boundaries associated with different behaviors (boundaries calculated in (Trelea, 2003) are shown in Fig. 6(a)). Fig. 6(a) shows that the behavior of particles for the frequently used setting of coefficients in IPSO ($\omega = 0.73$ and $\phi = 1.5$) is harmonic.

As investigation of the characteristic equation of PSO variants is not always as easy, we propose an estimation approach that uses frequency domain information of particles positions in order to specify these boundaries. Let's have a closer look at the behavior of some sample particles in the time (see Fig. 4) and frequency (see Fig. 5) domains (amplitude in the frequency domain). We used the Fourier transform to convert the first 200 samples of the particles positions from time to frequency domain. One may observe the followings:

- In Fig. 4(a), the signal (particle position) converges to its steady point without any oscillation (or with small oscillations). Thus, the maximum amplitude of this signal in the frequency domain belongs to very low frequencies (see Fig. 5(a)).
- In Fig. 4(b), the signal converges to its steady point with some harmonic oscillations. Thus, the maximum amplitude in the frequency domain belongs to a low to mid-range frequencies (see Fig. 5(b)).
- In Fig. 4(c), the signal converges to its steady point with some fast together with harmonic oscillations. Thus, the maximum amplitude in the frequency domain belongs to a mid-range to high frequencies (see Fig. 5(c)).
- In Fig. 4(d), the signal converges to its steady point with some zigzagging oscillations. Thus, the maximum amplitude in the frequency domain belongs to very high frequencies (see Fig. 5(d)).

Thus, the boundaries in Fig. 6(a) can be estimated by drawing the contour of the maximum frequency of the position of particles for different values of ω and ϕ. This contour has been presented in Fig. 6(b). By compare Fig. 6(a) and 6(b), it is clear that the pattern of the maximum amplitude in the frequency domain of particles position is very similar to what was found theoretically (Trelea, 2003) for the behavior of particles.

According to Fig. 6(b) small values for ϕ and negative values for ω result in low frequency oscillations. This is in contradict with what was observed in Fig. 6(a). The reason is that the oscillation of particles in that area (ω negative and ϕ small) is a combination of zigzagging and non-oscillatory, which results in large amplitudes in both very high and very low frequencies. Note also that, according to Fig. 6(b), the behavior of particles is zigzagging for $\phi = 0$ and negative ω. The reason is that when the value of ϕ is zero, only the inertia component affect the behavior of movement. As $\omega < 0$, the sign of the value of each dimension of velocity changes in each iteration, which implies zigzagging behavior (high frequency changes).

We calculated the maximum frequency of particles positions in SPSO2011 for 1 and 10 dimensional spaces (see Fig. 7). It is clear that the spectrum of maximum frequencies in SPSO2011 is different from that of IPSO. For a positive value of ω the behavior of particles changes from non-oscillatory to harmonic when ϕ increases. If the value of ω is negative, however, particles behave more non-oscillatory for small values of ϕ and more zigzagging for larger values of ϕ. These patters are almost consistent in both 1 and 10 dimensional cases.

It is also clear in Fig. 7 that if the value of ω is positive and smaller than 1 and the value of ϕ is positive and smaller than 2 then particles behave harmonic or non-oscillatory. Particularly, for the two sets of coefficients tested in Section 3.2 ($(\omega, \phi) =$

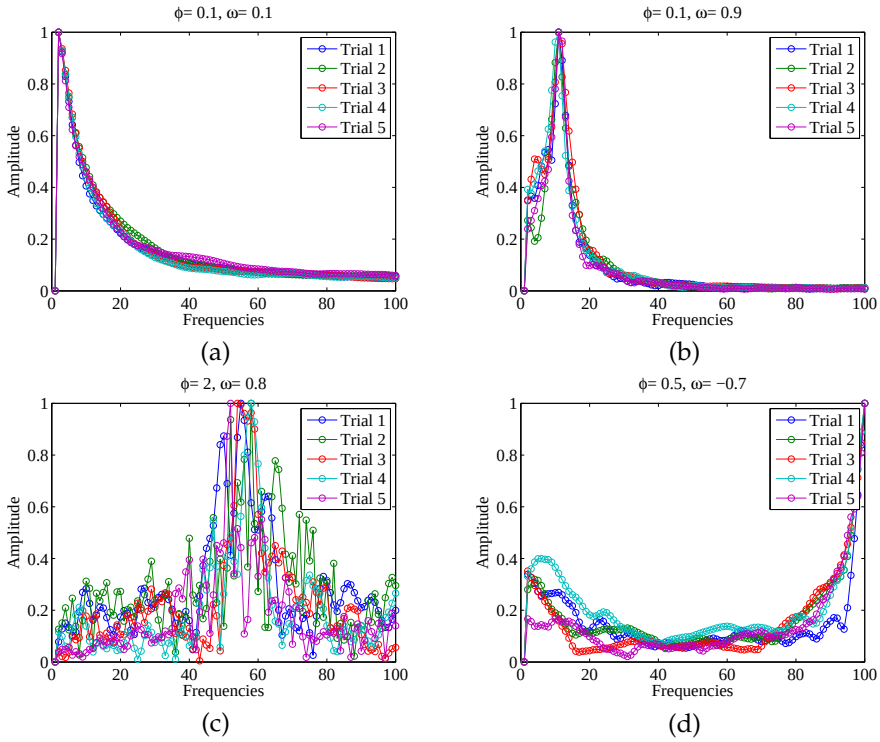


Figure 5: (a) non-oscillatory convergence, (b) slow harmonic oscillation, (c) harmonic combined with zigzagging, and (d) zigzagging

$(0.72, 1.1)$ and (0.73, 1.5)), the maximum frequency is around 20, which belongs to low to mid-range frequencies. Thus, a harmonic behavior is expected from the particles for these sets of coefficients.

4 Local convergence

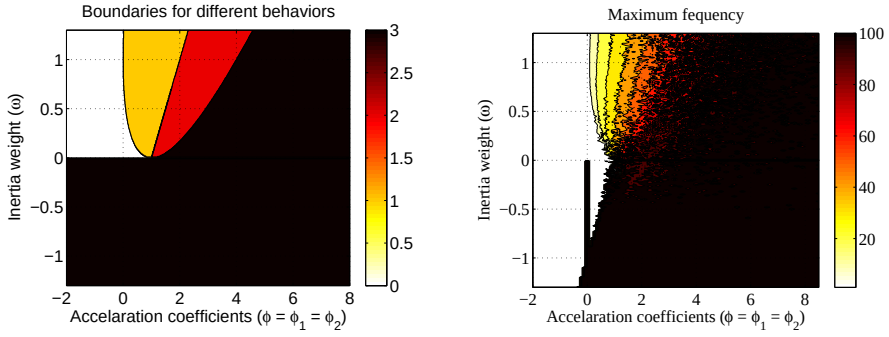
If the point $\vec{X}$ in Eq. 5 is a local optimum and an stochastic algorithm guarantees to satisfy Eq. 5 then that algorithm is said to be locally convergent. In the context of PSO, the local convergence condition is written as follows (Bonyadi and Michalewicz, 2014):

$$\forall \epsilon > 0 \; \forall i, \lim_{t \to \infty} P(||\vec{p}_t^{\,i} - \vec{X}|| < \epsilon) = 1 \tag{6}$$

In fact, if the personal bests of all particles in a PSO variant converge to a local optimum[4] then that PSO variant is locally convergent. Alternatively, as the final output of a PSO variant is the best personal best over the swarm, a PSO variant is locally convergent if (Bonyadi and Michalewicz, 2014):

$$\forall \epsilon > 0 \; \forall i, \lim_{t \to \infty} P(||\vec{g}_t - \vec{X}|| < \epsilon) = 1 \tag{7}$$

[4]A local minimum of an objective function F over the search space S is defined as: c_i is a local minimum if there exists an open interval $I_i \subseteq S$ such that $F(c_i) \leq F(x)$ for all $x \in I_i$



(a) (b)

Figure 6: Different behavior of particles before convergence in IPSO. The spectrum in (a) shows different behaviors found in (Trelea, 2003), where non-oscillatory convergence, harmonic oscillation, harmonic combined with zigzagging, and zigzagging have been represented by integers from 0 to 4 respectively. (b) shows the maximum frequency of oscillation of particles with different coefficient values

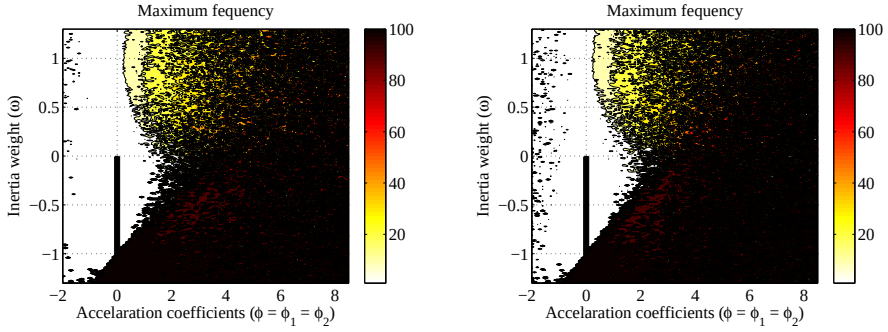
where g_t is the best found solution over the swarm. In this section, some earlier studies which have analyzed the local convergence property of IPSO are briefly reviewed and then the local convergence for SPSO2011 is investigated.

4.1 Local convergence for IPSO

Local convergence for IPSO has been investigated in (Van den Bergh and Engelbrecht, 2010; Schmitt and Wanka, 2013; Bonyadi and Michalewicz, 2014). A PSO variant is said to be locally convergent if it guarantees finding a local optimum in the search space. As it was mentioned in (Van den Bergh and Engelbrecht, 2010), if $\vec{p}_t^i = \vec{g}_t = \vec{x}_t^i$, then $\vec{x}_t^i$ is moved in the next step if $\omega\vec{V}_t^i$ is non-zero. However, it might be the case that $\vec{V}_t^i$ becomes zero, which results in $\vec{V}_{t+1}^i = 0$ and, consequently, no movement takes place in the next step. Also, as $\vec{V}_{t+1}^i = 0$ and $\vec{p}_t^i = \vec{g}_t = \vec{x}_t^i = \vec{x}_{t+1}^i$, there is no movement in even further steps (the particle is in its equilibrium). As no claim has been made about the equilibrium point, it is not clear whether $\vec{x}_t^i$ is a local optimum while it is not updated in the next steps. Thus, IPSO algorithm is not locally convergent.

A mutation operator was proposed in (Van den Bergh and Engelbrecht, 2010) moved the position of the global best particle (the particle which its personal best is $\vec{g}_t$) to a random point generated "around" current $\vec{g}_t$ (according to some distribution) rather than moving the particle according to its velocity. In this case, even if all particles stop moving, the global best particle still continue movement randomly around $\vec{g}_t$, which prevents complete stop of that particle (the algorithm was called Guaranteed Convergent PSO, GCPSO). It was proven that the algorithm is locally convergent.

The local convergence property of IPSO was also investigated in (Schmitt and Wanka, 2013). It was proven that, in a one dimensional space, IPSO is locally convergent if $n > 1$ and coefficients are set in the convergence boundary. However, the algorithm might not converge to a local optimum if $d > 1$. To address this issue, a strategy was proposed to detect if $\|\vec{V}_t^i\| + \|\vec{g}_t - \vec{x}_t^i\|$ is smaller than a constant δ. In this case, the velocity vector of particle i was regenerated randomly within a hyper-rectangle with the side length δ. It was proven that this approach guarantees local convergence.



(a) (b)

Figure 7: Different behavior of particles before convergence in SPSO2011. The spectrum in (a) shows the maximum frequency of position of a particle in a 1 dimensional space, (b) shows the maximum frequency of a randomly chosen dimension of the position of a particle in a 10 dimensional space

A PSO variant called Locally convergent Rotationally invariant Particle Swarm Optimization (LcRiPSO) was proposed by Bonyadi and Michalewicz (2014). The velocity update rule for LcRiPSO was written as:

$$\vec{V}^i_{t+1} = \omega \vec{V}^i_t + \sum_{j \in T^i_t} r^i_{jt} \phi_j (f^i_{jt}(\vec{p}^i_t) - \vec{x}^i_t) \tag{8}$$

where r^i_{jt} is a uniform random scalar in the interval $[0, 1]$ and f^i_{jt} is a function and $f : R^d \rightarrow R^d$, T^i_t is the set of all neighbors of the particle i that contribute into its velocity update rule. It was proven that LcRiPSO is locally convergent if the function f satisfies the following condition:

$$\forall \vec{y} \in S \; \exists A_y \subseteq S \; \forall z \in A_y \; \forall \delta > 0, P(||f(\vec{y}) - \vec{z}|| < \delta) > 0$$

where $\vec{y}$ is an arbitrary point in the search space S, A_y is an open set which contains $\vec{y}$, $\vec{z}$ is an arbitrary point in A_y, and δ is a positive value. In other words, LcRiPSO is locally convergent if the function f is designed in such a way that for any input vector $\vec{y}$ in the search space, there exists a region A which contains $\vec{y}$ and $f(\vec{y})$ can be located anywhere in A.

4.2 Local convergence for SPSO2011

We show that SPSO2011 is not locally convergent in at least two cases:

Case 1: assume that $\vec{p}^i_t = \vec{l}^i_t = \vec{x}^i_t$ in SPSO2011. The value of $\vec{G}^i_t$ in Eq. 4 is then calculated by

$$\vec{G}^i_t = \frac{\vec{x}^i_t + \phi_2 R_{2t} \left(\vec{l}^i_t - \vec{x}^i_t\right) + \vec{x}^i_t + \phi_1 R_{1t} \left(\vec{p}^i_t - \vec{x}^i_t\right) + \vec{x}^i_t}{3} = \vec{x}^i_t$$

Thus, in this case, $\|\vec{G}^i_t - \vec{x}^i_t\|$ is zero, which implies that $H(\vec{G}^i_t, \|\vec{G}^i_t - \vec{x}^i_t\|) = \vec{G}^i_t = \vec{x}^i_t$ for all t. Hence, the value of velocity for $t + 1$ is $\vec{V}^i_{t+1} = \omega \vec{V}^i_t$. This is exactly the same as the case we discussed in section 4.1 for IPSO (i.e. if $\vec{V}^i_t = 0$, there is no chance for the particle to move in further iterations). With using the same analysis in section

4.1, SPSO2011 is not locally convergent in general case. This issue also exists in the formulation explained in (Clerc, 2011) for SPSO2011 because if $\vec{p}_t^i = \vec{x}_t^i$:

$$\vec{G}_t^i = \frac{\vec{x}_t^i + \phi_1 R_{1t}\left(\vec{p}_t^i - \vec{x}_t^i\right) + \vec{x}_t^i}{2} = \vec{x}_t^i$$

Case 2: we also prove that SPSO2011 is not locally convergent for any one dimensional problem when $n = 1$, under specific initialization for x_0 and V_0 that happen with non-zero probability.

Theorem 1. *SPSO2011 is not locally convergent for any one dimensional problem if n=1, the particle is initialized at a point x_0 which is not a local optimum, the objective function is monotonically decreasing at x_0, $0 < \omega < 1$, $x_0 < a$, $0 < V_0 < \frac{(a-x_0-\epsilon)(1-\omega)}{\omega}$, where a is the closest local optimum to x_0 in the direction of V_0 and $0 < \epsilon < a - x_0$.*

Proof. clearly, $x_0 = g_0 = p_0$, which results in $V_1 = \omega V_0$ with any initialization. Also, $x_1 = x_0 + V_1 = x_0 + \omega V_0$. It is clear that $\omega < \omega/(1-\omega)$ which implies that $x_0 + \omega V_0 < x_0 + \frac{(a-x_0-\epsilon)(1-\omega)\omega}{(1-\omega)\omega} = a - \epsilon$. Also, as $V_0 > 0$, thus $x_0 < x_1 < a - \epsilon$. Because a is the closest optimum to x_0 and the objective function is monotonically decreasing, x_1 is a better point than x_0 for any value of $0 < \omega < 1$, which results in updating $g_1 = p_1 = x_1$. In this case, it is clear that $V_2 = \omega V_1$ which results in $x_2 = x_1 + V_2 = x_0 + V_0(\omega + \omega^2)$. As it is obvious that $\omega + \omega^2 < \omega/(1-\omega)$, thus $x_2 = x_0 + V_0(\omega + \omega^2) < x_0 + \frac{(a-x_0-\epsilon)(1-\omega)\omega}{(1-\omega)\omega} = a - \epsilon$. Also, as $V_0 > 0$, it is clear that $x_1 < x_2 < a - \epsilon$. Because $x_2 < a - \epsilon$, x_2 is better than $g_1 = p_1$. Thus, $g_2 = p_2 = x_2$. By continuing the same strategy (note that $\omega + \omega^2 + ... + \omega^t \leq \omega/(1-\omega)$ for any t), x_t is calculated by:

$$x_t = x_0 + V_0 \sum_{i=1}^{t} \omega^i < x_0 + \frac{V_0 \omega}{1-\omega} = x_0 + \frac{(a - x_0 - \epsilon)(1-\omega)\omega}{(1-\omega)\omega} = a - \epsilon$$

Also, as the objective function is monotonically decreasing, thus $x_0 < ... < x_{t-1} < x_t < a - \epsilon$. Hence, x_t moves towards a in each iteration but never meets a, which completes the proof for the theorem. □

In fact, the particle always moves towards the local optimum a, however, the velocity decreases in every iteration to a portion (ω), and, consequently, it does not arrive to the local optimum if the velocity is initialized inappropriately. Also, note that the probability of this initialization is non-zero.

4.3 A locally convergent SPSO2011

A simple modification of SPSO2011 to guarantee local convergence is to bound $\|\vec{G}_t^i - \vec{x}_t^i\|$ to a small non-zero value $\delta > 0$. Before we prove that this strategy actually fixes the local convergence issue, we introduce some definitions and lemmas.

We define four measures, $r_h(\vec{q})$, $n_h(\vec{q})$, $\psi_h(\vec{q})$, and $n_{h_1,h_2}(\vec{q}_1, \vec{q}_2)$ as follows:

Definition 1. *$r_h(\vec{q})$ is the set of all points in a hyper-ball with the center $\vec{q}$ and radius $h > 0$.*

Definition 2. *$n_h(\vec{q})$ is a random point that can be any point in $r_h(\vec{q})$ with non-zero probability.*

Definition 3. *$\psi_h(\vec{q})$ is a connected set such that $\psi_h(\vec{q}) \subseteq S$, $\vec{q} \in \psi_h(\vec{q})$, $\psi_h(\vec{q}) - \{\vec{q}\}$ is non-empty, and any point in $\psi_h(\vec{q})$ is closer than or as far as h to $\vec{q}$.*

Definition 4. *$n_{h_1,h_2}(\vec{q}_1, \vec{q}_2)$ is a random point that can be any point in $\{r_{h_1}(\vec{q}_1) \cup \psi_{h_2}(\vec{q}_2)\}$ with non-zero probability.*

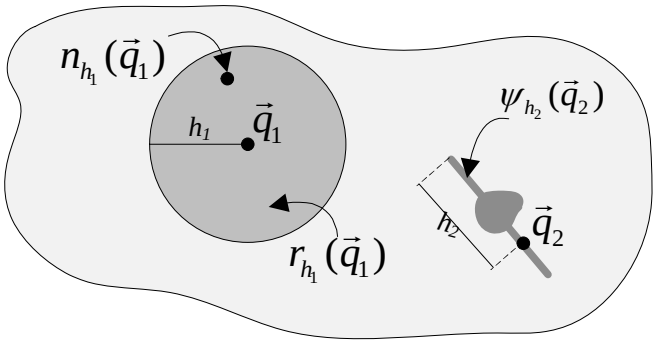


Figure 8: The light-gray area represents a search space S, the dark gray circle represents $r_{h_1}(\vec{q_1})$, and a sample random point $n_{h_1}(\vec{q_1})$ has been indicated in the circle. Also, an example of the set $\psi_{h_2}(\vec{q_2})$ has been shown in the figure. $n_{h_1,h_2}(\vec{q}_1, \vec{q}_2)$ is a random point that can be any point inside the dark gray areas with non-zero probability

According to Definition 1 any point in $r_h(\vec{q})$ can be written as $\vec{q}+\vec{a}$ where $\|\vec{a}\| \leq h$. Also, Definition 3 implies that for any point inside $\psi_h(\vec{q})$ there exists a vector $\vec{a}$ such that $\vec{q}+\vec{a} \in \psi_h(\vec{q})$ and $\|\vec{a}\| < h$. Fig. 8 shows examples of the measures defined in Definitions 1, 2, 3, and 4.

We define a local minimizer and an optimality region of the objective function F over the search space S as follows:

Definition 5. *c_i is a local minimizer of an objective function F over the search space S if there exists an open set $I_i \subset S$ such that $F(c_i) \leq F(x)$ for all $x \in I_i$.*

Definition 6. *The optimality region of the objective function F is defined as $R_\epsilon = \bigcup_i R_{\epsilon,i}$ where $R_{\epsilon,i} = \{x \in I_i : F(x) < F(c_i) + \epsilon\}$ and ϵ is an arbitrarily small positive value.*

The aim of a local search algorithm is to find a point in the search space that is within the optimality region (Solis and Wets, 1981; Van den Bergh and Engelbrecht, 2010; Bonyadi and Michalewicz, 2014).

We define a general form of stochastic algorithms, (General Stochastic Algorithm, GSA) as follows:

Definition 7. *GSA has following three steps:*

1. *initialize $\vec{p}_0$ from the search space S and set $t = 1$*
2. *generate a random sample $\vec{x}_t$ from S*
3. *generate the candidate solution $\vec{p}_t = D(\vec{p}_{t-1}, \vec{x}_t)$, set $t = t + 1$, and go to 2*

where $D(a, b)$ is defined by $D(a,b) = \begin{cases} b & F(b) < F(a) - \epsilon_0 \\ a & otherwise \end{cases}$

and ϵ_0 is a positive value that is smaller than or equal to ϵ (ϵ in the definition of $R_{\epsilon,i}$).

The operator D updates $\vec{p}_t$ if and only if the new solution $\vec{x}_t$ is better than $\vec{p}_{t-1}$ by at least the constant ϵ_0. One can set ϵ_0 to the smallest possible float/double value (Matyas, 1965; Bonyadi and Michalewicz, 2014) in computer simulations.

Lemma 1. *If a GSA guarantees*

$$\exists \epsilon > 0 \; \exists \eta > 0 \; \exists \delta \in (0, 1] \; \forall t \geq 0 \; \exists t' > 0, P(F(\vec{p}_{t+t'}) \leq F(\vec{p}_t) - \eta) \geq \delta \text{ or } \vec{p}_t \in R_\epsilon$$

then that GSA is locally convergent.

Proof. This has been proven as Lemma 1 in (Bonyadi and Michalewicz, 2014). □

In fact, if the probability that $F(\vec{p}_{t+t'})$, F is the objective function, is smaller than $F(\vec{p}_t)$ by at least η is larger than δ unless $\vec{p}_t$ is already in the optimality region then GSA is locally convergent.

Lemma 2. *If a GSA guarantees*

$$\forall t > 0 \; \exists z, h' > 0, P(\vec{x}_{t+z} = n_{h'}(\vec{p}_t)) > 0$$

then the condition in Lemma 1 is satisfied.

Proof. There are two cases: 1) there exists a set $A \subset r_{h'}(\vec{p}_t)$ such that all points in A are better than $\vec{p}_t$ by at least ϵ_0 or, 2) there is no point inside $r_{h'}(\vec{p}_t)$ that is better than $\vec{p}_t$ by at least ϵ_0. In the first case, because $\vec{x}_{t+z} = n_{h'}(\vec{p}_t)$ with non-zero probability, and $A \subset r_{h'}(\vec{p}_t)$, the point $\vec{x}_{t+z}$ has non-zero probability to be inside A, that satisfies the condition in Lemma 1 with setting $\epsilon_0 = \eta$. In the second case, $\vec{p}_t$ is already in the optimality region that also satisfies the condition in Lemma 1. Thus, in both cases the condition in Lemma 1 is satisfied, that completes the proof. □

According to Lemma 2, if there exist an iteration z that $\vec{x}_{t+z}$ has non-zero probability to be any point inside $r_{h'}(\vec{p}_t)$ then the condition in Lemma 1 is satisfied.

Lemma 3. *Any recursion in the form of $\vec{x}_{t+1} = \omega(\vec{x}_t - \vec{x}_{t-1}) + n_{h_1,h_2}(\vec{p}_t, \vec{x}_t)$, where $\vec{p}_t = D(\vec{p}_{t-1}, \vec{x}_t)$, is locally convergent for any $\omega \in (0,1)$, $\vec{x}_0 \in S$, and $\vec{p}_0 \in S$.*

Proof. Obviously, the recursion $\vec{x}_{t+1} = \omega(\vec{x}_t - \vec{x}_{t-1}) + n_{h_1,h_2}(\vec{p}_t, \vec{x}_t)$, where $\vec{p}_t = D(\vec{p}_{t-1}, \vec{x}_t)$, is an instance of GSA. Thus, if we prove that

$$\exists \epsilon > 0 \; \exists \eta > 0 \; \exists \delta \in (0,1] \; \forall t \geq 0 \; \exists t' > 0, P(F(\vec{p}_{t+t'}) \leq F(\vec{p}_t) - \eta) \geq \delta \text{ or } \vec{p}_t \in R_\epsilon$$

then, according to Lemma 3, this recursion is locally convergent.

For any $z > 0$, two cases might take place until iteration $t + z > 0$:

1. $F(\vec{p}_{t+z}) < F(\vec{p}_t)$
2. $F(\vec{p}_{t+z}) = F(\vec{p}_t)$

In the case 1, because the operator D updates $\vec{p}_t$ if the new found solution is better than $\vec{p}_t$ at least by ϵ_0, condition in Lemma 1 is satisfied for $\delta = 1$ if we set $\epsilon_0 = \eta$ because $\vec{p}_t$ has been already updated. We continue with the case 2. Note that, from here on we always write $\vec{p}_t$ as $\vec{p}_{t+z} = \vec{p}_t$ for all z. For the case 2, we prove that there exists $z > 0$ and $h' > 0$ such that $P(\vec{x}_{t+z} = n_{h'}(\vec{p}_t)) > 0$, that, according to Lemma 2, satisfies the condition in Lemma 3 and completes the proof for the local convergence.

One can write $\vec{x}_{t+2} = \omega(\vec{x}_{t+1} - \vec{x}_t) + n_{h_1,h_2}(\vec{p}_t, \vec{x}_{t+1})$ that indicates $\vec{x}_{t+2} = \omega\left(\omega(\vec{x}_t - \vec{x}_{t-1}) + n_{h_1,h_2}(\vec{p}_t, \vec{x}_t) - \vec{x}_t\right) n_{h_1,h_2}(\vec{p}_t, \vec{x}_{t+1})$. According to Definition 4, $n_{h_1,h_2}(\vec{p}_t, \vec{x}_t)$ can be a random point written as $\vec{x}_t + \vec{a}$ where $\|\vec{a}\| \leq h_2$. Thus, with non-zero probability, $\vec{x}_{t+2} = \omega\left(\omega(\vec{x}_t - \vec{x}_{t-1}) + \vec{x}_t + \vec{a_1} - \vec{x}_t\right) + n_{h_1,h_2}(\vec{p}_t, \vec{x}_{t+1}) = \omega^2(\vec{x}_t - \vec{x}_{t-1}) + \omega\vec{a_1} + n_{h_1,h_2}(\vec{p}_t, \vec{x}_{t+1})$ where $\vec{a_1}$ is a vector and $\|\vec{a_1}\| \leq h_2$. Also, one can write $\vec{x}_{t+3} = \omega(\vec{x}_{t+2} - \vec{x}_{t+1}) + n_{h_1,h_2}(\vec{p}_t, \vec{x}_{t+2}) = \omega(\omega^2(\vec{x}_t - \vec{x}_{t-1}) + \omega\vec{a_1} + n_{h_1,h_2}(\vec{p}_t, \vec{x}_{t+1}) - \vec{x}_{t+1}) + n_{h_1,h_2}(\vec{p}_t, \vec{x}_{t+2})$. Again, according to Definition 4, $\vec{x}_{t+3} = \omega^3(\vec{x}_t - \vec{x}_{t-1}) + \omega^2\vec{a_1} + \omega\vec{a_2} + n_{h_1,h_2}(\vec{p}_t, \vec{x}_{t+2})$ with non-zero probability, where $\|\vec{a_2}\| \leq h_2$. If we continue the same strategy, we can write $\vec{x}_{t+z}$ as follows:

$$\vec{x}_{t+z} = \omega^z(\vec{x}_t - \vec{x}_{t-1}) + \sum_{j=1}^{z-1} \omega^{z-j}\vec{a_j} + n_{h_1,h_2}(\vec{p}_t, \vec{x}_{t+z-1}) \quad (9)$$

where $\|\vec{a_j}\| \leq h_2$ for all j.

If we prove that

$$\forall h_1 > 0 \; \exists z, h_2, h' > 0, r_{h'}(\vec{p}_t) \subseteq Y$$

where Y is any possible point that can be sampled by $\omega^z(\vec{x}_t - \vec{x}_{t-1}) + \sum_{j=1}^{z-1} \omega^{z-j}\vec{a_j} + n_{h_1,h_2}(\vec{p}_t, \vec{x}_{t+z-1})$, then $\vec{x}_{t+z}$ can be any point in $r_{h'}(\vec{p}_t)$ with non-zero probability, that satisfies condition in Lemma 2. Clearly, the maximum length of $\sum_{j=1}^{z-1} \omega^{z-j}\vec{a_j}$ is $\frac{\omega-\omega^z}{1-\omega}h_2$. Let us assume $h' = h_1 - [\omega^z(\|\vec{x}_t - \vec{x}_{t-1}\|) + \frac{\omega-\omega^z}{1-\omega}h_2]$. Thus, we need to guarantee

$$\forall h_1 > 0 \; \exists z, h_2 > 0, h_1 - [\omega^z(\vec{x}_t - \vec{x}_{t-1}) + \frac{\omega - \omega^z}{1-\omega}h_2] > 0$$

If we set $h_2 < \frac{(1-\omega)(h_1 - \omega^z(\|\vec{x}_t - \vec{x}_{t-1}\|))}{\omega - \omega^z}$ then $h_1 - [\omega^z(\vec{x}_t - \vec{x}_{t-1}) + \frac{\omega-\omega^z}{1-\omega}h_2] > 0$ is guaranteed for large enough z. The reason is that for any h_1 the term $\frac{(1-\omega)(h_1 - \omega^z(\|\vec{x}_t - \vec{x}_{t-1}\|))}{\omega - \omega^z}$ is positive if z is large enough. In fact, a large value for z diminishes the effect of $\omega^z\|\vec{x}_t - \vec{x}_{t-1}\|$ and, as $\omega - \omega^z$, h_1, and $1 - \omega$ are larger than 0, $\frac{(1-\omega)(h_1 - \omega^z(\|\vec{x}_t - \vec{x}_{t-1}\|))}{\omega - \omega^z}$ is also larger than 0. Hence, for any h_1, there exists z and h_2 such that $h' > 0$ with non-zero probability. Therefore, $P(\vec{x}_{t+z} = n_{h'}(\vec{p}_t)) > 0$, i.e., $\vec{x}_{t+z}$ can be any point inside a hyper-ball with the center $\vec{p}_t$ and radius $h' > 0$ with non-zero probability. According to Lemma 2, in this case, the condition in Lemma 1 is guaranteed, that guarantees local convergence of the algorithm and completes the proof. □

This lemma can be useful to prove local convergence of any PSO variant that follows the mentioned recursion form. In order to prove local convergence of the modified SPSO2011, we first need to find whether the modified SPSO2011 follows the recursion form in Lemma 3.

Lemma 4. *If*

$$\exists \delta > 0 \; \forall t > 0, \|\vec{G}_t^\tau - \vec{x}_t^\tau\| > \delta$$

then there exist $h_1, h_2 > 0$ such that $H(\vec{G}_t^\tau, \|\vec{G}_t^\tau - \vec{x}_t^\tau\|)$ can be any point generated by $n_{h_1,h_2}(\vec{p}_t^\tau, \vec{x}_t^\tau)$, where τ is the index of the best particle in the swarm.

Proof. Obviously $H(\vec{G}_t^\tau, \|\vec{G}_t^\tau - \vec{x}_t^\tau\|)$ always contains $\vec{x}_t^\tau$ because $\|\vec{G}_t^\tau - \vec{x}_t^\tau\|$ is the radius of the hyper-sphere. Also, $H(\vec{G}_t^\tau, \|\vec{G}_t^\tau - \vec{x}_t^\tau\|)$ can be any point on the line segment that connects $\vec{G}_t^\tau$ and $\vec{x}_t^\tau$. As $\|\vec{G}_t^\tau - \vec{x}_t^\tau\| > \delta$ thus always this line segment exists, and we can consider $0 < h_2 < \delta$. Hence, this line segment can be considered as the set $\psi_{h_2}(\vec{G}_t^\tau)$ in Definition 3 and $h_2 > 0$ always exists[5] (see Fig. 9). If we prove that $\|\vec{G}_t^\tau - \vec{p}_t^\tau\| < \|\vec{G}_t^\tau - \vec{x}_t^\tau\|$ with non-zero probability then there exists a $h_1 > 0$ such that $H(\vec{G}_t^\tau, \|\vec{G}_t^\tau - \vec{x}_t^\tau\|)$ contains $r_{h_1}(\vec{p}_t^\tau)$ as well (see Fig. 9), which completes the proof. We calculate $\vec{G}_t^\tau = \vec{x}_t^\tau + \frac{2\phi}{3}R(\vec{p}_t^\tau - \vec{x}_t^\tau)$ where $R = R_1 + R_2$ (note that for the particle τ, $\vec{p}_t^\tau = \vec{l}_t^\tau = \vec{g}_t$, where g_t is the best solution found over the swarm, see Eq. 4). Thus, $\|\vec{G}_t^\tau - \vec{p}_t^\tau\| =$

[5] Another alternative for this set is the intersection between $r_{h_2}(\vec{x}_t^\tau)$ and $H(\vec{G}_t^\tau, \|\vec{G}_t^\tau - \vec{x}_t^\tau\|)$, where $h_2 < \delta$.

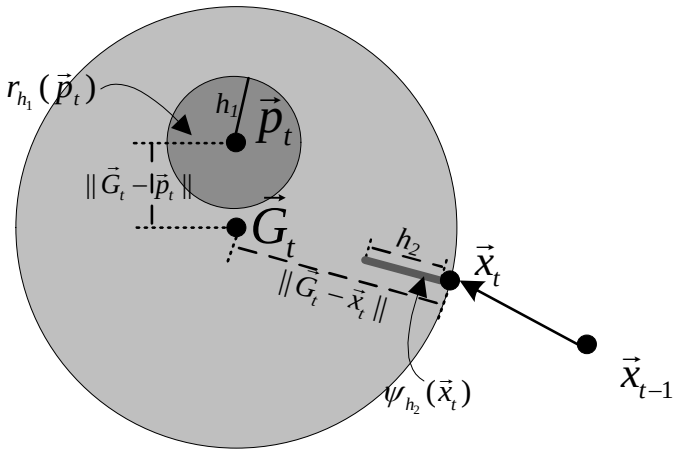


Figure 9: The dark gray area represent $n_{h_1,h_2}(\vec{p}_t^{\tau}, \vec{x}_t^{\tau})$, see Lemma 4.

$\|\vec{x}_t^{\tau} + \frac{2\phi}{3}R(\vec{p}_t^{\tau} - \vec{x}_t^{\tau}) - \vec{p}_t^{\tau}\| = \|(\frac{2\phi}{3}R - I)(\vec{p}_t^{\tau} - \vec{x}_t^{\tau})\|$ where I is the identity matrix. This value is obviously smaller than $\|\vec{G}_t^{\tau} - \vec{x}_t^{\tau}\| = \|\vec{x}_t^{\tau} + \frac{2\phi}{3}R(\vec{p}_t^{\tau} - \vec{x}_t^{\tau}) - \vec{x}_t^{\tau}\| = \|\frac{2\phi}{3}R(\vec{p}_t^{\tau} - \vec{x}_t^{\tau})\|$ with non-zero probability. Hence, there exists $h_1, h_2 > 0$ such that $H(\vec{G}_t^{\tau}, \|\vec{G}_t^{\tau} - \vec{x}_t^{\tau}\|)$ can be any point generated by $n_{h_1,h_2}(\vec{p}_t^{\tau}, \vec{x}_t^{\tau})$ with non-zero probability. □

Now we are ready to prove the following theorem:

Theorem 2. *If $\|\vec{G}_t^{\tau} - \vec{x}_t^{\tau}\| \geq \delta > 0$ and $0 < \omega < 1$ are guaranteed then SPSO2011 is locally convergent, where τ is the index of the particle that its personal best is the best solution found by the swarm at iteration t.*

Proof. The position update rule for the best particle in the swarm in SPSO2011 is written as: $\vec{x}_{t+1} = \omega(\vec{x}_t - \vec{x}_{t-1}) + H(\vec{G}_t^{\tau}, \|\vec{G}_t^{\tau} - \vec{x}_t^{\tau}\|)$. According to Lemma 4, if $\|\vec{G}_t^{\tau} - \vec{x}_t^{\tau}\| \geq \delta > 0$ then there exist $h_1, h_2 > 0$ such that $H(\vec{G}_t^{\tau}, \|\vec{G}_t^{\tau} - \vec{x}_t^{\tau}\|)$ can be any point that is generated by $n_{h_1,h_2}(\vec{p}_t^{\tau}, \vec{x}_t^{\tau})$ with non-zero probability. Thus, the best particle in SPSO2011 follows the recursion introduced in Lemma 3. Also, the best particle in SPSO2011 is updated if the new found solution is better than the previous best found solution at least by the precision of the simulation (float/double precision). Thus, SPSO2011 is locally convergent if $\|\vec{G}_t^{\tau} - \vec{x}_t^{\tau}\| \geq \delta > 0$ and $0 < \omega < 1$. □

5 Sensitivity to transformations

Sensitivity to transformations, specifically to scale, rotation, and translation, has been investigated for optimization algorithms (Wilke et al., 2007; Bonyadi and Michalewicz, 2014). It is expected that the performance of an optimization algorithm does not change if the search space is scaled, rotated, or translated (see Fig. 10).

An optimization algorithm is invariant from scale, rotation, and translation if $\hat{x}_{t+1} = sQ\vec{x}_{t+1} + \vec{b}$ for all $t > 0$ for any scalar s, orthogonal matrix Q, and vector b, where $\vec{x}_{t+1}$ is the solution in the iteration $t+1$, $\hat{x}_{t+1}^i$ is the solution in the scaled, rotated, and translated space (Spiegel, 1959; Wilke et al., 2007; Bonyadi and Michalewicz, 2014).

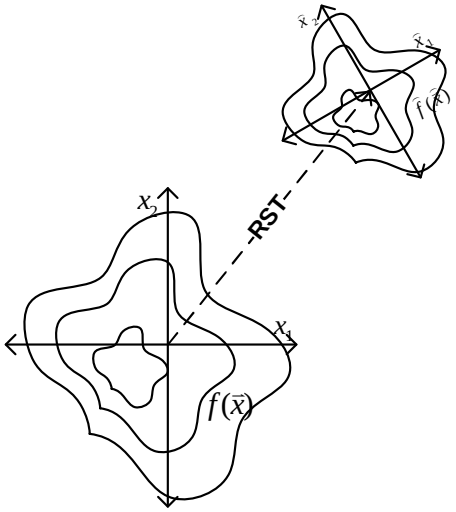


Figure 10: Rotation, scale, and translation (RST) of a search space

Among different transformations, sensitivity to rotation has received especial attention in optimization algorithms. The reason is that rotation usually causes the search space to become non-separable (Salomon, 1995). Hence, if an algorithm is not sensitive to rotation, it most likely performs better on non-separable search spaces.

In this section we investigate transformation (including scale, rotation, and translation) sensitivity of IPSO and SPSO2011.

5.1 Transformation sensitivity for IPSO

IPSO is scale and translation invariant while it is rotation variant (Wilke, 2005; Wilke et al., 2007). An algorithm is said to be rotation variant if its performance is change by rotating the search space. Note that, intuitively, the performance of algorithm should remain the same when the search space is rotated as the optimization problem remains unchanged. Wilke et al. (2007) proved that the linear PSO (a PSO which uses the same random value for all dimensions, i.e., all values on the diagonal of R_{1t} and R_{2t} are equal) is scale and rotation invariant (this variant is called LPSO). However, in LPSO, if $(\vec{p}_t^i - \vec{x}_t^i)||(\vec{g}_t - \vec{x}_t^i)$ and $\vec{V}_t^i||(\vec{p}_t^i - \vec{x}_t^i)$, the particle i starts oscillating between its personal best and the global best (Bonyadi et al., 2013), which is considered as a defect in the algorithm.

A new PSO variant was proposed in (Wilke et al., 2007) that was rotation invariant. The algorithm (called Rotation invariant PSO, RPSO) used random rotation matrices rather than random diagonal matrices to rotate the velocity vector. An approximation idea (an exponential map) was used for generating the rotation matrices to reduce the computational expenses arising from matrix multiplication. In the exponential map, a rotation matrix M is generated by:

$$M = I + \sum_{i=1}^{max_i} \frac{1}{i!} \left(\frac{\alpha\pi}{180} \left(E - E^T \right) \right)^i \tag{10}$$

where E is a $d \times d$ matrix with elements generated randomly in the interval [-0.5, 0.5], α is a real value representing the angle, and I is the identity matrix. The generated matrix M is a random rotation matrix with the angle α. It is clear that the matrix M becomes a more accurate estimation of a rotation matrix with the angle α when max_i is larger (see Eq. 10). The value of max_i was set to 1 Wilke et al. (2007), limiting the approximation to one term of the summation only. Thus, the approximation error of the random rotation matrix method grows with the rotation angle (α). Note that for larger values of max_i, the time complexity of generating rotation matrices grows rapidly. The time complexity for generating and multiplying the approximated rotation matrix (with $max_i = 1$) into a vector is in $O(d^2)$, i.e. including generating the approximated matrix ($O(d^2)$) plus vector-matrix multiplication ($O(d^2)$).

It was shown (Hansen et al., 2011) that IPSO performs poorly in comparison with Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) (Hansen and Ostermeier, 1996) and Differential Evolution (DE) (Storn and Price, 1997) when it is applied to non-separable optimization problems. Because of the fact that rotating a landscape makes an optimization problem non-separable (Salomon, 1995), IPSO performs also poorly when the landscape is rotated (it is rotation variant). The same comparison was also conducted in (Auger et al., 2009). That study confirmed that, although IPSO performs well in separable optimization problems, its performance in non-separable optimization problems is poor. The authors argued that the main reason behind this poor performance is that all calculations in IPSO are done for each dimension separately (Auger et al., 2009), which makes the algorithm rotation variant.

The impact of rotation matrices on the performance of PSO variants was studied by Bonyadi et al. (2014), where the random diagonal matrices in several PSO variants were replaced by Euclidean rotation matrices. The directions of rotations were generated by a normal distribution with mean of current direction of velocity and a variance that was set experimentally for different variants. Experiments showed that random rotation matrices can improve the performance of PSO variants in most cases on the tested benchmarks. The IPSO in which the rotation matrices are used is called RotPSO in this paper.

The rotation variance issue was also investigated by Bonyadi and Michalewicz (2014) from theoretical perspective (see Eq. 8). It was proven (Bonyadi and Michalewicz, 2014) that if $sQf(\vec{y}) + \vec{b} = f(sQ\vec{y} + \vec{b})$ is true for all scaler s, orthogonal matrix Q ($Q \in \mathbb{R}^d \times \mathbb{R}^d$), and vectors $\vec{b}$ and $\vec{y}$ ($\vec{b}, \vec{y} \in \mathbb{R}^d$), then the PSO variant that uses velocity update rule in Eq. 8 is scale, rotation, and translation invariant.

5.2 Transformation sensitivity in SPSO2011

A PSO algorithm is rotation, scale, and translation invariant (RST invariant) if $\hat{x}^i_{t+1} = sQ\vec{x}^i_{t+1} + \vec{b}$ for all $t > 0$, i.e., updating the position of particle i and then rotating, scaling, and translating the search space is the same as rotating, scaling, and translating the search space and then updating the position, for any scalar s, orthogonal matrix Q, and vector b, and $\hat{x}^i_{t+1}$ is the position vector in the rotated, scaled, and translated space (Spiegel, 1959; Wilke et al., 2007; Bonyadi and Michalewicz, 2014) at iteration $t + 1$. Let us calculate $sQ\vec{x}^i_{t+1} + \vec{b}$ for SPSO2011:

$$sQ\vec{x}^i_{t+1} + \vec{b} = sQ\left(\vec{x}^i_t + \omega\vec{V}^i_t + H\left(\vec{G}^i_t, \|\vec{G}^i_t - \vec{x}^i_t\|\right) - \vec{x}^i_t\right) + \vec{b} \tag{11}$$

It is clear that $\vec{V}^i_t = \vec{x}^i_t - \vec{x}^i_{t-1}$, thus Eq. 11 can be rewritten as

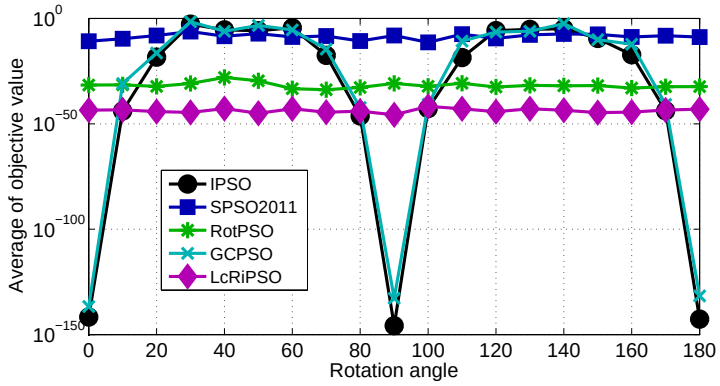


Figure 11: Application of five PSO varients to a two dimensional Ellipse function

$$sQ\vec{x}^i_{t+1} + \vec{b} = sQ\vec{x}^i_t + sQ\omega(\vec{x}^i_t + \vec{x}^i_{t-1}) + sQH\left(\vec{G}^i_t, \|\vec{G}^i_t - \vec{x}^i_t\|\right) - sQ\vec{x}^i_t + \vec{b} \quad (12)$$

Also

$$\hat{x}^i_{t+1} = \hat{x}^i_t + \hat{V}^i_{t+1} = sQ\vec{x}^i_t + \vec{b} + \omega\left(sQ\vec{x}^i_t - sQ\vec{x}^i_{t-1}\right) + \hat{H}(\vec{G}^i_t, \|\vec{G}^i_t - \vec{x}^i_t\|) - sQ\vec{x}^i_t - \vec{b} \quad (13)$$

Comparing Eq. 12 and 13 it is clear that in order to guarantee $\hat{x}^i_{t+1} = sQ\vec{x}^i_{t+1} + \vec{b}$ (and consequently guarantee that the algorithm is RST invariant), the following equality must be satisfied:

$$\hat{H}_i(\vec{G}^i_t, \|\vec{G}^i_t - \vec{x}^i_t\|) = sQH(\vec{G}^i_t, \|\vec{G}^i_t - \vec{x}^i_t\|) + \vec{b} \quad (14)$$

According to the definition of RST invariance, if $H(\vec{G}^i_t, \|\vec{G}^i_t - \vec{x}^i_t\|)$ is RST invariant then the equality in Eq. 14 is true. Also, it is clear that generating a random point in a hyper-sphere with a given radius and center ($H(.,.)$) and then rotating, scaling, and translating the search space is the same as rotating, scaling, and translating the search space and then generating a random point in a hyper-sphere with the same radius and the transformed center. Thus, the operator $H(.,.)$ is RST invariant, which implies that SPSO2011 is RST invariant.

Although the above proof is sufficient to guarantee rotation invariance of SPSO2011, in order to see the impact of this characteristic in different algorithms, we applied the algorithms (SPSO2011, IPSO, RotPSO, LcRiPSO, and GCPSO) to a benchmark problem designed in (Spears et al., 2010) for analyzing the rotation invariance of PSO variants. The benchmark is a two dimensional Ellipse that is rotated in the space (from 0 to 180 degree) and the algorithms are applied 20 times to find an optimum point in the search space (see Fig. 11).

According to Fig. 11 the performances of SPSO2011, LcRiPSO, and RotPSO are not significantly changed by rotating the search space with different angles. The slight changes in the performance of these methods are because of random components in the methods. However, performance of IPSO and GCPSO is substantially changed by

rotating the search space, which shows that IPSO and GCPSO are not rotation invariant in general case (this test case serves a counter example for the statement IPSO or GCPSO is rotation invariant).

6 Conclusions and future work

In this paper stability of particles, local convergence, and rotation sensitivity of SPSO2011 were investigated. In the stability analysis we proposed an experimental approach, called Estimate Convergence Boundary (ECB), which could estimate the convergence boundary (the boundary of coefficients for a PSO variant where the particles converge to a point in the search space) for PSO variants. ECB was applied to estimate the convergence boundary for a particular PSO variant (called IPSO) that its convergence boundary is known. Results indicated that the proposed ECB algorithm can estimate the convergence boundary for IPSO. ECB was also applied to estimate the convergence boundary for SPSO2011. Results showed that the convergence boundary for SPSO2011 is different from that of IPSO. Also it was shown that, unlike IPSO, the convergence boundary for SPSO2011 is changed when the number of dimensions changes. Further experiments showed that picking previous frequently-used coefficients for PSO variants is not necessarily a good choice for SPSO2011. In terms of local convergence it was proven that SPSO2011 is not able to guarantee locating a local optimum in the search space. We proved that any algorithm (PSO variant) that follows the form of $\vec{x}_{t+1} = \omega(\vec{x}_t - \vec{x}_{t-1}) + n_{h_1,h_2}(\vec{p}_t, \vec{x}_t)$, where $\vec{p}_t = D(\vec{p}_{t-1}, \vec{x}_t)$, is locally convergent for all $\omega \in (0,1), \vec{x}_0, \vec{p}_0 \in S$ (Lemma 3). This lemma is applicable to study local convergence property of a large class of PSO variants. We modified SPSO2011 in a way that this condition is satisfied so that the modified SPSO2011 is locally convergent. Finally, we proved that SPSO2011 is rotation, scale, and translation invariant. Theoretical analysis of stability of particles in SPSO2011 is a potential future work in this area of research.

7 Acknowledgment

This work was partially funded by the ARC Discovery Grant DP130104395 and by grant N N519 5788038 from the Polish Ministry of Science and Higher Education (MNiSW).

References

Auger, A., Hansen, N., Zerpa, J. P., Ros, R., and Schoenauer, M. (2009). Empirical comparisons of several derivative free optimization algorithms. In *Acte du 9ieme colloque national en calcul des structures (Proceedings of the 9th National Symposium in calculation of structures)*, volume 1.

Baba, N. (1981). Convergence of a random optimization method for constrained optimization problems. *Journal of Optimization Theory and Applications*, 33(4):451–461.

Bonyadi, M. R., Li, X., and Michalewicz, Z. (2013). A hybrid particle swarm with velocity mutation for constraint optimization problems. In *Genetic and Evolutionary Computation Conference*, pages 1–8. ACM.

Bonyadi, M. R. and Michalewicz, Z. (2014). A locally convergent rotationally invariant particle swarm optimization algorithm. *Swarm Intelligence*, 8(3):159–198.

Bonyadi, M. R., Michalewicz, Z., and Li, X. (2014). An analysis of the velocity updating rule of the particle swarm optimization algorithm. *Journal of Heuristics*, 20(4):417–452.

Bratton, D. and Kennedy, J. (2007). Defining a standard for particle swarm optimization. In *Swarm Intelligence Symposium*, pages 120–127. IEEE.

Campana, E., Fasano, G., and Pinto, A. (2010). Dynamic analysis for the selection of parameters and initial population, in particle swarm optimization. *Journal of Global Optimization*, 48(3):347–397.

Clerc, M. (2006). *Particle swarm optimization*. Wiley-ISTE.

Clerc, M. (2011). Standard particle swarm optimisation from 2006 to 2011.

Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73.

Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *International Symposium on Micro Machine and Human Science*, pages 39–43. IEEE.

Engelbrecht, A. (2005). *Fundamentals of computational swarm intelligence*. Wiley, Hoboken, NJ.

Hansen, N. and Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Congress on Evolutionary Computation*, pages 312–317. IEEE.

Hansen, N., Ros, R., Mauny, N., Schoenauer, M., and Auger, A. (2011). Impacts of invariance in search: When cma-es and pso face ill-conditioned and non-separable problems. *Applied Soft Computing*, 11(8):5755–5769.

Jiang, M., Luo, Y. P., and Yang, S. Y. (2007). Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters*, 102(1):8–16.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE.

Lehre, P. K. and Witt, C. (2013). *Finite first hitting time versus stochastic convergence in particle swarm optimisation*, pages 1–20. Springer.

Matyas, J. (1965). Random optimization. *Automation and Remote Control*, 26(2):246–253.

Montes de Oca, M. A., Sttzle, T., Birattari, M., and Dorigo, M. (2009). Frankenstein's pso: A composite particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 13(5):1120–1132.

PCS (2006). Standard particle swarm optimisation.

Poli, R. (2008). Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artifficial Evolution and Application*, 2008(3):1–10.

Poli, R. (2009). Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *IEEE Transactions on Evolutionary Computation*, 13(4):712–721.

Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization: an overview. *Swarm Intelligence*, 1(1):33–57.

Salomon, R. (1995). Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions - a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39:263–278.

Schmitt, M. and Wanka, R. (2013). Particle swarm optimization almost surely finds local optima. In *Genetic and Evolutionary Computation Conference*, pages 1629–1636. ACM.

Shi, Y. and Eberhart, R. (1998). A modified particle swarm optimizer. In *World Congress on Computational Intelligence*, pages 69–73. IEEE.

Solis, F. J. and Wets, R. J.-B. (1981). Minimization by random search techniques. *Mathematics of Operations Research*, 6(1):19–30.

Spears, W., Green, D., and Spears, D. (2010). Biases in particle swarm optimization. *International Journal of Swarm Intelligence Research*, 1(2):34–57.

Spiegel, M. R. (1959). *Theory and problems of vector analysis*. Schaum's Outline Series.

Storn, R. and Price, K. (1997). Differential evolutiona simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359.

Trelea, I. C. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, 85(6):317–325.

Van den Bergh, F. and Engelbrecht, A. (2010). A convergence proof for the particle swarm optimiser. *Fundamenta Informaticae*, 105(4):341–374.

Van den Bergh, F. and Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information Sciences*, 176(8):937–971.

Wilke, D. (2005). *Analysis of the particle swarm optimization algorithm*. PhD thesis.

Wilke, D., Kok, S., and Groenwold, A. (2007). Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on scale and frame invariance. *International journal for numerical methods in engineering*, 70(8):985–1008.

Zambrano-Bigiarini, M., Clerc, M., and Rojas, R. (2013). Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In *Congress on Evolutionary Computation*, pages 2337–2344. IEEE.

4 Application of PSO to continuous space COPs

This chapter includes 4 subsections; each of these subsections contains one paper related to application of PSO to COPs.

4.1 PSO for COPs

The article included in this subsection studies application of PSO to constrained optimization problems (COPs).

M. R. Bonyadi, X. Li, and Z. Michalewicz, "A hybrid particle swarm with velocity mutation for constraint optimization problems," in *Genetic and Evolutionary Computation Conference*, ACM, Amsterdam, The Netherlands, 2013, pp. 1-8

Bonyadi, M.R., Li, X. & Michalewicz, Z. (2013) A hybrid particle swarm with velocity mutation for constraint optimization problems.
Presented at: Genetic and Evolutionary Computation Conference, 6-10 July, Amsterdam, The Netherlands

NOTE:
This publication is included on pages 162-169 in the print copy of the thesis held in the University of Adelaide Library.

It is also available online to authorised users at:

http://doi.org/10.1145/2463372.2463378

4.2 Disjoint feasible ragions in a COP

Two articles included in this subsection. The first article, entitled “Locating potentially disjoint feasible regions of a search space with a particle swarm optimizer” studies application of PSO to locate disjoint feasible regions of a COP. The second article, entitled “A hybrid particle swarm with a time-adaptive topology for constrained optimization”, extends the idea of locating disjoint feasible regions to find optimal solutions in a COP.

4.2.1 Locating Disjoint feasible ragions in a COP

The paper included in this subsection studies application of PSO to locate disjoint feasible regions of a COP.

M. R. Bonyadi and Z. Michalewicz, "Locating potentially disjoint feasible regions of a search space with a particle swarm optimizer," in *Evolutionary Constrained Optimization*, K. Deb, Ed., ed: Springer-Verlag, 2014, pp. 205-230

Bonyadi, M.R. & Michalewicz, Z. (2015) Locating potentially disjoint feasible regions of a search space with a particle swarm optimizer.
In: Evolutionary Constrained Optimization, pp. 205-230

NOTE:
This publication is included on pages 171-195 in the print copy of the thesis held in the University of Adelaide Library.

It is also available online to authorised users at:

http://doi.org/10.1007/978-81-322-2184-5_8

4.2.2 An adptive topology for PSO to address COPs

The article included in this section extends the idea of locating disjoint feasible regions to find optimal solutions in a COP.

M. R. Bonyadi, X. Li, and Z. Michalewicz, "A hybrid particle swarm with a time-adaptive topology for constrained optimization," *Swarm and Evolutionary Computation*,vol. 18, pp. 22-37, 2014

Bonyadi, M.R., Li, X. & Michalewicz, Z. (2014) A hybrid particle swarm with a time-adaptive topology for constrained optimization.
Swarm and Evolutionary Computation, v. 18(October) , pp. 22-37

NOTE:
This publication is included on pages 197-212 in the print copy of the thesis held in the University of Adelaide Library.

It is also available online to authorised users at:

http://doi.org/10.1016/j.swevo.2014.06.001

4.3 PSO on the edge of feasiblity

The article included in this subsection studies searching the edges of feasibility and how it can be conducted through mathematical formulations in the context of PSO.

M. R. Bonyadi and Z. Michalewicz, "On the edge of feasibility: a case study of the particle swarm optimizer," in *Congress on Evolutionary Computation*, IEEE, Beijing, China, 2014, pp. 3059-3066

Bonyadi, M.R. & Michalewicz, Z. (2014) On the edge of feasibility: a case study of the particle swarm optimizer.
Presented at: IEEE Congress on Evolutionary Computation (CEC), 6-11 July, Beijing, China, pp. 3059-3066.

NOTE:
This publication is included on pages 214-221 in the print copy of the thesis held in the University of Adelaide Library.

It is also available online to authorised users at:

http://doi.org/10.1109/CEC.2014.6900343

4.4 Application of PSO to discrete space COPs

The article included in this subsection studies a coding scheme for a discrete space COP that maps the problem into a continuous space COP. PSO is used to deal with constraints and find solutions.

M. R. Bonyadi and Z. Michalewicz, "A fast particle swarm optimization algorithm for the multidimensional knapsack problem," in *Congress on Evolutionary Computation*, IEEE, 2012, pp. 1-8

Bonyadi, M.R. & Michalewicz, Z. (2012) A fast particle swarm optimization algorithm for the multidimensional knapsack problem.
Presented at: IEEE Congress on Computational Intelligence, 10-15 June, Brisbane, Australia, pp. 1-8.

NOTE:
This publication is included on pages 223-231 in the print copy of the thesis held in the University of Adelaide Library.

It is also available online to authorised users at:

http://doi.org/10.1109/CEC.2012.6256113

5 Conclusions

In this thesis, the Particle Swarm Optimization (PSO) algorithm was investigated from theoretical and application perspectives. Theoretical analysis included investigation on transformation sensitivity, stability, and local convergence properties of the original algorithm. We found that the usage of rotation matrices in the formulation of many PSO variants not only guarantees rotation invariance of the algorithm but also improves their overall performance. We also defined sufficient conditions to guarantee local convergence and rotation invariance for a general formulation of topology of the original PSO. We further generalized the formulation so that it covers a large class of PSO variants and we identified sufficient condition of local convergence for this formulation. This can be useful to examine whether a PSO variant is locally convergent. For a specific PSO variant, we also conducted stability analysis and transformation sensitivity. Moreover, we analyzed the behavior of particles in a PSO variant and proposed an experimental approach based on frequency domain analysis to predict the type of oscillation of particles before convergence according to the coefficient values. This can assist researchers to pick the best coefficient values for the problem at hand.

From application perspective, we investigated different aspects of the algorithm to deal with constrained optimization problems (COPs). We defined an important direction of future research in COPs, namely locating disjoint feasible regions, and we proposed a PSO variant to deal with this problem. We used niching techniques and mapped the problem of locating disjoint feasible regions into finding niches in a search space. We extended that PSO variant in a way that not only the algorithm was able to locate disjoint feasible regions, but it could also concentrate the search on the most promising region so that better feasible solutions are found at the end of the search. In addition, we formally formulated edge of feasibility and proposed a formulation to restrict the search to only these edges with defined thicknesses. We applied a PSO variant and showed that this methodology is effective for solving COPs which their optimal solutions are on the edges of feasibility. Further, we encoded a discrete COP to the form of a continuous COP and applied a PSO variant to find optimum solutions. Results showed that the coding scheme and the proposed PSO method can solve the problem faster than existing algorithms.

Appendix

This appendix contains a paper that was related to theoretical analysis of PSO. The paper received the best paper award in the Genetic and Evolutionary Computation Conference, GECCO, in the track ACSI. It was invited to be published in the Evolutionary Computation journal. Thus, we included the extended verision of this paper to section 3.3 and we provide the conference version here in this appendix.

M. R. Bonyadi and Z. Michalewicz, "SPSO2011 – Analysis of stability, local convergence, and rotation sensitivity," in *Genetic and Evolutionary Computation Conference*, ACM, Vancouver, Canada, 2014, pp. 9-16 (**Received the best paper award in the track ACSI**)

Bonyadi, M.R. & Michalewicz, Z. (2011) SPSO2011 - Analysis of stability, local convergence and rotation sensitivity.
Presented at: Genetic and Evolutionary Computation Conference, 12-16 July, Vancouver, Canada, pp. 9-16.

NOTE:
This publication is included on pages 232-239 in the print copy of the thesis held in the University of Adelaide Library.

It is also available online to authorised users at:

http://doi.org/10.1145/2576768.2598263