This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2014.2371334, IEEE Transactions on Industrial Informatics

Notes on feasibility and optimality conditions of small-scale multi-function robotic cell scheduling problems with pick up restrictions

Mehdi Foumani, Indra Gunawan, Kate Smith-Miles, *Senior Member, IEEE*, and M. Yousef Ibrahim, *Senior Member, IEEE*

***Abstract*— Optimization of robotic workcells is a growing concern in automated manufacturing systems. This study develops a methodology to maximize the production rate of a multi-function robot (MFR) operating within a rotationally arranged robotic cell. A MFR is able to perform additional special operations while in transit between transferring parts from adjacent processing stages. Considering the free-pick up scenario, the cycle time formulas are initially developed for small-scale cells where a MFR interacts with either two or three machines. A methodology for finding the optimality regions of all possible permutations is presented. The results are then extended to the no-wait pick up scenario in which all parts must be processed from the input hopper to the output hopper, without any interruption either on or between machines. This analysis enables insightful evaluation of the productivity improvements of MFRs in real-life robotized workcells.**

***Index Terms*— Automated Manufacturing Systems, Cyclic Scheduling, Robotic Cells, Multi-function, No-wait**

I. Introduction

Today's automated systems predominantly incorporate material handling robots interacting well with other equipment such as computer numerical control (CNC) machines, and automated storage and retrieval systems in the production line [1]. Any savings in robot movement time enhances the competitiveness of world class companies. Two classes of problem are Single-Function Robotic Cell (SFRC) and Multi-Function Robotic Cell (MFRC) scheduling problems, where determining a cyclic robot move sequence which yields the highest throughput gain is critical to success.

The first problem, which addresses a manufacturing cell equipped with a pick-and-place robot to perform a single task, is common in practice [2]. This kind of transporting robot is usually called a Single-Function Robot (SFR). For the second problem, the cell is served by a Multi-Function Robot (MFR), which concurrently performs an arbitrary task in addition to part transportation tasks. One of the most recent industrial developments is the use of these MFRs in manufacturing cells.

As an instance of MFRs, the application of Grip-Gage-Go (GGG) grippers performing in-process control as its additional task has become popular in manufacturing cells recently. The grippers, installed at the end of a MFR arm, perform quality control (e.g. accurately measure diameters) while carrying a part to the next machine. Fig. 1 shows an example of these grippers used for measuring the diameter of a crankshaft [3]. The measuring heads are integrated into the automation by adding gages and crankshaft locating features to MFRs [4]. Here, we present a detailed study regarding GGG grippers.

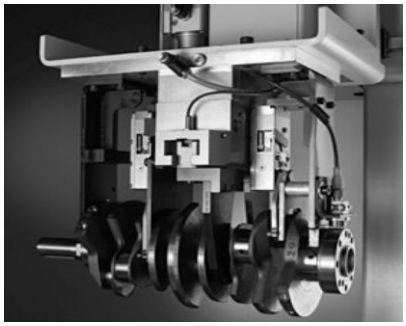

Fig. 1. Measurement of crankshaft diameters in transit [3]

Because a gripper is an independent tool at the end of a robot's mechanical arm which can adapt to various production environments, the GGG gripper can be attached to a wide range of robots. A simple example of this is depicted in Fig. 2 where a GGG gripper is added to the arm of Fanuc M-710iB/45 Robot. Hence, the Fanuc M-710iB/45 Robot can measure the thickness of shaft in transit between machines [3].

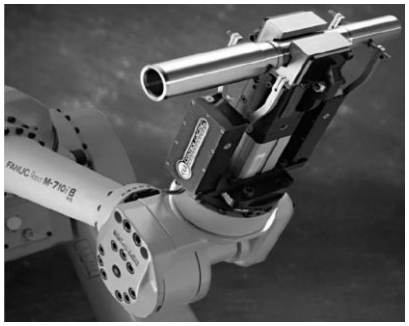

Fig. 2. The arm of Fanuc M-710iB/45 Robot equipped by a GGG gripper [3]

M. Foumani is with the School of Applied Sciences & Engineering, Faculty of Science, Monash University, Melbourne, Australia (e-mail: mehdi.foumani@monash.edu).

I. Gunawan and M. Y. Ibrahim are with the School of Engineering & Information Technology, Faculty of Science & Technology, Federation University Australia, Gippsland Campus, Churchill, VIC 3842, Australia (e-mail: indra.gunawan@federation.edu.au; yousef.ibrahim@federation.edu.au).

K. Smith-Miles is with the School of Mathematical Sciences, Monash University, Clayton, Victoria 3800, Australia (e-mail: kate.smith-miles@monash.edu).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2014.2371334, IEEE Transactions on Industrial Informatics

A SFRC is generally composed of two machines M_1 and M_2 or three machines M_1, M_2, and M_3. A stationary base SFR rotating on its axis is used in this robotized shop to transfer parts from each machine to the next, and between machines and a joint input/output hopper *I*/*O*. Any arbitrary machine M_j placed in the cell performs operation O_j with the corresponding processing time P_j [5].

Fig. 3a. Two-machine SFRCs with rotational layout

Fig. 3b. Three-machine SFRCs with rotational layout [6]

Fig. 3a and 3b show real-life applications of two- and three-machine SFRCs at Haas Automation Incorporation. Physically, one SFR is assigned to each cell to avoid collisions. In these manufacturing cells, a SFR is in charge of picking up a part from *I*/*O*, loading it on CNC machine M_1 to be processed, transferring it through other machines and eventually dropping off this part at *I*/*O* where both the raw material and completed parts are stored. Two scenarios for unloading the part can be considered as soon as the part's operation on a machine is completed. Under the free-pick up scenario, which is the predominant type in real-world cells, the part can stay indefinitely on the machine waiting for the SFR. However, under the no-wait pick up scenario, which is stricter, the part must be unloaded from the machine without delay and then carried to the down-stream machine. Consequently, the SFR must reach the machine on time.

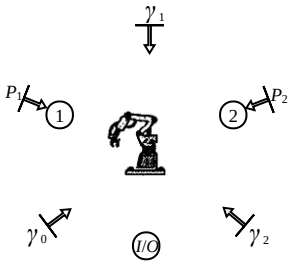

Fig. 4a. Two-machine MFRCs with rotational layout

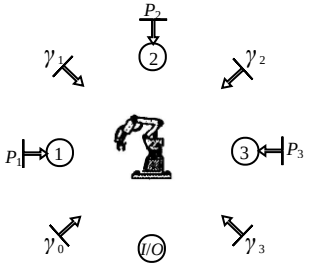

Fig. 4b. Three-machine MFRCs with rotational layout

Fig. 4a and 4b show two- and three-machine rotationally-arranged MFRCs in which Γ_j represents the robots' operation while in transit between transferring parts from M_j to M_{j+1}. Also, γ_j denotes the processing time required by the robot to perform Γ_j. In Fig. 4a and 4b, a single MFR is in charge of moving the parts through $\Gamma_0 \rightarrow O_1 \rightarrow \Gamma_1 \rightarrow O_2 \rightarrow \Gamma_2$ and $\Gamma_0 \rightarrow O_1 \rightarrow \Gamma_1 \rightarrow O_2 \rightarrow \Gamma_2 \rightarrow O_3 \rightarrow \Gamma_3$, respectively. In fact, the MFR is also responsible for performing processes $\{\Gamma_0, \Gamma_1, \Gamma_2\}$ and $\{\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3\}$ in transit, respectively. The time taken to perform these operations can be shown as $\{\gamma_0, \gamma_1, \gamma_2\}$ and $\{\gamma_0, \gamma_1, \gamma_2, \gamma_3\}$. The goal of this paper is to find a periodic MFR's task set that satisfies both the timing and other constraints [7].

Thus, the rest of the paper is organized as follows. After presenting a brief literature review in Section II, the authors describe the problem definitions and notation in Section III. Section IV is dedicated to find the optimal permutation in MFRCs with the free-pick up scenario. Similar analysis for a MFRC with the no-wait pick up scenario is conducted in Section V to find an optimal permutation if residency time is restricted. Section VI is devoted to the conclusions and discussion of future work.

II. Related Research

Considering the free-pick up scenario, Sethi et al. [8] presented a case study of two- and three SFRCs which performed drilling and boring operations on twenty pound castings to be used in truck differential assemblies. They succeeded in optimizing the production lines adapted from PRAB Robotic Company. Shortly afterwards, Sethi et al. [9] focused on analyzing a class of two- and three-machine SFRCs served by two-unit SFRs. Other studies have also addressed multiple part-types, for example, scheduling multiple part-types in a dual-gripper robot cell was addressed in [10]. The developed algorithm in [10] was only able to achieve a near optimal permutation with the worst-case performance ratio of 3/2. Note that a linear programming approach was employed in their research to compute the performance ratio without finding a lower bound.

Considering a case study in metal cutting industries, [11] established a unified notational and modelling structure to optimize two- and three-machine flexible SFRCs. They defined a flexible SFRC as the combination of a flexible manufacturing system (FMS) with a flow shop. Then, they derived the highest performance which could be obtained by changing the assignment of operations to production machines. Furthermore, an enumerative technique was applied for finding the worst-case performance ratio similar to [10]. This worst case performance was $14\frac{2}{7}\%$ for the three-machine case, which means the maximum productivity increase of using a flexible SFRC instead of inflexible was $14\frac{2}{7}\%$. Also, Nambiar and Judd [12] used max-plus algebra as a tool to develop a mathematical model for cyclic production lines. The newly-modeled max-plus formulation was able to facilitate the calculation of cycle time. In fact, it was used as the underlying mechanism to calculate cycle time precisely when an improvement heuristic algorithm such as Tabu Search (TS) or Genetic Algorithm (GA) was used to search for the optimal (or near-optimal) permutation. Subsequently, a reentrant SFRC that combined two machines with a SFR in a closed environment was optimized in [13]. The employed SFR with temporary buffer had the ability to swap a part on a busy machine with a part on a busy SFR. The regions of optimality of all permutations were presented in [13] after performing a comparative analysis.

The no-wait pick up scenario is more suitable for real-life scheduling problems than other simplified scenarios. In this regard, Agnetis [14] established polynomial algorithms for scheduling of two- and three-machine SFRCs. Also, Paula et

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2014.2371334, IEEE Transactions on Industrial Informatics

al. [15] developed a heuristic for a scheduling problem of a SFR used by an aircraft manufacturer with the surface treatment of component parts attached to both wings of transport aircrafts. Afterwards, Alcaide et al. [16] took into account a scheduling problem appearing in the electroplating line, and established a graph model of operations for this small-scale SFRC with no-wait scenario. The SFR used in this automated cell was a part of the computer-integrated manufacturing system CIM-2000 Mechatronics manufactured by DEGEM Systems Company. A real-life radar scheduling problem, which is equivalent to single machine SFRC with no-wait pick up scenario, was studied in [17]. They proved a radar system can be simulated by a no-wait SFRC due to the fact that the first task is a wave transmission and the second task is reflected wave receiving without delay.

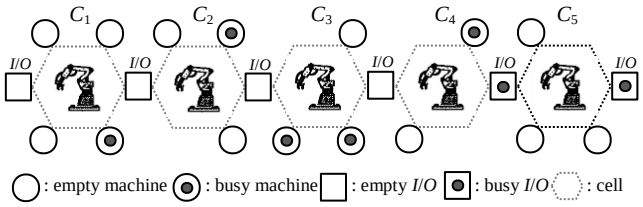

Fig. 5. A clustering system for connection between five small-scale MFRCs.

A few recent papers are closely related to MFRCs with the free-pick up scenario. For MFRCs where the MFR performs both printing and milling operations to supply large printed foam structures, an optimal schedule is generated in [18]. A MFR transferring the part between two adjacent processing stages and simultaneously performing an inspection operation in this transit was introduced for the first time in [19]. They considered the restricted model of the linearly-configured MFRCs producing identical parts, and only compared the performance of these MFRCs with SFRCs. The proposed approach for this MFRC involved deriving the lower bound of cycle time, and then finding some permutations with the cycle time as close as possible to this lower bound. It is known that the number of feasible permutations for a MFRC with k machines is $k!$, whereas the research by [19] was only restricted to studying two permutations. As a consequence, the results from [19] could not be fully beneficial to MFRCs throughput analysis. Following that, Foumani et al. [20] considered rotational MFRCs instead of in-line ones and discussed some results for replacing related MFRCs with SFRCs. Similar to [19], the parameter values for which only two special permutations are optimal were determined. As a consequence, once again, the analysis was not complete and the impact for the remaining feasible region was not analyzed. Therefore, it is vital to develop a detailed analysis that fully covers all feasible regions, especially for two- and three-machine MFRCs.

The approach proposed in this paper determines the regions of optimality of all permutations and performs a comparative analysis after computing their cycle time. When the part processing routes in MFRCs are complicated, one of the most economic strategies is breaking these MFRCs into small-scale clusters. A MFR serves within one cluster consisting of two or three machines [21]. Fig. 5 provides an example of converting a 15-machine semiconductor production line into five MFRCs. From the left side to the right side, we have four, three, three, two, and three-machine MFRCs. At first, parts must enter to the system from left-side *I*/*O* and then pass through cells C_1, C_2, C_3, C_4, and C_5. Finally, the part is stored at the right-side *I*/*O*. This paper also extends the results to the no-wait pick up scenario to consider more realistic conditions.

The most important contribution of this paper is to provide managerial insights into the advantages that can be achieved by applying MFRs for small-scale cells. In more detail, the novelty of this study is developing a methodology to maximize the production rate of MFRCs under both the free and no-wait pick up scenarios. For all possible combinations of parameters, the feasibility and optimality regions of all permutations are presented. This research will provide a bridge between academic research on MFRCs and relevant real-world problems.

III. Problem Notation and Definitions

Compact SFRCs generally restrict intermediate hoppers, and consequently blocking or delay may happen. Scheduling MFR movement is also not deadlock-free and this results in the following operational restrictions: *The receiving device (MFR or anyone of the machines) and sending device (MFR or anyone of the machines) must be empty and loaded before the load/unload process, respectively* [20]. When the pick-up scenario is no-wait, there is also an additional feasibility constraint: *unloading the machine by the MFR with delay is not permitted*. MFR is subjected to two types of waits, full and partial waits, in keeping with these constraints. In fact, after loading a part on a machine, MFR either stays on this machine until the end of the operation or moves to the next machine to remove a part [22]. The MFRC scheduling is expressed extending the notations and definitions below from [4]:

ε The load (or unload) time of machines by MFR
δ The time taken by empty MFR to travel from M_i to M_{i+1}
S^{i}_{jmf} The j^{th} permutation of a MFRCs in which i, m and f denote the number of machines, multi-functionality and free-pick up scenario, respectively
$T_{S^{i}_{jmf}}$ The cycle time of S^{i}_{jmf}
S^{i}_{jmw} The j^{th} permutation in which i, m and f denote the number of machines, multi-functionality and no-wait pick up
$T_{S^{i}_{jmw}}$ The cycle time of S^{i}_{jmw}
P_l The processing time of M_l dominating all machines as $\beta_{l-1}+P_l+\beta_l \geq \beta_{i-1}+P_i+\beta_i$
w_i MFR's waiting time at M_i for free- and no-wait scenarios

Definition 1. Having a MFR, $2\varepsilon+max\{\gamma_i,\delta\}$ is the time elapsed of an *activity* A_i, $\forall i$ =[0, 1, 2, 3], with sequence: 1) Empty MFR unloads a part from busy M_i. 2) MFR carries this part to M_{i+1}. 3) Busy MFR loads this part onto empty M_{i+1}.

We know two cases may occur when MFR performs activity A_i: 1) $\gamma_i \leq \delta$: this means MFR finishes the operation before arriving at M_{i+1}. Therefore, it loads the part to M_{i+1} as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2014.2371334, IEEE Transactions on Industrial Informatics

soon as the transfer of the part is finished, which totally takes δ. 2) $\gamma_i>\delta$: in contrast to previous case, MFR finishes the operation after arriving at M_{i+1}: thus, MFR stops in front of M_{i+1} to finish the operation and then loads the part to the machine. This takes γ_i time unit. Hence, as mentioned in Definition 1, the time taken by busy MFR to perform activity A_i is a couple of load/unload operations plus the *max* term of these two values: $2\varepsilon+max\{\gamma_i,\delta\}$. Note SFRC is a simplified subdivision of MFRC if $\gamma_i=0$, $\forall i\in[0,1, 2, 3]$. For simplicity, hereinafter β_i is used instead of $max\{\gamma_i,\delta\}$. The definition below deriving from [8] is applicable to MFRCs as well.

Definition 2. Having a MFR in the cell, a permutation of all activities in which one finished parts are dropped at *I*/*O* in each implementation is called a one-unit permutation.

These permutations are referred to as *one-unit* since each A_i occurs once. Note one-unit permutations are actually the easiest to understand, implement and also control in comparison to other permutations [9]. Also, focusing on one-unit permutations gives us insight into the behavior of complex permutations [10]. Hence, this study is restricted to one-unit permutations. It is also assumed that the empty and occupied machines of each permutation are specified in advance since this permutation must meet the steady state cyclic requirement following from [11].

Definition 3. Having a one-unit permutation starting with A_0, activity A_i is a *pushed* (*pulled*) *activity* if A_{i-1} is completed before (after) it. The *pushed* (*pulled*) *activity* A_i implies M_i is empty (occupied) at the starting stage of the permutation.

It should be noted A_{i-1} is A_2 and A_3 when i=0 for two- and three-machine MFRCs. For example, A_1,A_2 are pushed and A_3 is pulled for permutation A_0,A_3,A_1,A_2 of three-machine case. This means M_1, M_2 are empty and M_3 is busy before starting it.

IV. Free-pick up scenario

In essence, the robot with multi-functionality never results in increasing the number of permutations. Actually, $S^2_{1mf}=A_0$, A_1, A_2 and $S^2_{2mf}=A_0$, A_2, A_1 represent permutations which can be occur for MFRCs with two production machines. Note the regions of optimality for both S^2_{1mf} and S^2_{2mf} should be obtained later than reformulating the cycle time of these permutations. S^2_{1mf} only has pushed activities resulting in full stop on M_1 and M_2. This means that $T_S{}^2_{1mf}$ is made up the following independent portions: six load/unload operations, three dextrorotary and occupied MFR rotations, and two full waiting. So, $T_S{}^2_{1mf}= 6\varepsilon+\sum_{i=0}^{2}\beta_i+P_1+P_2$. Regarding S^2_{2mf}, MFR picks up an unprocessed part from *I*/*O* and loads it to M_1 ($2\varepsilon+\beta_0$). Then, based on the activity's route described above, MFR removes the previous part from M_2 and drops it at *I*/*O* after an empty rotation from M_1 to M_2 and a partial stop on M_2 ($\delta+w_2+2\varepsilon+\beta_2$). Likewise, the empty MFR comes back M_1, waits on M_1, unloads the part, and loads it on M_2 ($\delta+w_1+2\varepsilon+\beta_1$), and returns to *I*/*O* (δ). So, $T_S{}^2_{2mf}$ consists of six load/unload, three empty MFR rotations, three busy MFR rotations, and two partial stops: $w_1=max\{0,P_1-(2\varepsilon+2\delta+\beta_2+w_2)\}$ and $w_2= max\{0, P_2-(2\varepsilon+2\delta+\beta_0)\}$. Because the summation of stops is $max\{0, P_1-(2\varepsilon+2\delta+\beta_2), P_2-(2\varepsilon+2\delta+\beta_0)\}$, we conclude $T_S{}^2_{2mf}=6\varepsilon+3\delta+\sum_{i=0}^{2}\beta_i+max\{0,P_1-(2\varepsilon+2\delta+\beta_2), P_2-(2\varepsilon+2\delta+\beta_0)\}$.

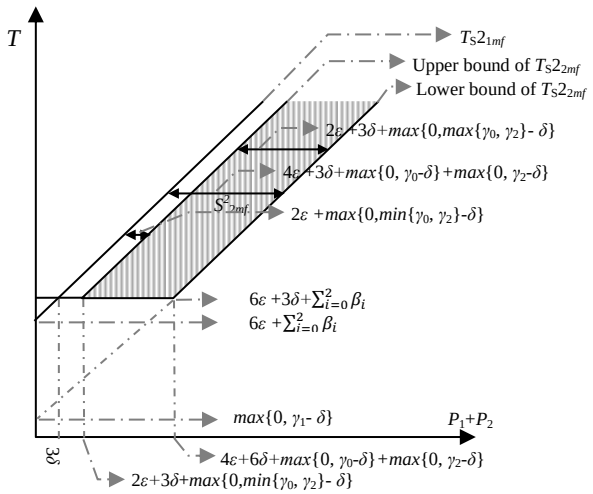

Fig. 6. The lower and upper bound of $T_S{}^2_{1mf} \le T_S{}^2_{2mf}$

Fig. 6 above illustrates the optimality regions of S^2_{1mf} and S^2_{2mf} by applying their cycle times. Obviously, S^2_{1mf} is optimal when $P_1+P_2\le 3\delta$, and S^2_{2mf} is optimal in the rest of feasible area. Optimizing three-machine MFRCs are complicated in comparison with two-machine ones because the number of permutations grows from two to six permutations below:

$S^3_{1mf}=A_0,A_1,A_2,A_3$ $S^3_{2mf}=A_0,A_2,A_1,A_3$ $S^3_{3mf}=A_0,A_1,A_3,A_2$
$S^3_{4mf}=A_0,A_3,A_1,A_2$ $S^3_{5mf}=A_0,A_2,A_3,A_1$ $S^3_{6mf}=A_0,A_3,A_2,A_1$

We name S^3_{1mf} and S^3_{6mf} uphill and downhill permutations, and the rest of permutations rolling hill permutations. For the sake of simplicity, the cycle time of two the most complex ones, S^3_{2mf} and S^3_{6mf}, are calculated, and then the rest of cycle times are shown in this section. Since $S^3_{2mf}=A_0$, A_2, A_1, A_3, total load/unload time, empty MFR rotation, busy MFR rotation, and partial waiting times is $8\varepsilon+4\delta+\sum_{i=0}^{3}\beta_i+\sum_{i=0}^{3}w_i$. Clearly, $8\varepsilon+4\delta+\sum_{i=0}^{3}\beta_i$ is a constant value, whereas w_1, w_2, w_3 are variable values below: $w_1= max\{0, P_1-(2\varepsilon+3\delta+\beta_2+w_2)\}$, $w_2= max\{0, P_2-(4\varepsilon+2\delta+\beta_0+\beta_3+w_3)\}$, $w_3= max\{0, P_3-(2\varepsilon+3\delta+\beta_1+w_1)\}$. Each one of waiting times w_1, w_2, w_3 can be zero or nonzero meaning there are eight subdivisions as follows:

- $w_1=0,w_2=0,w_3=0\rightarrow\sum_{i=0}^{3}w_i=0$
- $w_1=0\rightarrow w_3=max\{0, P_3-(2\varepsilon+3\delta+\beta_1)\}$
  $\sum_{i=0}^{3}w_i=max\{0,P_2-(4\varepsilon+2\delta+\beta_0+\beta_3),P_3-(2\varepsilon+3\delta+\beta_1)\}$
- $w_2=0\rightarrow w_1=max\{0, P_1-(2\varepsilon+3\delta+\beta_2)\}$
  $\sum_{i=0}^{3}w_i=max\{0,P_1-(2\varepsilon+3\delta+\beta_2),P_3-(2\varepsilon+3\delta+\beta_1)\}$
- $w_3=0\rightarrow w_2=max\{0, P_2-(4\varepsilon+2\delta+\beta_0+\beta_3)\}$
  $\sum_{i=0}^{3}w_i=max\{0,P_1-(2\varepsilon+3\delta+\beta_2),P_2-(4\varepsilon+2\delta+\beta_0+\beta_3)\}$
- $w_1=0,w_2=0\rightarrow\sum_{i=0}^{3}w_i= w_3= max\{0, P_3-(2\varepsilon+3\delta+\beta_1)\}$
- $w_1=0,w_3=0\rightarrow\sum_{i=0}^{3}w_i=w_2=max\{0,P_2-(4\varepsilon+2\delta+\beta_0+\beta_3)\}$
- $w_2=0,w_3=0\rightarrow\sum_{i=0}^{3}w_i= w_1= max\{0, P_1-(2\varepsilon+3\delta+\beta_2)\}$
- $w_1\ne0,w_2\ne0,w_3\ne0$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2014.2371334, IEEE Transactions on Industrial Informatics

It is easy to calculate all combinations, excluding the last one. The simplex method is applied for computation of the last $\sum_{i=0}^{3} w_i$. Assuming $A=P_1-(2\varepsilon+3\delta+\beta_2)$, $B=P_2-(4\varepsilon+2\delta+\beta_0+\beta_3)$, and $C=P_3-(2\varepsilon+3\delta+\beta_1)$, we rewrite w_1, w_2, and w_3 as:

$w_1 \geq 0,\ w_1 \geq P_1-(2\varepsilon+3\delta+\beta_2+w_2) \rightarrow w_1+w_2 \geq A$
$w_2 \geq 0,\ w_2 \geq P_2-(4\varepsilon+2\delta+\beta_0+\beta_3+w_3) \rightarrow w_2+w_3 \geq B$
$w_3 \geq 0,\ w_3 \geq P_3-(2\varepsilon+3\delta+\beta_1+w_1) \rightarrow w_1+w_3 \geq C$

If s_1, s_2, s_3 were slack variables of these three inequalities, the execution of this algorithm is as Table I. The algorithm deals with the maximization problem, whereas our goal is minimizing $\sum_{i=0}^{3} w_i$. Thus, $\sum_{i=0}^{3} w_i = A + \frac{C-A+B}{2} = \frac{P_1+P_2+P_3}{2} - (4\varepsilon+4\delta+\frac{1}{2}\sum_{i=0}^{3}\beta_i) = max\{0,\ P_1-(2\varepsilon+3\delta+\beta_2),\ P_2-(4\varepsilon+2\delta+\beta_0+\beta_3),\ P_3-(2\varepsilon+3\delta+\beta_1),\ \frac{P_1+P_2+P_3}{2}-(4\varepsilon+4\delta+\frac{1}{2}\sum_{i=0}^{3}\beta_i)\}$.

TABLE I. THE IMPLEMENTATION OF THE SIMPLEX ALGORITHM FOR S^3_{2MF}

	w_1	w_2	w_3	s_1	s_2	s_3	Z
	1	1	1	0	0	0	0
s_1	1	1	0	-1	0	0	A
s_2	0	1	1	0	-1	0	B
s_3	1	0	1	0	0	-1	C
	0	0	1	1	0	0	$-A$
w_1	1	1	0	-1	0	0	A
s_2	0	1	1	0	-1	0	B
s_3	0	-1	1	1	0	-1	$C-A$
	0	0	1	1	0	0	$-A$
w_1	1	0	-1	-1	1	0	$A-B$
w_2	0	1	1	0	-1	0	B
s_3	0	0	2	1	-1	-1	$C-A+B$
	0	0	0	1/2	1/2	1/2	$-A-(C-A+B)/2$
w_1	1	0	0	-1/2	1/2	-1/2	$A-B+(C-A+B)/2$
w_2	0	1	0	-1/2	-1/2	1/2	$B-(C-A+B)/2$
w_3	0	0	1	1/2	-1/2	-1/2	$(C-A+B)/2$

Also, S^3_{6mf} is made up four closed loops. Note the corresponding machine is located in the center of each one of them (See Fig. 7), and the required time is $4\varepsilon+2\delta+\beta_{i-1}+\beta_i+w_{i-1}+w_i$. Due to overlap between closed-loops, $T_S{}^3_{6mf} = 8\varepsilon+8\delta+\sum_{i=0}^{3}\beta_i+w_1+w_2+w_3$ where $w_1=max\{0,\ P_1-(4\varepsilon+6\delta+\beta_2+\beta_3+w_2+w_3)\}$, $w_2=max\{0,\ P_2-(4\varepsilon+6\delta+\beta_0+\beta_3+w_3)\}$, $w_3=max\{0,\ P_3-(4\varepsilon+6\delta+\beta_0+\beta_1+w_1)\}$. This means that $w_1+w_2+w_3=\{0,\ P_1-(4\varepsilon+6\delta+\beta_2+\beta_3),\ P_2-(4\varepsilon+6\delta+\beta_0+\beta_3),\ P_3-(4\varepsilon+6\delta+\beta_0+\beta_1)\}$.

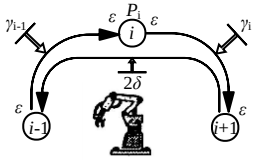

Fig. 7. The closed-loop i of three-machine MFRCs

We can conclude that the cycle times of six permutations are:

$T_S{}^3_{1mf} = 8\varepsilon+\sum_{i=0}^{3}\beta_i+P_1+P_2+P_3$ (1)
$T_S{}^3_{2mf} = 8\varepsilon+4\delta+\sum_{i=0}^{3}\beta_i+max\{0,\ P_1-(2\varepsilon+3\delta+\beta_2),\ P_2-(4\varepsilon+2\delta+\beta_0+\beta_3),\ P_3-(2\varepsilon+3\delta+\beta_1),\ \frac{P_1+P_2+P_3}{2}-(4\varepsilon+4\delta+\frac{1}{2}\sum_{i=0}^{3}\beta_i)\}$ (2)
$T_S{}^3_{3mf} = 8\varepsilon+4\delta+\sum_{i=0}^{3}\beta_i+P_1+max\{0,\ P_2-(2\varepsilon+3\delta+\beta_3),\ P_3-(4\varepsilon+2\delta+\beta_0+\beta_1+P_1)\}$ (3)
$T_S{}^3_{4mf} = 8\varepsilon+4\delta+\sum_{i=0}^{3}\beta_i+P_2+max\{0,\ P_1-(2\varepsilon+3\delta+\beta_3),\ P_3-(2\varepsilon+3\delta+\beta_0)\}$ (4)
$T_S{}^3_{5mf} = 8\varepsilon+4\delta+\sum_{i=0}^{3}\beta_i+P_3+max\{0,\ P_1-(4\varepsilon+2\delta+\beta_2+\beta_3+P_3),\ P_2-(2\varepsilon+3\delta+\beta_0)\}$ (5)
$T_S{}^3_{6mf} = 8\varepsilon+8\delta+\sum_{i=0}^{3}\beta_i+max\{0,\ P_1-(4\varepsilon+6\delta+\beta_2+\beta_3),\ P_2-(4\varepsilon+6\delta+\beta_0+\beta_3),\ P_3-(4\varepsilon+6\delta+\beta_0+\beta_1)\}$ (6)

Now, let us find the optimality regions of these permutations. The results about the regions of optimality for six possible permutations are depicted in Table II. Giving an example, the common region where S^3_{1mf} dominates all permutations must be obtained to introduce S^3_{1mf} as the optimal permutation. This common region is the intersection of all possible dominant conditions. Therefore, S^3_{6mf} is optimal if $P_1+P_2+P_3 \leq 4\delta$ as can be seen from Table II. Giving other example, S^3_{6mf} is optimal if $\beta_{l-1}+P_l+\beta_l \geq 4\varepsilon+6\delta+\sum_{i=0}^{3}\beta_i$ or $P_1+P_2+P_3 \geq 8\varepsilon+16\delta+\sum_{i=0}^{3}\beta_i$. The reason behind this is that the last part of Table II lists the conditions in which S^3_{6mf} dominates any one of another permutations, and the intersection of them equals to $\beta_{l-1}+P_l+\beta_l \geq 4\varepsilon+6\delta+\sum_{i=0}^{3}\beta_i$ or $P_1+P_2+P_3 \geq 8\varepsilon+16\delta+\sum_{i=0}^{3}\beta_i$. This table gives a practical framework to use the robot's permutation with maximum production rate for two- and three machine MFRCs with free-pick up scenario. This framework makes a meaningful contribution to industrial automation, and assists industry in designing and developing appropriate MFRCs.

V. NO-WAIT PICK UP SCENARIO

There is no study which concentrated on MFRCs with no-wait pick up scenario arising when the part must be immediately unload from the machine when its process is finished by the machine. This kind MFRC where machines cannot act as intermediate hoppers is generally called the no-wait MFRC. Since MFR also does operation on the part in transit, the no-wait restriction is not applicable about MFR's operation. The reason behind this is that MFR does secondary operations such as inspection, not primary operations. All secondary operations have same nature and do not respect to no-wait restriction [20]. Finding an optimal permutation for a MFRC with no-wait pick up scenario is a two-phase problem where all feasible permutations are determined in the first phase, and then optimal one is found in the second phase. To make the feasibility condition more clearly, let us present the following counterexample: For ε=0.5, δ=1, P_1=5, P_2=3, β_0=2, β_1=1, β_2=2, the cycle S^2_{2mw} is infeasible because MFR cannot unload a part from M_2 as soon as it is processed by M_2. In fact, the time taken for MFR returns to M_2 is 5, whereas P_2=3.

S^2_{1mw} has no partial waiting; thus, it is always feasible and its cycle time is $T_S{}^2_{1mw} = 6\varepsilon+\sum_{i=0}^{2}\beta_i+P_1+P_2$ regardless of the values of different parameters. However, S_2 has two partial stops on M_1 and M_2 which maybe cause of infeasibility. So, S^2_{2mw} is called feasible when both these partial stops satisfy. In fact, MFR must arrive at M_1 and M_2 not later than finishing the part processing. This means $P_1 \geq 2\varepsilon+2\delta+\beta_2$ and $P_2 \geq 2\varepsilon+2\delta+\beta_0$ are feasibility conditions of S^2_{2mw}.

As mentioned before, I/O is similar to auxiliary machine which should not meet the no-wait restriction. So, a strategy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2014.2371334, IEEE Transactions on Industrial Informatics

TABLE II. THE OPTIMALITY REGION OF SIX PERMUTATIONS OF THREE-MACHINE MFRCS WITH FREE-PICK UP SCENARIO.

S^3_{1mf}		S^3_{2mf}		S^3_{3mf}	
Compared permutations	Dominant Conditions	Compared permutations	Dominant Conditions	Compared permutations	Dominant Conditions
1&2	$P_1+P_2+P_3\leq 4\delta$	2&1	$P_1+P_2+P_3\geq 4\delta$	3&1	$P_2+P_3\geq 4\delta$
1&3	$P_2+P_3\leq 4\delta$	2&3	$P_3-P_1\leq 2\varepsilon+3\delta+\beta_1$ or $P_3-P_1-P_2\leq\beta_1-\beta_3$	3&2	$P_3-P_1\geq 2\varepsilon+3\delta+\beta_1$ and $P_3-P_1-P_2\geq\beta_1-\beta_3$
1&4	$P_1+P_3\leq 4\delta$	2&4	$P_2\geq\beta_3-\beta_2$ or $P_1-P_2\leq 2\varepsilon+3\delta+\beta_2$ or $P_1-P_2-P_3\leq\beta_2+\beta_0$ and $P_2\geq\beta_0-\beta_1$ or $P_3-P_2\leq 2\varepsilon+3\delta+\beta_1$ or $P_3-P_1-P_2\leq\beta_1+\beta_3$	3&4	$P_1\leq P_2$ or $P_2+P_3-P_1\geq 2\varepsilon+3\delta+\beta_0$
1&5	$P_1+P_2\leq 4\delta$	2&5	$P_1-P_3\leq 2\varepsilon+3\delta+\beta_2$ or $P_1-P_2-P_3\leq\beta_2+\beta_0$	3&5	$P_1\leq P_3$ or $P_2+P_3-P_1\geq 2\varepsilon+3\delta+\beta_0$ and $P_1+P_3-P_3\leq 2\varepsilon+3\delta+\beta_3$ or $P_1-P_3\leq\beta_3+\beta_0$
1&6	$P_1+P_2+P_3\leq 8\delta$	2&6	$P_1\leq 2\varepsilon+7\delta+\beta_2$ and $P_2\leq 4\varepsilon+6\delta+\beta_0+\beta_3$ and $P_3\leq 2\varepsilon+7\delta+\beta_1$ and $P_1+P_2+P_3\leq 8\varepsilon+16\delta+\sum_{i=0}^{3}\beta_i$	3&6	$P_1\leq 4\delta$ and $P_1+P_2\leq 2\varepsilon+7\delta+\beta_3$ and $P_3\leq 4\varepsilon+6\delta+\beta_0+\beta_1$
S^3_{4mf}		S^3_{5mf}		S^3_{6mf}	
Compared permutations	Dominant Conditions	Compared permutations	Dominant Conditions	Compared permutations	Dominant Conditions
4&1	$P_1+P_3\geq 4\delta$	5&1	$P_1+P_2\geq 4\delta$	6&1	$P_1+P_2+P_3\geq 8\delta$ or $P_1\geq 4\varepsilon+6\delta+\beta_2+\beta_3$ or $P_2\geq 4\varepsilon+6\delta+\beta_0+\beta_3$ or $P_3\geq 4\varepsilon+6\delta+\beta_0+\beta_1$
4&2	$P_1-P_2\geq 2\varepsilon+3\delta+\beta_2$ or $P_3-P_2\geq 2\varepsilon+3\delta+\beta_1$ and $P_2\leq\beta_3+\beta_2$ or $P_3-P_1-P_2\geq\beta_1+\beta_3$ and $P_1-P_2-P_3\geq\beta_2+\beta_0$ or $P_2\leq\beta_0+\beta_1$	5&2	$P_1-P_3\geq 2\varepsilon+3\delta+\beta_2$ and $P_1-P_2-P_3\geq\beta_2-\beta_0$	6&2	$P_1\geq 2\varepsilon+7\delta+\beta_2$ or $P_2\geq 4\varepsilon+6\delta+\beta_0+\beta_3$ or $P_3\geq 2\varepsilon+7\delta+\beta_1$ or $P_1+P_2+P_3\geq 8\varepsilon+16\delta+\sum_{i=0}^{3}\beta_i$
4&3	$P_2\leq P_1$ and $P_2+P_3-P_1\leq 2\varepsilon+3\delta+\beta_0$	5&3	$P_3\leq P_1$ or $P_1+P_2-P_3\geq 2\varepsilon+3\delta+\beta_3$ and $P_2+P_3-P_1\leq 2\varepsilon+3\delta+\beta_0$ or $P_1-P_3\geq\beta_3-\beta_0$	6&3	$P_1\geq 4\delta$ or $P_1+P_2\geq 2\varepsilon+7\delta+\beta_3$ or $P_3\geq 4\varepsilon+6\delta+\beta_0+\beta_1$
4&5	$P_2\leq P_3$ and $P_1+P_2-P_3\leq 2\varepsilon+3\delta+\beta_3$	5&4	$P_3\leq P_2$ or $P_1+P_2-P_3\geq 2\varepsilon+3\delta+\beta_3$	6&4	$P_2\geq 4\delta$ or $P_1+P_2\geq 2\varepsilon+7\delta+\beta_3$ or $P_2+P_3\geq 2\varepsilon+7\delta+\beta_0$
4&6	$P_2\leq 4\delta$ and $P_1+P_2\leq 2\varepsilon+7\delta+\beta_3$ and $P_2+P_3\leq 2\varepsilon+7\delta+\beta_0$	5&6	$P_3\leq 4\delta$ and $P_1\leq 4\varepsilon+6\delta+\beta_2+\beta_3$ and $P_2+P_3\leq 2\varepsilon+7\delta+\beta_0$	6&5	$P_3\geq 4\delta$ or $P_1\geq 4\varepsilon+6\delta+\beta_2+\beta_3$ or $P_2+P_3\geq 2\varepsilon+7\delta+\beta_0$

for making S^2_{2mw} feasible is that the part enters the MFRC with a time delay. This release time is indicated by R to calculate $T_{S^2_{2mw}}$. Clearly, the cycle time is $T_{S^2_{2mw}}=R+6\varepsilon+3\delta+\sum_{i=0}^{2}\beta_i+w_1+w_2$ where R and w_1+w_2 are not constant parts. We have:

$$w_1= P_1-(2\varepsilon+2\delta+\beta_2+w_2) \text{ and } w_2= P_2-(2\varepsilon+2\delta+\beta_0+R) \quad (7)$$
$$\leftrightarrow w_1+w_2= P_1-(2\varepsilon+2\delta+\beta_2) \quad (8)$$
$$\leftrightarrow R= max\{0, P_2+\beta_2-(P_1+\beta_0)\} \quad (9)$$

So, $T_{S^2_{2mw}}$ is shown by the double-sided function $max\{4\varepsilon+\delta+\beta_0+P_1+\beta_1, 4\varepsilon+\delta+\beta_1+P_2+\beta_2\}$. After derivation of $T_{S^2_{1mw}}$ and $T_{S^2_{2mw}}$, the performance of S^2_{1mw} and S^2_{2mw} should be compared to optimize two-machine MFRCs with no-wait scenario. Since $T_{S^2_{1mw}} > T_{S^2_{2mw}}$, we can conclude that *$S^2_{2mw}$ is certainly optimal if it be feasible. It is only enough to check S^2_{2mw} meets the feasibility conditions ($P_1\geq 2\varepsilon+2\delta+\beta_2$ and $P_2\geq 2\varepsilon+2\delta+\beta_0$).*

Optimizing three-machine MFRCs are complicated in comparison with two-machine ones. This is even more difficult when pick up scenario is no-wait. In fact, it is possible that no overlap exist between three machine operations and four MFR operations. In other words, every one of machines and MFR is potentially critical equipment if it shortly processes the part. Initially, we should take problem feasibility into consideration to better formulate no-wait restriction and estimate the gain of productivity. The cycle time and feasibility region of six permutations are listed in Table III. This table indicates that the scheduling problem is never infeasible because S^3_{1mw} always gives a guarantee of feasibility. For the sake of simplicity, we present the process of cycle time calculation for S^3_{3mw} and S^3_{4mw} here, to show how we obtained the rest of cycle times and feasibility conditions except for S^3_{2mw} in Table III. At first glance in S^3_{3mw}, there are two partial waits on M_2 and M_3. Only these two critical points may make S^3_{3mw} infeasible. Indeed, P_2 and P_3 must not be smaller than the time elapses between when the corresponding machine was loaded and when MFR come back to remove it. Two inequalities $P_2\geq 2\varepsilon+3\delta+\beta_3$ and $P_3\geq 4\varepsilon+2\delta+P_1+\beta_0+\beta_1$ cover the state space of S^3_{3mw} in keep with A_0, A_1, A_3, A_2. Also, $T_{S^3_{3mw}}=R+8\varepsilon+4\delta+\sum_{i=0}^{3}MRP_i+P_1+w_2+w_3$ where w_1 and w_2 are:

$$w_2= P_2-(2\varepsilon+3\delta+\beta_3+w_3) \text{ and } w_3=P_3-(4\varepsilon+2\delta+\beta_0+\beta_1+P_1+R) \quad (10)$$
$$\leftrightarrow w_2+w_3= P_2-(2\varepsilon+3\delta+\beta_3) \quad (11)$$

Therefore, $R=max\{0, P_3+\delta+\beta_3-(P_1+P_2+2\varepsilon+\beta_0+\beta_1)\}$ and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2014.2371334, IEEE Transactions on Industrial Informatics

$T_{S^3_{3mw}}=max\{6\varepsilon+\delta+\beta_0+\beta_1+\beta_2+P_1+P_2,\quad 4\varepsilon+2\delta+\beta_2+\beta_3+P_3\}$. Also, $S^3_{4mw}=A_0,A_1,A_3,A_2$ has two partial waits P_1-$2\varepsilon+3\delta+\beta_3$ and P_3-$2\varepsilon+3\delta+\beta_0+R$ on M_1 and M_3, respectively. Since both of these partial waiting must be positive; the intersection of $P_1\geq 2\varepsilon+3\delta+\beta_3$ and $P_3\geq 2\varepsilon+3\delta+\beta_0$ shows feasible state space of S^3_{4mw}. Also, the cycle time of S^3_{4mw} is $T_{S^3_{4mw}}=R+8\varepsilon+4\delta+\sum_{i=0}^{3}\beta_i+P_1+w_1+w_3$ where $w_1+w_3=P_1-(2\varepsilon+3\delta+\beta_3)$. This result:

$$R=max\{0,\ P_3+\beta_3-(P_1+\beta_0)\} \tag{12}$$

$$T_{S^3_{4mw}}=max\{6\varepsilon+\delta+\beta_0+\beta_1+\beta_2+P_1+P_2, 6\varepsilon+\delta+\beta_1+\beta_2+\beta_3+P_2+P_3\} \tag{13}$$

S^3_{2mw} is a tough permutation to deal with. Indeed, MFR has three partial stops in addition to artificial stop R at I/O during execution of this permutation. Note R can be called w_0 or w_4. The constant portion of S^3_{2mw} is $8\varepsilon+4\delta+\sum_{i=0}^{3}\beta_i$, whereas $w_1+w_2+w_3+R$ is the variable portion of it that should be minimized. $w_1+w_2+w_3+R$ is built up four sub portions $w_1=P_1-(2\varepsilon+3\delta+\beta_2+w_2)\geq 0$, $w_2=P_2-(4\varepsilon+2\delta+\beta_0+\beta_3+w_3+R)\geq 0$, $w_3=P_3-(2\varepsilon+3\delta+\beta_1+w_1)\geq 0$, and $w_4=R\geq 0$. We rewrite this minimization problem as the following formulation reassuming $A=P_1-(2\varepsilon+3\delta+\beta_2)$, $B=P_2-(4\varepsilon+2\delta+\beta_0+\beta_3)$, and $C=P_3-(2\varepsilon+3\delta+\beta_1)$:

$$\text{Mini } Z=w_1+w_2+w_3+w_4 \tag{14}$$

$$\text{Subject to } w_1+w_2=A \tag{15}$$

$$w_2+w_3+w_4=B \tag{16}$$

$$w_1+w_3=C \tag{17}$$

$$w_1,\ w_2,\ w_3,\ w_4\geq 0$$

$Z=w_1+B$ is an indirect result from (16). Thus, it is enough to find minimum amount of w_1 which is presented in four sub-cases representing the corner points the feasibility region:

1. $w_1=0,\ w_2=A\geq 0,\ w_3=C\geq 0,\ w_4=B-(A+C)\geq 0 \rightarrow B\geq A+C$
2. $w_1=A\geq 0,\ w_2=0,\ w_3=C-A\geq 0\rightarrow C\geq A,\ w_4=B-(C-A)\geq 0\rightarrow A+B\geq C$
3. $w_1=C\geq 0,\ w_2=A-C\geq 0\rightarrow A\geq C,\ w_3=0,\ w_4=B-(A-C)\geq 0\rightarrow B+C\geq A$
4. $w_1=(A+C-B)/2 \rightarrow A+C\geq B,\ w_2=(B+A-C)/2 \rightarrow A+B\geq C$
   $w_3=(B+C-A)/2 \rightarrow B+C\geq A,\ w_4=0$

Let us assume $B=A+C$ is the breakpoint dividing the feasible regions of corner points 1 and 4. The corner points 1 is feasible for the left side of this breakpoint ($B\geq A+C$), and the amount of $w_1=0$ for this corner point is smaller than the amount for second and third corner points ($w_1=A\geq 0$ & $w_1=C\geq 0$). On the other hand, the corner points 4 is feasible for the right side of ($B<A+C$). Then, the amount of w_1 of the corner point 4 is $0\leq w_1\leq A$ and $0\leq w_1\leq C$ if $A\leq C$ and $C\leq A$. This prove that the amount w_1 of the corner point 4 is smaller than both 2 and 3 which are $w_1=A$ & $w_1=C$. So, the corner points 2 and 3 should be omitted from the formulation of $T_{S^3_{2mw}}$ in that one of the corner points 1 or 4 always dominates both of them and has smaller w_1. Note it is impossible to execute S^3_{2mw} if $B+C<A$ or $A+B<C$. In fact, $B+C=w_1+w_2+2w_3+w_4\geq A=w_1+w_2$ and $A+B=w_1+2w_2+w_3+w_4\geq C=w_1+w_3$ with respect to (15)-(17). We calculate two possible subcases of R using the original value of A, B, and C. Then, $T_{S^3_{2mw}}$ is obtained from the summation of the constant potion $8\varepsilon+4\delta+\sum_{i=0}^{3}\beta_i$ and the variable portion $Z=w_1+w_2+w_3+w_4=w_1+B$. This result:

$$R\begin{cases} P_2+4\delta+\beta_1+\beta_2-(P_1+P_3+\beta_0+\beta_3) & B\geq A+C \\ 0 & B<A+C \end{cases} \tag{18}$$

$$T_{S^3_{2mw}}\begin{cases} 4\varepsilon+2\delta+\beta_1+\beta_2+P_2 & B\geq A+C \\ 4\varepsilon+\dfrac{\sum_{i=0}^{3}\beta_i+P_1+P_2+P_3}{2} & B<A+C \end{cases} \tag{19}$$

Considering feasibility condition of S^3_{2mw}, (18) and (19) are rewritten by two *max* terms in the second row of Table III. Now, we need an algorithm to reach the optimal permutation using the outcome of the Table III. This algorithm is:

Search Algorithm: Finding the feasible and optimal permutation.

Input: State information (Machines and MFR's process times, empty MFR travel time, load/unload time).

for j=1 to 6
  if S^3_{jmw} is feasible according to conditions in Table III **then**
    $S \leftarrow S + S^3_{jmw}$
  end
Initialization of $T^*= \infty$
for x=1 to s
  if $T_{S^3_{xmf}} \leq T^*$ **then**
    $S^* \leftarrow S^3_{xmw}$
    $T^* \leftarrow T_{S^3_{xmf}}$
  else
    $S^* \leftarrow S^*$
    $T^* \leftarrow T^*$
  end
Output: The optimal permutation S^* and its cycle time T^*

As shown above, Search Algorithm is constructed from Table III. The mechanism to reach the optimal permutation in

TABLE III. THE CYCLE TIME AND FEASIBILITY REGION OF SIX PERMUTATIONS OF THREE-MACHINE MFRCS WITH NO-WAIT PICK UP SCENARIO.

Permutation	Feasibility Conditions	Release Time	Cycle Time
S^3_{1mw}	*Always*	0	$8\varepsilon+\sum_{i=0}^{3}\beta_i+P_1+P_2+P_3$
S^3_{2mw}	$P_1\geq 2\varepsilon+3\delta+\beta_2$ $P_2\geq 4\varepsilon+2\delta+\beta_0+\beta_3$ $P_3\geq 2\varepsilon+3\delta+\beta_1$ $B+C\geq A$ & $A+B\geq C$	$max\{0,\ P_2+4\delta+\beta_1+\beta_2-(P_1+P_3+\beta_0+\beta_3)\}$	$max\{4\varepsilon+2\delta+\beta_1+\beta_2+P_2,\ 4\varepsilon+\frac{\sum_{i=0}^{3}\beta_i+P_1+P_2+P_3}{2}\}$
S^3_{3mw}	$P_2\geq 2\varepsilon+3\delta+\beta_3$ $P_3\geq 4\varepsilon+2\delta+\beta_1+\beta_0+\beta_1$	$max\{0,\ P_3+\delta+\beta_3-(P_1+P_2+2\varepsilon+\beta_0+\beta_1)\}$	$max\{6\varepsilon+\delta+\beta_0+\beta_1+\beta_2+P_1+P_2,\ 4\varepsilon+2\delta+\beta_2+\beta_3+P_3\}$
S^3_{4mw}	$P_1\geq 2\varepsilon+3\delta+\beta_3$ $P_3\geq 2\varepsilon+3\delta+\beta_0$	$max\{0,\ P_3+\beta_3-(P_1+\beta_0)\}$	$max\{6\varepsilon+\delta+\beta_0+\beta_1+\beta_2+P_1+P_2,\ 6\varepsilon+\delta+\beta_1+\beta_2+\beta_3+P_2+P_3\}$
S^3_{5mw}	$P_1-P_3\geq 4\varepsilon+2\delta+\beta_2+\beta_3$ $P_2\geq 2\varepsilon+3\delta+\beta_0$	$max\{0,\ P_2+P_3+2\varepsilon+\beta_2+\beta_3-(P_1+\delta+\beta_0)\}$	$max\{4\varepsilon+2\delta+\beta_0+\beta_1+P_1,\ 6\varepsilon+\delta+\beta_1+\beta_2+\beta_3+P_2\}$
S^3_{6mw}	$P_1\geq 4\varepsilon+6\delta+\beta_2+\beta_3$ $P_2\geq 4\varepsilon+6\delta+\beta_0+\beta_3$ $P_3\geq 4\varepsilon+6\delta+\beta_0+\beta_1$	$max\{0,\ P_2+\beta_2-(P_1+\beta_0),\ P_3+\beta_2+B_3-(P_1+\beta_0+\beta_1)\}$	$max\{4\varepsilon+2\delta+\beta_0+\beta_1+P_1,\ 4\varepsilon+2\delta+\beta_1+\beta_2+P_2,\ 4\varepsilon+2\delta+\beta_2+\beta_3+P_3\}$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2014.2371334, IEEE Transactions on Industrial Informatics

trivial time is defining the set of feasible permutation s∈*S*, and then finding the optimal permutation S^* and its cycle time T^* using two For Loops. Anyone of permutations is stopped when an infeasible activity occurs in its activity route. In brief, it is expected the outcome of this algorithm be a practical help for robotic cell manufacturers who face difficult task of forming and scheduling a no-wait MFRC.

VI. Conclusion

An effective methodology was developed in this study for addressing the issue of industrial robots' functionality within a cellular production system. Two and six feasible permutations are developed for two- and three-machine MFRCs with the free pick up scenario, and the optimality regions of these permutations and their formulas are determined. Then, the results are extended to the no-wait pick up scenario. Through this research it was found there is no unique optimal permutation for MFR movement between different stations with different parameter inputs. To state the matter differently, it should be noted any one of the permutations has the chance of obtaining optimality considering different values of ε, δ, P_1, P_2, P_3, γ_0, γ_1, γ_2, γ_3. It is enough to check whether it meets the optimality conditions or not. The scheduling method developed in this research can be broadened for multi-unit permutations in future research directions. In addition, some mathematical formalism such as max-plus algebra can be an important tool for research in this area to simplify the procedure for determination of cycle times. In fact, the analysis of all partial waits can be eliminated using max-plus algebra since synchronization is an inherent property of max-plus algebra systems. Lastly, reentrant MFRCs where a part visits a machine more than once in its processing route can be taken into account in future work.

References

[1] A. Ferrolho, and M. Crisóstomo, "Intelligent Control and Integration Software for Flexible Manufacturing Cells," *IEEE Trans. Ind. Inf.*, vol.3, no.1, pp.3–11, Feb. 2007.

[2] I. Sindicic, S. Bogdan, and T. Petrovic, "Resource Allocation in Free-Choice Multiple Reentrant Manufacturing Systems Based on Machine-Job Incidence Matrix," *IEEE Trans. Ind. Inf.*, vol.7, no.1, pp.105–114, Feb. 2011.

[3] www.controlgaging.com/products-grip-gage-go.html

[4] M. Foumani, Y. Ibrahim and I. Gunawan, "Cyclic scheduling in small-scale robotic cells served by a multi-function robot" *IEEE 39th Ann. Conf. Ind. Elec. Soc.*, Nov 2013, pp. 4362–4367.

[5] M. Foumani, and K. Jenab, "Analysis of flexible robotic cells with improved pure cycle," *Int. J. Compu. Integrate. Manuf.*, vol. 26, pp. 201–215, Mar. 2013.

[6] www.haascnc.com/techfo_links.asp#gsc.tab=0

[7] G. Quan, and V. Chaturvedi, "Feasibility Analysis for Temperature-Constraint Hard Real-Time Periodic Tasks" *IEEE Trans. Ind. Inf.*, vol. 6, no. 3, pp. 329–339, Aug. 2010.

[8] S.P. Sethi, C. Sriskandarajah, G. Sorger, J. Blazewicz, and W. Kubiak, "Sequencing of parts and robot moves in a robotic cell," *Int. J. Flex. Manuf. Syst.*, vol. 4, pp. 331–358, Mar. 1992.

[9] S.P. Sethi, J.B. Sidney and C. Sriskandarajah, "Scheduling in dual gripper robotic cells for productivity gains," *IEEE Trans. Robot. & Autom.*, vol. 17, pp. 324–341, Jun. 2001.

[10] I.G. Drobouchevitch, S.P. Sethi, J.B. Sidney, and C. Sriskandarajah, "A note on scheduling multiple parts in two-machine dual-gripper robot cells: heuristic algorithm and performance guarantee," *Int. J. Operat. Quantitative Manage.*, vol. 10, pp. 297–314, Jan. 2004.

[11] H.N Geismar, S.P. Sethi, J.B. Sidney, C. Sriskandarajah, "A note on productivity gains in flexible robotic cells," *Int. J. Flex. Manuf. Syst.*, vol. 17, pp. 5–21, Jan. 2005.

[12] A. Nambiar, and R. Judd, "Max-plus based mathematical formulation for cyclic permutation flow-shops," *International Journal of Mathematical Modeling and Numerical Optimization*, vol.2, no.1, pp.85–97, Feb. 2011.

[13] M. Foumani, and K. Jenab, "Cycle time analysis in reentrant robotic cells with swap ability," *Int. J. Prod. Res.*, vol. 50, pp. 6372–6387, Nov. 2012.

[14] A. Agnetis, "Scheduling no-wait robotic cells with two and three machines," *Eur. J. Oper. Res.*, vol. 123, pp. 303–314, Jun. 2000.

[15] H.J. Paula, C. Bierwirth, and H. Kopfer, "A heuristic scheduling procedure for multi-item hoist production lines" *Int. J. Prod. Econ*, vol. 105, pp. 54–69, Jan. 2007.

[16] D. Alcaide et al, "Cyclic multiple-robot scheduling with time-window constraints using a critical path approach" *Eur. J. Oper. Res.*, vol. 177, pp. 147–162, Feb. 2007.

[17] N. Brauner et al, "Scheduling of coupled tasks and one-machine no-wait robotic cells" *Comput. & Oper. Res.*, vol. 36, pp. 301–307, Feb. 2009.

[18] S. Keating, and N. Oxman, "Compound Fabrication: A Multi-functional Robotic Platform for Digital Design and Fabrication" Robot. Computer-Integrated Manuf., vol. 29, pp. 439–448, Dec. 2013.

[19] M. Foumani and K. Jenab, "An operation-oriented analysis of hybrid robotic cells" *Int. J. Robot. & Autom.*, vol. 28, pp. 123–128, May. 2013.

[20] M. Foumani, I. Gunawan, and Y. Ibrahim, "Scheduling rotationally arranged robotic cells served by a multi-function robot" *Int. J. Prod. Res.*, vol. 52, pp. 4037–4058, Jan. 2014.

[21] W-K. Chan, J. Yi, S. Ding, and D. Song, "Optimal scheduling of k-unit production of cluster tools with single-blade robots" *IEEE Int. Conf. Auto. Science & Eng.*, Aug. 2008, pp. 335–340.

[22] P. Yan, C. Chu, A. Che, and N. Yang, "An algorithm for optimal cyclic scheduling in a robotic cell with flexible processing times," *IEEE Int. Conf. Ind. Eng. & Eng. Manage.*, Dec 2008, pp. 153–157.

Biography

Mehdi Foumani is a graduated in Industrial Engineering at the University of Tehran. His interests include Sequencing and Scheduling, Reliability, Optimization, Stochastic Modelling, and Project Management. Now, he is a Ph.D. student of Engineering at the School of Applied Sciences and Engineering at Monash University, where he is working for his thesis on the topic of advanced robotic cells scheduling.

Indra Gunawan completed his Ph.D. degree in Industrial Engineering from Northeastern University, USA. He is currently a Senior Lecturer and Coordinator of Postgraduate Programs in Maintenance and Reliability Engineering in the School of Engineering and Information Technology at Federation University Australia. His main areas of research are maintenance and reliability engineering, project management, and operations management.

Kate Smith-Miles is a Senior Member of IEEE, and received a B.Sc(Hons) in Mathematics and a Ph.D., both from the University of Melbourne. She has published 2 books and over 230 refereed journal and international conference papers in the areas of neural networks, intelligent systems and data mining. She is a Professor in the School of Mathematical Sciences at Monash University, and Director of the Monash for Cross & Interdisciplinary Mathematical Applications.

M. Yousef Ibrahim is a Senior Member of IEEE, received the Ph.D. degree in the field of robotics from the University of Wollongong, Australia, in 1993. He is a professor of engineering and leader of Mechatronics at Federation University Australia. His main research areas are in mechatronics applications, industrial automation and reliability engineering. He was also the founder of the Master program on Maintenance and Reliability Engineering.