Transaction E: Industrial Engineering
Vol. 16, No. 1, pp. 36–54


# Robot Movements in a Cyclic Multiple-Part Type Three-Machine Flexible Robotic Cell Problem

**I.N. Kamal Abadi[1,*] and S. Gholami[2]**

**Abstract.** *This paper recognizes thirty-six, potentially optimal robot movement policies, to schedule the movements of a robot in a three-machine flexible cell. The robotic cell produces multi-type parts, in which the robot is used as a material handling system. In this manufacturing cell, the machines have operational flexibility and can be set up for different operations; all parts have three operations. Finding the robot movement policy and sequence of parts to minimize the cycle time (i.e., maximize the throughput) is the aim of this work. It was proved that cycle time calculation, in twelve out of thirty-six policies, are unary NP-complete, and a polynomial time algorithm is introduced that can solve the twenty-four left policies. This paper develops the cycle times of all these thirty six robot movements policies, considering waiting times in a flexible three-machine robotic cell with multi-type parts, and introduces a parts sequence under a special condition, in which one of the policies minimizes the cycle time (i.e., maximize throughput). This kind of flexibility differs from other research into robotic cells, wherein a machine can process different operations. Moreover, we consider cells with multiple part types, which is more realistic than other developed models. Finally, a new mathematical model, based on Petri-nets, was provided for one of the robot movement policies. Furthermore, this mathematical model is also developed for the multi-type part problem.*

**Keywords**: *Production scheduling; Cyclic blocking open shop; Flexible robotic cell.*

## INTRODUCTION

In modern technology, the level of automation in manufacturing industries has increased dramatically. Some examples of these automation progresses are in cellular manufacturing and robotic cells. A growing body of evidence suggests that, in a wide variety of industrial settings, material handling within a cell can be accomplished very efficiently by employing industrial robots (see [1]). Among the interrelated issues to be considered in using robotic cells are their designs, the scheduling of robot movements and the sequencing of parts to be produced. If, in a robotic cell, CNC machines are used, it is possible to set up different operations on machines. This important property of CNC machines is considered, and the 1-unit cycles of robot movement policies are defined. In most previous studies, flow-shop robotic cells are considered [2]. In this paper, a robotic cell with operational flexibility that produces multiple-type parts, is considered and, also, the results of the successful studies of Hall et al. [3,4] and Sriskandarajah et al. [5]. Hall [3] analyzed the complexity of a scheduling problem and showed that two policies out of six possible policies are NP-Complete. In this study, these results are used to show the NP-Completeness of the scheduling problems of twelve out of thirty six possible policies. For calculating the cycle time and waiting times for a given sequence of jobs under these two NP-Complete policies, algorithms are introduced [4]. We introduce a mathematical model, based on Petri-nets, to find the best sequence of jobs and calculate the cycle time and waiting times for policies, wherein their scheduling problems are NP-Complete. Some instance problems are solved by this mathematical model and the results are illustrated. Sriskandarajah et al. [5] have introduced a classification scheme for the complexity of robotic

1. *Department of Industrial Engineering, University of Kurdistan, Sanandaj, Iran.*
2. *Department of Industrial Engineering, Tarbiat Modares University, Tehran, P.O. Box 14115-143, Iran.*
*. *Corresponding author. E-mail: nakhai_isa@yahoo.com*

*Received 19 May 2007; received in revised form 16 October 2007; accepted 17 February 2008*

cell scheduling problems. Finally, according to this classification scheme, some algorithms are proposed to solve this problem. This paper is organized as follows.

In the next section, the literature of the robotic cell scheduling problem is briefly reviewed. In the third section, the initial notions and required notations are introduced. Following that, the problem and calculation of cycle times for possible policies are described. In the fifth section, the problem is analyzed by Petri-nets and a mathematical model and its calculation results are shown. Finally, the proposed algorithms for solving the problem are described.

## LITERATURE REVIEW

The robotic cell problem, wherein a robot is used as a material handling system, has received considerable attention, such as in [6,7] and some other works which were pursued. Sethi et al. [6] suggested the following conjecture.

### Sethi's Conjecture

In bufferless single-gripper robotic cells producing a single part-type and having identical robot travel times between adjacent machines and identical load/unload times, a 1-unit cycle provides the minimum per unit cycle time in the class of all solutions, cyclic or otherwise.

Sethi et al. [6] proved this conjecture for two-machine robotic cell problems. It is proved that a 1-unit cycle solution is optimal over the class of all solutions, cyclic or otherwise. For a three-machine case, Crama and van de Klundert [8] and Brauner and Finke [9] showed that the best 1-unit cycle is the optimal solution for the class of all cyclic solutions.

Hall et al. [3,4] considered three-machine robotic cells, which produce multiple-type parts. They analyzed the complexity of this problem and proved that two out of six possible policies are unary NP-complete and that the other four policies are solvable in polynomial time.

Crama et al. [2] studied flow-shop scheduling problems, the models for such problems and their complexity. Dawande et al. [10] provided a classification scheme for a robotic cells scheduling problem.

Some other special cases have been studied. Gultekin et al. [11] studied the robotic cell scheduling problem with tooling constraints for a two-machine robotic cell, where some operations can only be processed on the first machine, some others can only be processed on the second machine and the remaining operations can be processed on both machines.

Gultekin et al. [12] considered a flexible manufacturing robotic cell with identical parts, wherein the machines are able to do different operations and the operations assignment to the machines can vary through different cycles. The aim is finding the operations assignment for three machines in different cycles and they proposed a lower bound for 1 and 2 unit cycles. Geismar et al. [13] found that, in a two-machine flexible robotic cell, no increase in throughput can be achieved by operational flexibility and, in three-machine and four-machine flexible robotic cells, at most, a $14\frac{2}{7}\%$ increase can be achieved by operational flexibility. They assume that identical parts are produced in the flexible robotic cell.

## INITIAL NOTIONS AND REQUIRED NOTATIONS

The robotic cell problem is a special case of the cyclic blocking flow-shop, where the jobs might block either the machine or the robot at the processing time. A cyclic schedule is one in which the same sequence of states is repeated over and over again. A cycle in such a schedule begins at any state and ends when that state is encountered next. In previous studies, authors assumed that the discipline for the movements of parts is an ordinary flow-shop discipline, i.e., a part meets machines $M_1$, $M_2$ and $M_3$ consequently. As Blazewicz et al. [7] showed, under a flow-shop discipline, there are six different potentially optimal policies for a robot to move the parts between these three machines. In this research, the robotic movement policies will be examined with more flexibility. We assume that the machines can be configured for different operations and the operations can be done in different sequences. Usually, this kind of sequencing, wherein each job needs to be processed exactly once on each of the machines, but in which the order of processing is immaterial, is called an open-shop discipline. In this study, the focus is on organizing the possible policies for robot movements in a three-machine robotic cell under the open-shop processing environment. The following notation is used to describe the robotic cell problem in:

| | |
|---|---|
| $m$: | The number of machines; |
| $I/O$: | The automated input-output system for the cell; |
| $\mathrm{PT}_1, \mathrm{PT}_2, \cdots, \mathrm{PT}_k$: | The part-types to be produced; |
| $r_1, r_2, \cdots, r_k$: | The minimal ratios of parts to be produced; |
| MPS: | A minimal part set consisting of $r_i$ parts of type $\mathrm{PT}_i$, $l = 1, 2, \cdots, k$; |
| $n = r_1 + r_2 + \cdots + r_k$: | The total number of parts to be produced in the MPS; |

$a_i, b_i, c_i, \cdots$: The processing times of part $i$ on 1st, 2nd, and 3rd $\cdots$ stages;

$\delta$: Time taken by robot when, traveling between two consecutive machines. $I/O$ is assumed as machine $M_0$;

$\varepsilon$: Time taken by the robot to pick up a part from $I/O$, drop a part at $I/O$, load a part onto machine $M_i$, or unload a part from machine $M_i$;

$w_j^i$: The time that the robot waits at machine $M_j$ to unload part $P_i$, where the machine is still processing the part;

$(\chi_1, \chi_2, \cdots, \chi_m)$: The current state of the system, where $\chi_i = \phi$(or $\Omega$) means that machine $M_i$ is free (or occupied by a part);

$S_I^k$: The movement policy of category $k$ where the operations sequence is $O_1$, $O_2$ and $O_3$;

$\hat{S}_I^k$: The movement policy of category $k$ where the operations sequence is reverse to policy $S_I^k$;

$S_{II}^k$: The movement policy of category $k$ where the operations sequence is $O_2$, $O_1$ and $O_3$;

$\hat{S}_{II}$: The movement policy of category $k$ where the operations sequence is reverse to policy $S_{II}^k$;

$S_{III}^k$: The movement policy of category $k$ where the operations sequence is $O_1$, $O_3$ and $O_2$;

$\hat{S}_{III}^k$: The movement policy of category $k$ where the operations sequence is reverse to policy $S_{III}^k$;

$T_i^k$: The steady state cycle time for the repetitive manufacturing of an MPS corresponding to movement policy $S_i^k$.

In this study, the standard classification scheme for scheduling problems, $\psi_1|\psi_2|\psi_3$, is used, where $\psi_1$ indicates the scheduling environment, $\psi_2$ describes the job characteristics and $\psi_3$ defines the objective function [10]. For example, $\text{FRC}_3|k \geq 2, S^1|C_t$ denotes the minimization of cycle time for a multi-type part problem in a three flow-shop robotic cell, restricted to robot move cycle $S^1$. Moreover, $\text{ORC}_m$ denotes the $m$ machines open-shop robotic cell.

## THREE MACHINE FLEXIBLE ROBOTIC CELL $\text{ORC}_3|k \geq 2, S^1|C_t$

In order to explain the problem, consider a machining cell, where three-machine tools are located. A robot is used to feed these three machines, namely, $M_1$, $M_2$ and $M_3$, in the cell, where parts are brought to and removed from the robotic cell by an Automated Storage & Retrieval System (AS/RS). The pallets and feeders of the AS/RS system allow hundreds of parts to be loaded into the cell without human intervention (see Figure 1) and the machines can be configured to perform any operation. The aim of this paper is to find a schedule for the robot movement and the sequence of parts to maximize throughput (i.e., to minimize cycle time).

In each cycle, $n$ parts (the total number of parts in the MPS) are produced, in which $r_1$ are the parts of part type 1 and $r_2$ are the parts of part type 2. In an $m$-machine flexible cell, all parts in a MPS visit each machine in the same order. However, the operations can be performed in any order and each machine can be configured to perform any operation.

Sethi et al. [6] showed that there are exactly $m!$ potentially optimal 1-unit cycles in a $m$-machine flow-shop robotic cell (note that a 1-unit cycle returns to the same state after the production of a single unit). They also showed that any potentially optimal 1-unit robot move cycle in a $m$-machine robotic cell can be described by exactly $m+1$, following basic activities:

$$M_i^- : \text{ Load a part on } M_i, \quad i = 1, 2, \cdots m,$$

$$M_m^+ : \text{ Unload a finished part from } M_m.$$

Note that, in a 1-unit cycle, every basic activity must be carried out exactly once. Moreover, since, in an optimal cycle we require that the robot move path

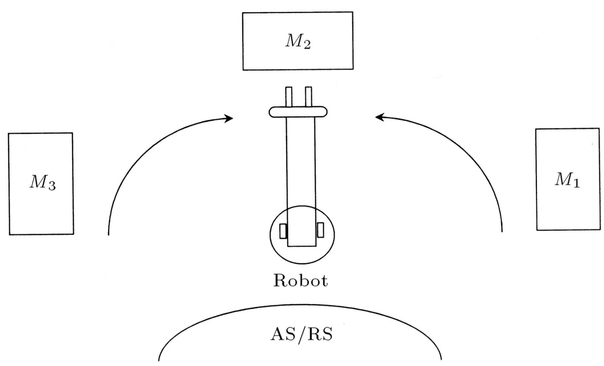


**Figure 1.** Robotic work cell layout with three machines.

be as short as possible, any two consecutive activities uniquely determine the robot moves between them. Therefore, a cycle can be uniquely described by a permutation of the above $m+1$ activities. The following are the available robot move cycles for a $m=3$ flow-shop robotic cell, as described by Sethi et al. [6]:

$$S^1:\{M_3^+, M_1^-, M_2^-, M_3^-, M_3^+\},$$

$$S^2:\{M_3^+, M_1^-, M_3^-, M_2^-, M_3^+\},$$

$$S^3:\{M_3^+, M_3^-, M_1^-, M_2^-, M_3^+\},$$

$$S^4:\{M_3^+, M_2^-, M_3^-, M_1^-, M_3^+\},$$

$$S^5:\{M_3^+, M_2^-, M_1^-, M_3^-, M_3^+\},$$

$$S^6:\{M_3^+, M_3^-, M_2^-, M_1^-, M_3^+\}.$$

In flow-shop robotic cells, the computational results obtained by Hall et al. [3] suggest that, besides their practical advantages, 1-unit robot move cycles routinely provide schedules with cycle times that are very close to the lower bounds available for all possible robot move cycles.

**Lemma 1**

For the flexible robotic cell there are $(m!)^2$ potentially optimal 1-unit cycles.

***Proof***

From Sethi et al. [6] we have $m!$ potential optimal 1-unit cycles in a $m$ machine flow-shop. For every potentially optimal cycle sequence of a flow-shop robotic cell, we have a $m$ machine arranged as $M_1, M_2, \cdots, M_m$. To produce a flexible robotic cell, we can rearrange the operations of every flow-shop robotic cell by $m!$ possible arrangements. Thus, we have $(m!)^2$ potentially optimal 1-unit cycles in flexible robotic cells and this completes the proof.

The category number is described as the superscript on the policy notations, i.e., $S^k$ represents policy $S$ under category $k$. Similar notations are also used for cycle times, $T$.

**Category 1**

Under this category, six possible move cycles for an open-shop problem are defined. The move cycle, $S_I^1$, is similar to $S_1$ in the three-machine flow-shop problem described by Sethi et al. [6], and the problem of finding the best part sequence to be processed using the robot move cycle, $S^1$, is solved trivially [3].

**Lemma 2**

The cycle time of one unit, for the policies under Category 1, are given by:

$$T^1_{I,\sigma(i)}=\hat{T}^1_{I,\sigma(i)}=T^1_{II,\sigma(\iota)}=\hat{T}^1_{II,\sigma(\iota)}=T^1_{III,\sigma(\iota)}$$
$$=\hat{T}^1_{III,\sigma(\iota)}=a_{\sigma(i)}+b_{\sigma(i)}+c_{\sigma(i)}+8\varepsilon+4\delta.$$

***Proof***

Refer to the Appendix, Table A1.

**Category 2**

Under this category, six possible move cycles for a flexible robotic cell problem are defined. The move cycle, $S_I^2$, is similar to $S_2$ in the three-machine flow-shop problem described by Sethi et al. [6] and the problem of finding the best part sequence to be processed, using the robot move cycle, $S^2$, is NP-Complete [4].

**Lemma 3**

The cycle times of one unit for the policies under Category 2 are given by:

$$T^2_{I,\sigma(i)\sigma(i+1)}=8\varepsilon+8\delta+\max\{0,c_{\sigma(i)}-4\delta-2\varepsilon,a_{\sigma(i+1)}-4\delta-2\varepsilon,a_{\sigma(i)}+b_{\sigma(i)},\frac{a_{\sigma(i+1)}+b_{\sigma(i+1)}+c_{\sigma(i+1)}}{2}-6\delta-4\varepsilon\},$$

$$\hat{T}^2_{I,\sigma(i)\sigma(i+1)}=8\varepsilon+8\delta+\max\{0,a_{\sigma(i)}-4\delta-2\varepsilon,c_{\sigma(i+1)}-4\delta-2\varepsilon,c_{\sigma(i)}+b_{\sigma(i)},\frac{a_{\sigma(i+1)}+b_{\sigma(i+1)}+c_{\sigma(i+1)}}{2}-6\delta-4\varepsilon\},$$

$$T^2_{II,\sigma(i)\sigma(i+1)}=8\varepsilon+8\delta+\max\{0,c_{\sigma(i)}-4\delta-2\varepsilon,b_{\sigma(i+1)}-4\delta-2\varepsilon,a_{\sigma(i)}+b_{\sigma(i)},\frac{a_{\sigma(i+1)}+b_{\sigma(i+1)}+c_{\sigma(i+1)}}{2}-6\delta-4\varepsilon\},$$

$$\hat{T}^2_{II,\sigma(i)\sigma(i+1)}=8\varepsilon+8\delta+\max\{0,b_{\sigma(i)}-4\delta-2\varepsilon,c_{\sigma(i+1)}-4\delta-2\varepsilon,a_{\sigma(i)}$$

$$+ c_{\sigma(i)}, \frac{a_{\sigma(i+1)} + b_{\sigma(i+1)} + c_{\sigma(i+1)}}{2} - 6\delta - 4\varepsilon\},$$

$$T^2_{III,\sigma(i)\sigma(i+1)} = 8\varepsilon + 8\delta + \max\{0, b_{\sigma(i)} - 4\delta - 2\varepsilon, a_{\sigma(i+1)} - 4\delta - 2\varepsilon, a_{\sigma(i)} + c_{\sigma(i)}, \frac{a_{\sigma(i+1)} + b_{\sigma(i+1)} + c_{\sigma(i+1)}}{2} - 6\delta - 4\varepsilon\},$$

$$\hat{T}^2_{III,\sigma(i)\sigma(i+1)} = 8\varepsilon + 8\delta + \max\{0, a_{\sigma(i)} - 4\delta - 2\varepsilon, b_{\sigma(i+1)} - 4\delta - 2\varepsilon, b_{\sigma(i)} + c_{\sigma(i)}, \frac{a_{\sigma(i+1)} + b_{\sigma(i+1)} + c_{\sigma(i+1)}}{2} - 6\delta - 4\varepsilon\}.$$

***Proof***

Refer to the Appendix, Table A2.

**Category 3**

Under this category, six possible move cycles for a flexible robotic cell problem are defined. The move cycle, $s^3_I$, is similar to $S^3$ in the three-machine flow-shop problem described by Sethi et al. [6] and the problem of finding the best part sequence to be processed using the robot move cycle, $S^3$, can be solved optimally in $O(n \log n)$ time [3].

**Lemma 4**

The cycle times of one unit for the policies under Category 3 are given by:

$$T^3_{I,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{a_{\sigma(i+1)}, c_{\sigma(i)} - 4\delta - 4\varepsilon, a_{\sigma(i+1)} + b_{\sigma(i+1)} - 4\delta - 2\varepsilon\},$$

$$\hat{T}^3_{I,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{c_{\sigma(i+1)}, a_{\sigma(i)} - 4\delta - 4\varepsilon, c_{\sigma(i+1)} + b_{\sigma(i+1)} - 4\delta - 2\varepsilon\},$$

$$T^3_{II,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{b_{\sigma(i+1)}, c_{\sigma(i)} - 4\delta - 4\varepsilon, a_{\sigma(i+1)} + b_{\sigma(i+1)} - 4\delta - 2\varepsilon\},$$

$$\hat{T}^3_{II,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{c_{\sigma(i+1)}, b_{\sigma(i)} - 4\delta - 4\varepsilon, a_{\sigma(i+1)} + c_{\sigma(i+1)} - 4\delta - 2\varepsilon\},$$

$$T^3_{III,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{a_{\sigma(i+1)}, b_{\sigma(i)} - 4\delta - 4\varepsilon, a_{\sigma(i+1)} + c_{\sigma(i+1)} - 4\delta - 2\varepsilon\},$$

$$\hat{T}^3_{III,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{b_{\sigma(i+1)}, c_{\sigma(i)} - 4\delta - 4\varepsilon, b_{\sigma(i+1)} + c_{\sigma(i+1)} - 4\delta - 2\varepsilon\}.$$

***Proof***

Refer to the Appendix, Table A3.

**Category 4**

Under this category, six possible move cycles for an open-shop problem are defined. The move cycle, $S^4_I$, is similar to $S_4$ in the three-machine flow-shop problem described by [6] and the problem of finding the best part sequence to be processed, using the robot move cycle $S^4$, can be solved optimally in $On \log(n)$ time [3].

**Lemma 5**

The cycle times of one unit for the policies under Category 4 are given by:

$$T^4_{I,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{b_{\sigma(i+1)}, b_{\sigma(i+1)} + c_{\sigma(i)} - 4\delta - 2\varepsilon, a_{\sigma(i+1)} + b_{\sigma(i+1)} - 4\delta - 2\varepsilon\},$$

$$\hat{T}^4_{I,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{b_{\sigma(i+1)}, b_{\sigma(i+1)} + a_{\sigma(i)} - 4\delta - 2\varepsilon, b_{\sigma(i+1)} + c_{\sigma(i+1)} - 4\delta - 2\varepsilon\},$$

$$T^4_{II,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{a_{\sigma(i+1)}, a_{\sigma(i+1)} + c_{\sigma(i)} - 4\delta - 2\varepsilon, a_{\sigma(i+1)} + b_{\sigma(i+1)} - 4\delta - 2\varepsilon\},$$

$$\hat{T}^4_{II,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{a_{\sigma(i+1)}, a_{\sigma(i+1)} + b_{\sigma(i)} - 4\delta - 2\varepsilon, a_{\sigma(i+1)} + c_{\sigma(i+1)} - 4\delta - 2\varepsilon\},$$

$$T^4_{III,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{c_{\sigma(i+1)}, b_{\sigma(i+1)} + c_{\sigma(i)} - 4\delta - 2\varepsilon, a_{\sigma(i+1)} + c_{\sigma(i+1)} - 4\delta - 2\varepsilon\},$$

$$\hat{T}^4_{III,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + \max\{c_{\sigma(i+1)}, c_{\sigma(i+1)} + a_{\sigma(i)} - 4\delta - 2\varepsilon, b_{\sigma(i+1)} + c_{\sigma(i+1)} - 4\delta - 2\varepsilon\}.$$

***Proof***

Refer to the Appendix, Table A4.

**Category 5**

Under this category, six possible move cycles for an open-shop problem are defined. The move cycle, $S_I^5$, is similar to $S_5$ in the three-machine flow-shop problem described by Sethi et al. [6] and the problem of finding the best part sequence to be processed, using the robot move cycle $S^5$, can be solved optimally in $On\log(n)$ time [3].

**Lemma 6**

The cycle times of one unit for the policies under Category 5 are given by:

$$T^5_{I,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + \max\{c_{\sigma(i)}, b_{\sigma(i)} + c_{\sigma(i)} - 4\delta - 2\varepsilon, a_{\sigma(i+1)} - 4\delta - 4\varepsilon\},$$

$$\hat{T}^5_{I,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + \max\{a_{\sigma(i)}, a_{\sigma(i)} + b_{\sigma(i)} - 4\delta - 2\varepsilon, c_{\sigma(i+1)} - 4\delta - 4\varepsilon\},$$

$$T^5_{II,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + \max\{c_{\sigma(i)}, a_{\sigma(i)} + c_{\sigma(i)} - 4\delta - 2\varepsilon, b_{\sigma(i+1)} - 4\delta - 4\varepsilon\},$$

$$\hat{T}^5_{II,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + \max\{b_{\sigma(i)}, a_{\sigma(i)} + b_{\sigma(i)} - 4\delta - 2\varepsilon, c_{\sigma(i+1)} - 4\delta - 4\varepsilon\},$$

$$T^5_{III,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + \max\{b_{\sigma(i)}, b_{\sigma(i)} + c_{\sigma(i)} - 4\delta - 2\varepsilon, a_{\sigma(i+1)} - 4\delta - 4\varepsilon\},$$

$$\hat{T}^5_{III,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + \max\{a_{\sigma(i)}, a_{\sigma(i)} + c_{\sigma(i)} - 4\delta - 2\varepsilon, b_{\sigma(i+1)} - 4\delta - 4\varepsilon\}.$$

***Proof***

Refer to the Appendix, Table A5.

**Category 6**

Under this category, six possible move cycles for an open-shop problem are defined. The move cycle, $S_I^6$, is similar to $S_6$ in the three-machine flow-shop problem described by Sethi et al. [6] and the problem of finding the best part sequence to be processed, using the robot move cycle $S^6$, is NP-complete [4].

**Lemma 7**

The cycle times of one unit for the policies under Category 6 are given by:

$$T^6_{I,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + \max\{0, a_{\sigma(i+2)} - 8\delta - 4\varepsilon, b_{\sigma(i+1)} - 8\delta - 4\varepsilon, c_{\sigma(i)} - 8\delta - 4\varepsilon\},$$

$$\hat{T}^6_{I,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + \max\{0, c_{\sigma(i+2)} - 8\delta - 4\varepsilon, b_{\sigma(i+1)} - 8\delta - 4\varepsilon, a_{\sigma(i)} - 8\delta - 4\varepsilon\},$$

$$T^6_{II,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + \max\{0, b_{\sigma(i+2)} - 8\delta - 4\varepsilon, a_{\sigma(i+1)} - 8\delta - 4\varepsilon, c_{\sigma(i)} - 8\delta - 4\varepsilon\},$$

$$\hat{T}^6_{II,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + \max\{0, c_{\sigma(i+2)} - 8\delta - 4\varepsilon, a_{\sigma(i+1)} - 8\delta - 4\varepsilon, b_{\sigma(i)} - 8\delta - 4\varepsilon\},$$

$$T^6_{III,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + \max\{0, a_{\sigma(i+2)} - 8\delta - 4\varepsilon, c_{\sigma(i+1)} - 8\delta - 4\varepsilon, b_{\sigma(i)} - 8\delta - 4\varepsilon\},$$

$$\hat{T}^6_{III,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + \max\{0, b_{\sigma(i+2)} - 8\delta - 4\varepsilon, c_{\sigma(i+1)} - 8\delta - 4\varepsilon, a_{\sigma(i)} - 8\delta - 4\varepsilon\}.$$

***Proof***

Refer to the Appendix, Table A6.

The computational results of Lemmas 2-7 are shown in Table A7. For the experiments, we considered that the values of $\varepsilon$ and $\delta$ are equal to 1; the processing times for all parts on all machines are uniformly generated in the range [10,100] and the parts are identical.

**Theorem 1**

If in a three-machine robotic cell with identical parts the processing times on all three machines are larger than, or equal to, $8\delta + 4\varepsilon$, and the operations are arranged in such a way that the longest operation is assigned to machine $M_1$, the shortest operation is assigned to machine $M_3$ and the last operation is assigned to machine $M_2$, the policies under Category 6 will have the minimum cycle time.

***Proof***

First, consider the following notations:

$$p^1 = \max\{a, b, c\}, \qquad P^3 = \min\{a, b, c\},$$

$$P^2 = \{a, b, c\} - P^1 - P^3.$$

According to Lemmas 2 to 7, the cycle time of different policies in this case will be as follows:

$$T^1 = P^1 + P^2 + P^3 + 4\delta + 8\varepsilon,$$

$$T^2 = \max\{P^1 + P^2 + 8\delta + 8\varepsilon, \frac{P^1 + P^2 + P^3}{2} + 2\delta + 4\varepsilon\},$$

$$T^3 = P^1 + P^2 + 4\delta + 6\varepsilon,$$

$$T^4 = \max\{P^1 + P^2 + 4\delta + 6\varepsilon, P^2 + P^3 + 4\delta + 4\varepsilon\},$$

$$T^5 = \max\{P^2 + P^3 + 4\delta + 6\varepsilon, P^1 + 4\delta + 4\varepsilon\},$$

$$T^6 = P^1 + 4\delta + 4\varepsilon.$$

By comparing $T^1 - T^6$, we can simply conclude that policies under Category 6 have a minimum cycle time and the proof will be complete.

From here, we will consider policy $S_I^6$.

## DEVELOPING THE MATHEMATICAL MODELS

In this section, we develop a systematic method to produce the necessary mathematical programming formulation for robotic cells. Therefore, first, we model a single-part type problem using Petri-nets and, then, we adapt the mathematical programming approach to the problem. Second, we extend the model to a multiple-part type problem.

### Single-Part Type Problem $\mathrm{ORC}_3|k = 1,\ S_I^6|C_t$

Without loss of generality in the modeling approach, the movements of the robot arm will be restricted to policy $S_I^6$, as shown in Figure 2. The robot arm, at steady state, is located at machine $M_2$, therefore, by coming back to this node, we have a complete cycle for the robot arm. This policy is described in Figure 2. For further formulation of the problem, we need to define the Petri-nets and their related characteristics.

A Petri-net is a quadruple, $PN(P, T, A, W)$, where $P = \{p_1, p_2, \cdots, p_n\}$ is a finite set of places, $T = \{t_1, t_2, \cdots, t_m\}$ is a finite set of transitions, $A \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs and $W : A \rightarrow \{1, 2, 3, \cdots\}$ is a weight function. Every place has an initial marking, $M_0 : P \rightarrow \{0, 1, 2, \cdots\}$. If we assign time to the transitions, we call it a timed Petri-net.

The behavior of many systems can be described by system states and their changes, in order to simulate the dynamic behavior of the system. The marking in a Petri-net is changed, according to the following transition (firing) rule:

1. A transition is said to be enabled, if each input place, $p$ of $t$, is marked at least with $w(p, t)$ tokens, where $w(p, t)$ is the weight of the arc from $p$ to $t$;
2. An enabled transition may or may not be fired (depending on whether or not the event takes place);
3. The firing of an enabled transition, $t$, removes $w(p, t)$ tokens from each input place, $p$ of $t$, and adds $w(t, p)$ tokens to each output place, $p$ of $t$, where $w(t, p)$ is the weight of the arc from $t$ to $p$.

The related timed Petri-net for robot movements under policy $S_I^6$ is shown in Figure 3. All the weights of the arcs are constant and can be ignored. The descriptions of the transitions for this graph, with respective execution times, would be as follows:

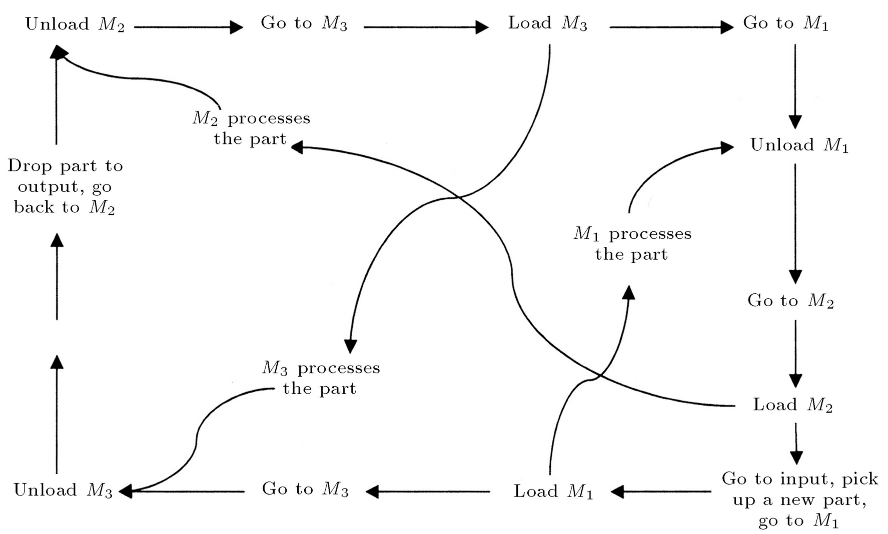


**Figure 2.** the robot movement in three-machine robotic cell under policy $S_I^6$.

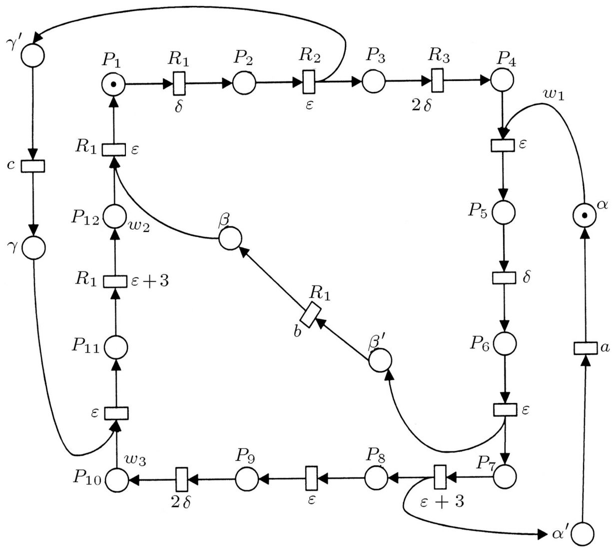


**Figure 3.** Petri net; presentation of policy $S_I^6$ under Category 6.

$R_1$ : go to $M_3(\delta)$,

$R_2$ : load $M_3(\varepsilon)$,

$R_3$ : go to $M_1(2\delta)$,

$RP_1$ : wait at $M_1(w_1^i)$,

$R_4$ : unload $M_1(\varepsilon)$,

$R_5$ : go to $M_2(2\delta)$,

$R_6$ : Load $M_2(\varepsilon)$,

$R_7$ : go to input, pickup a new part, move it to $M_1(\varepsilon + 3\delta)$,

$R_8$ : load $M_1(\varepsilon)$,

$R_9$ : go to $M_3(2\delta)$,

$RP_3$ : wait at $M_3(w_3^i)$,

$R_{10}$ : unload $M_3(\varepsilon)$,

$R_{11}$ : go to output, drop the part, go to $M_2(\varepsilon+3\delta)$,

$RP_2$ : wait at $M_2(w_2^i)$,

$R_{12}$ : unload $M_2(\varepsilon)$,

in which the execution times are as follows:

$s_i$ : Starting time of operating transition $R_i$,

where $i = 1, 2, \cdots, 12$,

$sp_j$ : Starting time of operating transition $RP_j$,

where $j = 1, 2, 3$,

The constraints of three machines process times are as follows:

$\alpha$: Machine $M_1$ has processed the job and is readyto be unloaded;

$\beta$ : Machine $M_2$ has processed the job and is ready to be unloaded;

$\gamma$: Machine $M_3$ has processed the job and is ready to be unloaded.

**Definition**

A marked graph is a Petri-net, such that every place has only one input and only one output.

**Theorem 2**

For a marked graph, wherein every place has $m_i$ tokens (see Figure 4), the following relation, $S_B \geq S_A + m_i C_t$, where $S_A$, $S_B$ are the starting times of transitions $A$ and $B$, respectively, and "$C_t$ is the cycle time", is true.

*Proof:* See [14].

For guaranteeing the liveness of the Petri-net, at the beginning, we put one token in place $p_1$ and one token in place $\alpha$. During the production cycle, the tokens are moving to different places and, after completion of a cycle, the tokens are replaced at the same beginning places; thereafter, another cycle can be repeated. Since the Petri-net model of the problem is a marked graph, based on Theorem 2, the following linear programming

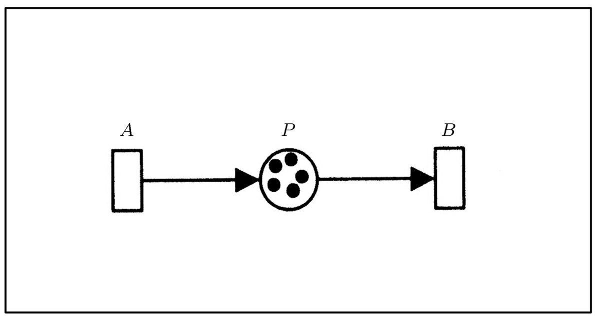


**Figure 4.** The marked graph in Theorem 2.

can be developed.

$$\min\ C_t^6,$$

subject to :

$$p_1 : s_1 - s_{12} + C_t^6 = \varepsilon, \quad (1)$$

$$p_2 : s_2 - s_1 = \delta, \quad (2)$$

$$P_3 : s_3 - s_2 = \varepsilon, \quad (3)$$

$$PP_1 : sp_1 - s_3 = 2\delta, \quad (4)$$

$$p_4 : s_4 - sp_1 - w_1 = 0, \quad (5)$$

$$P_5 : s_5 - s_4 = \varepsilon, \quad (6)$$

$$P_6 : s_6 - s_5 = \delta, \quad (7)$$

$$p_7 : s_7 - s_6 = \varepsilon, \quad (8)$$

$$P_8 : s_8 - s_7 = \varepsilon + 3\delta, \quad (9)$$

$$P_9 : s_9 - s_8 = \varepsilon, \quad (10)$$

$$PP_3 : sp_3 - s_9 = 2\delta, \quad (11)$$

$$P_{10} : s_{10} - sp_3 - w_3 = 0, \quad (12)$$

$$P_{11} : s_{11} - s_{10} = \varepsilon, \quad (13)$$

$$PP_2 : sp_2 - s_{11} = \varepsilon + 3\delta, \quad (14)$$

$$P_{12} : s_{12} - sp_2 - w_2 = 0, \quad (15)$$

$$\alpha : s_4 - s_7 + C_t^6 \geq a + \varepsilon, \quad (16)$$

$$\beta : s_{12} - s_6 \geq b + \varepsilon, \quad (17)$$

$$\gamma : s_{10} - s_2 \geq c + \varepsilon, \quad (18)$$

$$s_i \geq 0,\ i = 1, 2, \cdots, 12, \qquad sp_j \geq 0,\ i = 1, 2, 3,$$

$$w_j \geq 0,\ j = 0, 1, 2, 3.$$

Notice that, to avoid the deadlock in the steady state, it is assumed that machine $M_1$ has processed its job and is waiting for the robot, where machines $M_2$ and $M_3$ are in an idle status and the robot arm is moving one part to machine $M_3$. The above formulation has some significant advantages over the previous model. First, it is simply a network problem that has polynomial time complexity and, second, it computes the starting times for every status of robot movement, which is more convenient than computing the waiting times of the robot arm.

**Multi-Part Type Systems, $\mathrm{ORC}_3|k \geq 2,\ S_I^6|C_t$**

The single part type problem is not a very complicated problem. In this section, a system that allows a multiple part type will be studied. For example, at machine $M_1$, when we want to load a part on the machine, we have to decide which part should be chosen, such that the cycle time is minimized. The same thing can also be achieved for $M_2$ and $M_3$. Based on the choosing gate definition [15], we simply have three choosing gates, as $\alpha$, $\beta$ and $\gamma$. Thus, we can write the following formulations using 0-1 integer variables, $x1_{ij}$, $x2_{ij}$ and $x3_{ij}$, as:

$$\alpha_1 : s_{4,1} - s_{8,n} + C_t = \sum_{i=1}^{n} x1_{\text{in}}(a_i + \varphi_i) + \varepsilon,$$

$$\alpha_j : s_{4,j+1} - s_{8,j} = \sum_{i=1}^{n} x1_{ij}(a_i + \varphi_i) + \varepsilon,$$

$$j = 2, \cdots, n,$$

$$\beta_j : s_{12,j} - s_{6,j} = \sum_{i=1}^{n} x2_{ij}(b_i + \varphi_i) + \varepsilon,$$

$$j = 1, \cdots, n,$$

$$\gamma_j : s_{10,j} - s_{2,j} = \sum_{i=1}^{n} x3_{ij}(c_i + \varphi_i) + \varepsilon,$$

$$j = 1, \cdots, n.$$

In addition, the following feasibility constraints assign a unique positioning for every job:

$$\sum_{i=1}^{n} x1_{ij} = 1, \qquad j = 1, \cdots, n,$$

$$\sum_{j=1}^{n} x1_{ij} = 1, \qquad i = 1, \cdots, n.$$

To keep the sequence of the parts between the machines in the correct order, we have to add the following constraints:

$$x1_{i,j} = x2_{i,j+1}, \qquad i = 1, \cdots, n, \quad j = 1, \cdots, n,$$

$$x2_{i,j} = x3_{i,j+1}, \qquad i = 1, \cdots, n, \quad j = 1, \cdots, n,$$

where it is assumed that $x1_{i,n+1} = x1_{i,1}$, because of the cyclic repetition of parts.

Thus, the complete model for the three-machine robotic cells with multiple-parts would be as follows:

$\min\ C_t^6,$

subject to :

$$p_{1,1}: s_{2,1} - s_{12,n} + C_t = \varepsilon + \delta, \quad j = 2, \cdots, n, \tag{19}$$

$$p_{1,j}: s_{2j} - s_{12j} = \varepsilon + \delta, \quad j = 1, \cdots, n, \tag{20}$$

$$p_{3,j}: s_{4j} - s_{2j} - w_{1j} = \varepsilon + 2\delta, \quad j = 1, \cdots, n, \tag{21}$$

$$p_{5,j}: s_{6j} - s_{4j} = \varepsilon + \delta, \quad j = 1, \cdots, n, \tag{22}$$

$$p_{7,j}: s_{8j} - s_{6j} = 2\varepsilon + 3\delta, \quad j = 1, \cdots, n, \tag{23}$$

$$p_{9,j}: s_{10j} - s_{8j} - w_{3j} = \varepsilon + 2\delta, \quad j = 1, \cdots, n, \tag{24}$$

$$p_{11,j}: s_{12j} - s_{10j} - w_{2j} = 2\varepsilon + 3\delta, \quad j = 1, \cdots, n, \tag{25}$$

$$\alpha_1: s_{4,1} - s_{7,1} + C_t - \sum_{i=1}^{n} x1_{\text{in}}(a_i + \varphi_{a_i}) \geq \varepsilon, \tag{26}$$

$$\alpha_j: s_{4,j} - s_{7,j} - \sum_{i=1}^{n} x1_{ij}(a_i + \varphi_{a_i}) \geq \varepsilon, \quad j = 2, \cdots, n, \tag{27}$$

$$\beta_j: s_{12,j} - s_{6,j} - \sum_{i=1}^{n} x2_{ij}(b_i + \varphi_{b_i}) \geq \varepsilon, \quad j = 1, \cdots, n, \tag{28}$$

$$\gamma_j: s_{10,j} - s_{2,j} - \sum_{i=1}^{n} x3_{ij}(c_i + \varphi_{c_i}) \geq \varepsilon, \quad j = 1, \cdots, n, \tag{29}$$

$$x1_{i,j-1} = x2_{i,j}, \quad i, j = 1, \cdots, n, \tag{30}$$

$$x2_{i,j-1} = x3_{i,j}, \quad i, j = 1, \cdots, n, \tag{31}$$

$$\sum_{i=1}^{n} x1_{ij} = 1, \quad j = 1, \cdots, n, \tag{32}$$

$$\sum_{j=1}^{n} x1_{ij} = 1, \quad i = 1, \cdots, n, \tag{33}$$

$$s_{i,j} \geq 0, \quad i = 1, \cdots, 12, \quad j = 1, 2, \cdots, n,$$

$$w_{kj} \geq 0, \quad k = 0, 1, 2, 3, \quad j = 1, \cdots, n,$$

$$x1, x2, x3 \in \{0, 1\}.$$

This mathematical model can be developed for other policies under Category 6. These models are coded into lingo 8 and run on the Core (TM) 2 Due T7100 processor at 1.8 GHz and Windows vista, using 2 GB of RAM. For the experiments, we consider the values of $\varepsilon$ and $\delta$ as being equal to 1; the processing times for all parts on all machines are uniformly generated in the range [10, 100].

The problem instances are randomly generated as Table A8.

## SOLUTION ALGORITHMS

In a single-part type problem, the parts, which are produced in a cell, are identical. Thus, it is necessary to consider only robot movements to produce the best solution for the problem. In a multiple-part type robotic cell, the formation of MPS is defined, according to the market demand of different products. Therefore, two decisions need to be made: (a) Choosing a robot move sequence and (b) Determining a part sequence. In practice, the MPS can be larger than 50 parts, thus, the sequence of parts in a robotic cell is very important. According to the Sriskandarajah classification scheme [5], for the complexity of this robotic cell scheduling problem, scheduling problems of policies under Category 1 are sequence independent and are trivially solvable (they are $U$-class [5]). Scheduling problems of policies under Categories 3, 4 and 5 can be modeled as the special case of a travelling salesman problem, which is solvable in polynomial time by the Glimor and Gomary algorithm [16] (they are $V1$-class [5]). Scheduling problems of policies under Categories 2 and 6 are NP-Complete (they are W-class [5]). The mathematical model introduced in Section 5 can be used for medium size problems, but, further research is needed to introduce heuristics or meta-heuristics in the solving of large size problems. Baghchi [17] proposed an algorithm to solve $m$-machine cells. By using this algorithm, the sequence of parts in a MPS and robot movement policy that minimizes the 1-unit cycle time, is obtained. This algorithm is as follows:

Step 1 Let $T_u$ denotes the minimum cycle time of a part sequence for the policies in the $U$-class;

Step 2 Use the algorithm of Gilmore and Gomory to solve the part sequencing problem for all policies in a $V1$-class. Let $T_{v1}$ denote the minimum cycle time in the $V1$-class;

Step 3 Use a mathematical model (for small size problems) or a heuristic (for large size problems) to solve the part sequencing problems for all policies in a $W$-class. Let $T_w$ denote the minimum cycle time among $W$-cycles;

Step 4 Find $T_h = \min\{T_u, T_{v1}, T_{v2}, T_w\}$ and its associated policy and part sequence. Terminate.

## CONCLUSION

In this paper, we consider a manufacturing cell, in which a robot loads/ unloads machines and moves parts between machines. In this robotic cell, machines are flexible and are able to do all operations that are necessary for producing all parts. The robotic cell may produce different part types and each part consists of three operations. In this paper, we recognize thirty six potential optimal 1-unit cycles by considering operational flexibility and the policies that have been introduced by Sethi et al. [6]. Twelve policies, which are under categories $S^2$ and $S^6$, are Unary NP-complete. We introduce formulas for calculating the cycle time of thirty six policies when the sequences of parts are given. The formulas achieved by considering the waiting times of a robot for unloading a completed part from machines $M_1$, $M_2$, and $M_3$, and under the condition that one of the categories (i.e. $S^6$) had the minimum cycle time, were identified.

In this paper, a mathematical model using Petri-nets was proposed, to find the parts sequence and robot movements that minimize the cycle time (i.e. maximize throughput). This model has significant advantages over previous model formulations. Based on the single model formulation, a general mathematical model was developed for a multi-part type problem. Finally, an algorithm is introduced to solve this problem. In further works to appear soon, we implemented other issues, such as a no-wait robot environment.

## REFERENCES


1. Asfahl, C.R., *Robots and Manufacturing Automation*, 2nd Ed., New York, John Wiley and Sons (1992).
2. Crama, Y. et al. "Cyclic scheduling in robotic flow shops", *Annals of Operations Research*, **96**, pp. 97-124 (2000).
3. Hall, N.G., Kamoun, H. and Sriskandarajah, C. "Scheduling in robotic cells: Classification, two and three machine cells", *Operations Research*, **45**(2), pp. 421-439 (1997).
4. Hall, N.G., Kamoun, H. and Sriskandarajah, C. "Scheduling in robotic cells: complexity and steady state analysis", *European Journal of Operational Research*, **109**, pp. 43-65 (1998).
5. Sriskandarajah, C. et al. "Scheduling large robotic cells without buffers", *Annals of Operations Research*, **76**, pp. 287-321 (1998).
6. Sethi, S.P. et al. "Sequencing of parts and robot moves in a robotic cell", *International Journal of Flexible Manufacturing Systems*, **4**, pp. 331-358 (1992).
7. Blazewicz, J., Sethi, S.P. and Sriskandarajah, C. "Scheduling of robot moves and parts in a robotic cell", in *Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, Amsterdam, The Netherlands, Elsevier Science Publishers (1989).
8. Crama, Y. and Klundert, V.D. "Cyclic scheduling in 3-machine robotic flow shops", *Journal of Scheduling*, **2**, pp. 35-54 (1999).
9. Brauner, N. and Finke, G. "On a conjecture about robotic cells: New simplified proof for the three-machine case", *INFOR*, **37**(1), pp. 20-36 (1999).
10. Dawande, M, et al. "Sequencing and Scheduling in Robotic Cells: Recent Developments", *Journal of Scheduling*, **8**, pp. 387-426 (2005).
11. Gultekin, H., Akturk, M.S. and Karasan, O.E. "Cyclic scheduling of a 2-machine robotic cell with tooling constraints", *European Journal of Operational Research*, **174**, pp. 777-796 (2006).
12. Gultekin, H., Akturk, M.S. and Karasan, O.E. "Scheduling in a three-machine robotic flexible manufacturing cell", *Computers & Operations Research*, **34**, pp. 2463-2477 (2007).
13. Geismar, H.N., Dawande, M. and Sriskandarajah, C. "Approximation algorithms for k-unit cyclic solutions in robotic cells", *European Journal of Operational Research*, **162**, pp. 291-309 (2005).
14. Maggot, J. "Performance evaluation of concurrent systems using Petri-nets", *Inform. Processing Lett.*, **18**(1), pp. 7-13 (1984).
15. Abadi, I.N.K. "A new formulation for scheduling problems through Petri-nets", in *the Iranian Mathematical Conference*, Tabriz, Iran (1996).
16. Gilmore, P. and Gomory, R. "Sequencing a one-state variable machine: A solvable case of the traveling salesman problem", *Operations Research*, **12**, pp. 675-679 (1964).
17. Bagchi, T.P., Gupta, J.N.D. and Sriskandarajah, C., "A review of TSP based approaches for flow shop scheduling", *European Journal of Operational Research*, **169**, pp. 816-854 (2006).


## APPENDIX

The waiting times relating to Lemmas 2 to 7 are calculated, based on the Petri-nets of each policy, and cycle times are calculated in the following tables.

**Table A1.** Robot movements of Category 1.

| | |
|---|---|
| **Category 1**<br>$S^1 : \{M_3^+, M_1^-, M_2^-, M_3^-, M_3^+\}$<br>**The Initial State:**<br>$E_0 = (\phi_1, \phi_2, \phi_3)$ | |
| **Policy $S_I^1$** | **Cycle Time:** |
| **Robot Movement:**<br>Pickup a new part $p_i$ from $I/O(\varepsilon)$, move it to $M_1(\delta)$, load $p_i$ onto $M_1(\varepsilon)$, wait at $M_1(w_1^i)$, unload $p_i$ from $M_1(\varepsilon)$, move it to $M_2(\delta)$, load $p_i$ onto $M_2(\varepsilon)$, wait at $M_2(w_2^i)$, unload $p_i$ from $M_2(\varepsilon)$, move it to $M_3(\delta)$, load $p_i$ onto $M_3(\varepsilon)$, wait at $M_3(w_3^i)$, unload $p_i$ from $M_3(\varepsilon)$, move it to $I/O(\delta)$, drop the part $p_i$ at $I/O(\varepsilon)$, then start a new cycle by picking up the new part $p_{i+1}$<br>In this policy machine $M_1$ is assigned to the first operation, machine $M_2$ is assigned to the second operation, and machine $M_3$ is assigned to the third operation. | $T^1_{I,\sigma(i)} = w_1^i + w_2^i + w_3^i + 8\varepsilon + 4\delta$<br>$w_1^i = a_{\sigma(i)}$ $w_2^i = b_{\sigma(i)}$ $w_3^i = c_{\sigma(i)}$ |
| **Policy $\hat{S}_I^1$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_I^1$ is similar to $S_I^1$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the second operation, and machine $M_3$ assigned to the first operation. | $\hat{T}^1_{I,\sigma(i)} = w_1^i + w_2^i + w_3^i + 8\varepsilon + 4\delta$<br>$w_1^i = c_{\sigma(i)}$ $w_2^i = b_{\sigma(i)}$ $w_3^i = a_{\sigma(i)}$ |
| **Policy $S_{II}^1$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{II}^1$ is similar to $S_I^1$ but in this policy machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the first, and machine $M_3$ is assigned to the third operation. | $T^1_{II,\sigma(i)} = w_1^i + w_2^i + w_3^i + 8\varepsilon + 4\delta$<br>$w_1^i = b_{\sigma(i)}$ $w_2^i = a_{\sigma(i)}$ $w_3^i = c_{\sigma(i)}$ |
| **Policy $\hat{S}_{II}^1$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{II}^1$ is similar to $S_{II}^1$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the first operation, and machine $M_3$ is assigned to the second operation. | $\hat{T}^1_{II,\sigma(i)} = w_1^i + w_2^i + w_3^i + 8\varepsilon + 4\delta$<br>$w_1^i = c_{\sigma(i)}$ $w_2^i = a_{\sigma(i)}$ $w_3^i = b_{\sigma(i)}$ |
| **Policy $S_{III}^1$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{III}^1$ is similar to $S_I^1$ but in this policy machine $M_1$ is assigned to the first operation, machine $M_2$ is assigned to the third operation, and machine $M_3$ is assigned to the second operation. | $T^1_{III,\sigma(i)} = w_1^i + w_2^i + w_3^i + 8\varepsilon + 4\delta$<br>$w_1^i = a_{\sigma(i)}$ $w_2^i = c_{\sigma(i)}$ $w_3^i = b_{\sigma(i)}$ |
| **Policy $\hat{S}_{III}^1$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{III}^1$ is similar to $S_{III}^1$ but machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the third operation, and machine $M_3$ is assigned to the first operation. | $\hat{T}^1_{III,\sigma(i)} = w_1^i + w_2^i + w_3^i + 8\varepsilon + 4\delta$<br>$w_1^i = b_{\sigma(i)}$ $w_2^i = c_{\sigma(i)}$ $w_3^i = a_{\sigma(i)}$ |

**Table A2.** Robot movements of Category 2.

| | |
|---|---|
| **Category 2**<br>$S^2 : \{M_3^+, M_1^-, M_3^-, M_2^-, M_3^+\}$<br>**The Initial State:**<br>$E_0 = (\phi_1, M_2, \phi_3)$ | |
| **Policy $S_I^2$** | **Cycle Time:** |
| **Robot Movement:**<br>Pickup part $p_{i+1}$ from $I/(\varepsilon)O$, move it to $M_1(\delta)$, load $p_{i+1}$ on $M_1(\varepsilon)$, go to $M_2(\delta)$, if necessary wait at $M_2(w_2^i)$, unload $p_i$ from $M_2(\varepsilon)$, move it to $M_3(\delta)$, load $p_i$ onto $M_3(\varepsilon)$, go to $M_1(2\delta)$, if necessary wait at $M_1(w_1^{i+1})$, unload $p_{i+1}$ from $M_1(\varepsilon)$, move it to $M_2(\delta)$, load $p_{i+1}$ onto $M_2(\varepsilon)$, go to $M_3(\delta)$, if necessary wait at $M_3(w_3^i)$, unload $p_i$ from $M_3(\varepsilon)$, move it to $I/O(\delta)$, drop $p_i$ at $I/O(\varepsilon)$, start a new cycle by picking up part $p_{i+2}$ | $T_{I,\sigma(i)\sigma(i+1)}^2 = 8\varepsilon + 8\delta + w_2^i + w_1^{i+1} + w_3^i$<br>$w_2^i = \max\{0, b_{\sigma(i)} - w_3^{i-1} - 4\delta - 4\varepsilon\}$<br>$w_1^{i+1} = \max\{0, a_{\sigma(i+1)} - w_2^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, c_{\sigma(i)} - w_1^{i+1} - 4\delta - 2\varepsilon\}$ |
| **Policy $\hat{S}_I^2$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_I^2$ is similar to $S_I^2$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the second operation, and machine $M_3$ assigned to the first operation. | $\hat{T}_{I,\sigma(i)\sigma(i+1)}^2 = 8\varepsilon + 8\delta + w_2^i + w_1^{i+1} + w_3^i$<br>$w_2^i = \max\{0, b_{\sigma(i)} - w_3^{i-1} - 4\delta - 4\varepsilon\}$<br>$w_1^{i+1} = \max\{0, c_{\sigma(i+1)} - w_2^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, a_{\sigma(i)} - w_1^{i+1} - 4\delta - 2\varepsilon\}$ |
| **Policy $S_{II}^2$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{II}^2$ is similar to $S_I^2$ but in this policy machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the first, and machine $M_3$ is assigned to the third operation. | $T_{II,\sigma(i)\sigma(i+1)}^2 = 8\varepsilon + 8\delta + w_2^i + w_1^{i+1} + w_3^i$<br>$w_2^i = \max\{0, a_{\sigma(i)} - w_3^{i-1} - 4\delta - 4\varepsilon\}$<br>$w_1^{i+1} = \max\{0, b_{\sigma(i+1)} - w_2^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, c_{\sigma(i)} - w_1^{i+1} - 4\delta - 2\varepsilon\}$ |
| **Policy $\hat{S}_{II}^2$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{II}^2$ is similar to $S_{II}^2$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the first operation, and machine $M_3$ is assigned to the second operation. | $\hat{T}_{II,\sigma(i)\sigma(i+1)}^2 = 8\varepsilon + 8\delta + w_2^i + w_1^{i+1} + w_3^i$<br>$w_2^i = \max\{0, a_{\sigma(i)} - w_3^{i-1} - 4\delta - 4\varepsilon\}$<br>$w_1^{i+1} = \max\{0, c_{\sigma(i+1)} - w_2^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, b_{\sigma(i)} - w_1^{i+1} - 4\delta - 2\varepsilon\}$ |
| **Policy $S_{III}^2$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{III}^2$ is similar to $S_I^2$ but in this policy machine $M_1$ is assigned to the first operation, machine $M_2$ is assigned to the third and, machine $M_3$ is assigned to the second operation. | $T_{III,\sigma(i)\sigma(i+1)}^2 = 8\varepsilon + 8\delta + w_2^i + w_1^{i+1} + w_3^i$<br>$w_2^i = \max\{0, c_{\sigma(i)} - w_3^{i-1} - 4\delta - 4\varepsilon\}$<br>$w_1^{i+1} = \max\{0, a_{\sigma(i+1)} - w_2^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, b_{\sigma(i)} - w_1^{i+1} - 4\delta - 2\varepsilon\}$ |
| **Policy $\hat{S}_{III}^2$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{III}^2$ is similar to $S_{III}^2$ but machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the third operation, and machine $M_3$ is assigned to the first operation. | $\hat{T}_{III,\sigma(i)\sigma(i+1)}^2 = 8\varepsilon + 8\delta + w_2^i + w_1^{i+1} + w_3^i$<br>$w_2^i = \max\{0, a_{\sigma(i)} - w_3^{i-1} - 4\delta - 4\varepsilon\}$<br>$w_1^{i+1} = \max\{0, c_{\sigma(i+1)} - w_2^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, b_{\sigma(i)} - w_1^{i+1} - 4\delta - 2\varepsilon\}$ |

**Table A3.** Robot movements of Category 3.

| | |
|---|---|
| **Category 3**<br>$S^3: \{M_3^+, M_3^-, M_1^-, M_2^-, M_3^+\}$<br>**The Initial State:**<br>$E_0 = (\phi_1, \phi_2, M_3)$ | |
| **Policy $S_I^3$** | **Cycle Time:** |
| **Robot Movement:**<br>Pickup part $P_{i+1}$ from $I/O(\varepsilon)$ move it to $M_1(\delta)$ load $P_{i+1}$ onto $M_1(\varepsilon)$ wait at $M_1(W_1^{i+1})$, unload $P_{i+1}$ from $M_1(\varepsilon)$ move it to $M_2(\delta)$ load $P_{i+1}$ onto $M_2(\varepsilon)$ go to $M_3(\delta)$ if necessary wait at $M_3(W_3^i)$, unload $P_i$ from $M_3(\varepsilon)$, move it to $I/O(\delta)$, drop $P_i$ at $I/O(\varepsilon)$ go to $M_2(2\delta)$, if necessary wait at $M_2(W_2^{i+1})$, unload $P_{i+1}$ from $M_2(\varepsilon)$ move it to $M_3(\delta)$, load $P_{i+1}$ onto $M_3(\varepsilon)$ go to $I/O(\delta)$then start a new cycle by picking up the part $P_{i+2}$. | $T^3_{I,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = a_{\sigma(i+1)}$<br>$w_2^{i+1} = \max\{0, b_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, c_{\sigma(i)} - a_{\sigma(i+1)} - 4\delta - 4\varepsilon\}$ |
| **Policy $\hat{S}_I^3$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_I^3$ is similar to $S_I^3$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the second operation, and machine $M_3$ assigned to the first operation. | $\hat{T}^3_{I,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = c_{\sigma(i+1)}$<br>$w_2^{i+1} = \max\{0, b_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, a_{\sigma(i)} - c_{\sigma(i+1)} - 4\delta - 4\varepsilon\}$ |
| **Policy $S_{II}^3$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{II}^3$ is similar to $S_I^3$ but in this policy machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the first, and machine $M_3$ is assigned to the third operation. | $T^3_{II,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = b_{\sigma(i+1)}$<br>$w_2^{i+1} = \max\{0, a_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, c_{\sigma(i)} - b_{\sigma(i+1)} - 4\delta - 4\varepsilon\}$ |
| **Policy $\hat{S}_{II}^3$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{II}^3$ is similar to $S_{II}^3$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the first operation, and machine $M_3$ is assigned to the second operation. | $\hat{T}^3_{II,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = c_{\sigma(i+1)}$<br>$w_2^{i+1} = \max\{0, a_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, b_{\sigma(i)} - c_{\sigma(i+1)} - 4\delta - 4\varepsilon\}$ |
| **Policy $S_{III}^3$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{III}^3$ is similar to $S_I^3$ but in this policy machine $M_1$ is assigned to the first operation, machine $M_2$ is assigned to the third and, machine $M_3$ is assigned to the second operation. | $T^3_{III,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = a_{\sigma(i+1)}$<br>$w_2^{i+1} = \max\{0, c_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, b_{\sigma(i)} - a_{\sigma(i+1)} - 4\delta - 4\varepsilon\}$ |
| **Policy $\hat{S}_{III}^3$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{III}^3$ is similar to $S_{III}^3$ but machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the third operation, and machine $M_3$ is assigned to the first operation. | $\hat{T}^3_{III,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = b_{\sigma(i+1)}$<br>$w_2^{i+1} = \max\{0, c_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_3^i = \max\{0, c_{\sigma(i)} - b_{\sigma(i+1)} - 4\delta - 4\varepsilon\}$ |

**Table A4.** Robot movements of Category 4.

| | |
|---|---|
| **Category 4**<br>$S^4 : \{M_3^+, M_2^-, M_3^-, M_1^-, M_3^+\}$<br>**The Initial State:**<br>$E_0 = (\phi_1, \phi_2, M_3)$ | |
| **Policy $S_I^4$** | **Cycle Time:** |
| **Robot Movement:**<br>Pickup part $P_{i+1}$ from $I/O(\varepsilon)$, move it to $M_1(\delta)$, load $P_{i+1}$ onto $M_1(\varepsilon)$, go to $M_3(2\delta)$, if necessary wait at $M_3(w_3^i)$, unload $P_i$ from $M_3(\varepsilon)$, move it to $I/O(\delta)$, drop $P_i$ at $I/O(\varepsilon)$, go to $M_1(\delta)$, if necessary wait at $M_1(w_1^{i+1})$, unload $P_{i+1}$ from $M_1(\varepsilon)$, move it to $M_2(\delta)$, load $P_{i+1}$ onto $M_2(\varepsilon)$, wait at $M_2(w_2^{i+1})$, unload $P_{i+1}$ from $M_2(\varepsilon)$, move it to $M_3(\delta)$, load $P_{i+1}$ onto $M_3(\varepsilon)$, go to $I/O(\delta)$, then start a new cycle by picking up the part $P_{i+2}$. | $T^4_{I,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = \max\{0, a_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_2^{i+1} = b_{\sigma(i+1)}$<br>$w_3^i = \max\{0, c_{\sigma(i)} - 4\delta - 2\varepsilon\}$ |
| **Policy $\hat{S}_I^4$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_I^4$ is similar to $S_I^4$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the second operation, and machine $M_3$ assigned to the first operation. | $\hat{T}^4_{I,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = \max\{0, c_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_2^{i+1} = b_{\sigma(i+1)}$<br>$w_3^i = \max\{0, a_{\sigma(i)} - 4\delta - 2\varepsilon\}$ |
| **Policy $S_{II}^4$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{II}^4$ is similar to $S_I^4$ but in this policy machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the first, and machine $M_3$ is assigned to the third operation. | $T^4_{II,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = \max\{0, b_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_2^{i+1} = a_{\sigma(i+1)}$<br>$w_3^i = \max\{0, c_{\sigma(i)} - 4\delta - 2\varepsilon\}$ |
| **Policy $\hat{S}_{II}^4$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{II}^4$ is similar to $S_{II}^4$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the first operation, and machine $M_3$ is assigned to the second operation. | $T^4_{II,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = \max\{0, c_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_2^{i+1} = a_{\sigma(i+1)}$<br>$w_3^i = \max\{0, b_{\sigma(i)} - 4\delta - 2\varepsilon\}$ |
| **Policy $S_{III}^4$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{III}^4$ is similar to $S_I^4$ but in this policy machine $M_1$ is assigned to the first operation, machine $M_2$ is assigned to the third operation and, machine $M_3$ is assigned to the second operation. | $T^4_{III,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = \max\{0, a_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_2^{i+1} = c_{\sigma(i+1)}$<br>$w_3^i = \max\{0, b_{\sigma(i)} - 4\delta - 2\varepsilon\}$ |
| **Policy $\hat{S}_{III}^4$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{III}^4$ is similar to $S_{III}^4$ but machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the third operation, and machine $M_3$ is assigned to the first operation. | $\hat{T}^4_{III,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^{i+1} + w_3^i$<br>$w_1^{i+1} = \max\{0, b_{\sigma(i+1)} - w_3^i - 4\delta - 2\varepsilon\}$<br>$w_2^{i+1} = c_{\sigma(i+1)}$<br>$w_3^i = \max\{0, a_{\sigma(i)} - 4\delta - 2\varepsilon\}$ |

**Table A5.** Robot movements of Category 5.

| | |
|---|---|
| **Category 5**<br>$S^5: \{M_3^+, M_2^-, M_1^-, M_3^-, M_3^+\}$<br>**The Initial State:**<br>$E_0 = (\phi_1, M_2, \phi_3)$ | |
| **Policy $S_I^5$** | **Cycle Time:** |
| **Robot Movement:**<br>Pickup part $P_{i+1}$ from $I/O(\varepsilon)$ move it to $M_1(\delta)$ load $P_{i+1}$ onto $M_1(\varepsilon)$ go to $M_2(\delta)$ if necessary wait at $M_2(w_2^i)$, unload $P_i$ from $M_2(\varepsilon)$ move it to $M_3(\delta)$, load $P_i$ onto $M_3(\varepsilon)$, wait at $M_3(w_3^i)$, unload $P_i$ from $M_3(\varepsilon)$ move it to $I/O(\delta)$ drop $P_i$ at $I/O(\varepsilon)$ go to $M_1(\delta)$ if necessary wait at $M_1(w_1^{i+1})$, unload $P_{i+1}$ from $M_3(\varepsilon)$ move it to $M_2(\delta)$, load $P_{i+1}$ onto $M_2(\varepsilon)$ go to $I/O(2\delta)$ then start a new cycle by picking up the part $P_{i+2}$. | $T^5_{I,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^i + w_3^i$<br>$w_1^{i+1} = \max\{0., a_{\sigma(i+1)} - c_{\sigma(i)} - w_2^i - 4\delta - 4\varepsilon\}$<br>$w_2^i = \max\{0, b_{\sigma(i)} - 4\delta - 4\varepsilon\}$<br>$w_3^i = c_{\sigma(i)}$ |
| **Policy $\hat{S}_I^5$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_I^5$ is similar to $S_I^5$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the second operation, and machine $M_3$ assigned to the first operation. | $\hat{T}^5_{I,\sigma(i)\sigma(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^i + w_3^i$<br>$w_1^{i+1} = \max\{0., c_{\sigma(i+1)} - a_{\sigma(i)} - w_2^i - 4\delta - 4\varepsilon\}$<br>$w_2^i = \max\{0, b_{\sigma(i)} - 4\delta - 4\varepsilon\}$<br>$w_3^i = a_{\sigma(i)}$ |
| **Policy $S_{II}^5$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{II}^5$ is similar to $S_I^5$ but in this policy machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the first, and machine $M_3$ is assigned to the third operation. | $T^5_{II,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^i + w_3^i$<br>$w_1^{i+1} = \max\{0., b_{\sigma(i+1)} - c_{\sigma(i)} - w_2^i - 4\delta - 4\varepsilon\}$<br>$w_2^i = \max\{0, a_{\sigma(i)} - 4\delta - 4\varepsilon\}$<br>$w_3^i = c_{\sigma(i)}$ |
| **Policy $\hat{S}_{II}^5$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{II}^5$ is similar to $S_{II}^5$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the first operation, and machine $M_3$ is assigned to the second operation. | $\hat{T}^5_{II,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^i + w_3^i$<br>$w_1^{i+1} = \max\{0., c_{\sigma(i+1)} - b_{\sigma(i)} - w_2^i - 4\delta - 4\varepsilon\}$<br>$w_2^i = \max\{0, a_{\sigma(i)} - 4\delta - 4\varepsilon\}$<br>$w_3^i = b_{\sigma(i)}$ |
| **Policy $S_{III}^5$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{III}^5$ is similar to $S_I^5$ but in this policy machine $M_1$ is assigned to the first operation, machine $M_2$ is assigned to the third operation and, machine $M_3$ is assigned to the second operation. | $T^5_{III,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^i + w_3^i$<br>$w_1^{i+1} = \max\{0., a_{\sigma(i+1)} - b_{\sigma(i)} - w_2^i - 4\delta - 4\varepsilon\}$<br>$w_2^i = \max\{0, c_{\sigma(i)} - 4\delta - 4\varepsilon\}$<br>$w_3^i = b_{\sigma(i)}$ |
| **Policy $\hat{S}_{III}^5$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{III}^5$ is similar to $S_{III}^5$ but machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the third operation and machine $M_3$ is assigned to the first operation. | $\hat{T}^5_{III,\sigma(i)(i+1)} = 8\delta + 8\varepsilon + w_1^{i+1} + w_2^i + w_3^i$<br>$w_1^{i+1} = \max\{0., b_{\sigma(i+1)} - a_{\sigma(i)} - w_2^i - 4\delta - 4\varepsilon\}$<br>$w_2^i = \max\{0, c_{\sigma(i)} - 4\delta - 4\varepsilon\}$<br>$w_3^i = a_{\sigma(i)}$ |

**Table A6.** Robot movements of Category 6.

| | |
|---|---|
| **Category 6**<br>$S^6 : \{M_3^+, M_3^-, M_2^-, M_1^-, M_3^+\}$<br>**The Initial State:**<br>$E_0 = (\phi_1, M_2, M_3)$ | |
| **Policy $S_I^6$** | **Cycle Time:** |
| **Robot Movement:**<br>Pickup part $P_{i+2}$ from $I/O(\varepsilon)$, move it to $M_1(\delta)$, load $P_{i+2}$ onto $M_1(\varepsilon)$, go to $M_3(2\delta)$, if necessary wait at $M_3(w_3^i)$, unload $P_i$ from $M_3(\varepsilon)$, move it to $I/O(\delta)$, drop $P_i$ at $I/O(\varepsilon)$, go to $M_2(2\delta)$, if necessary wait at $M_2(w_2^{i+1})$, unload $P_{i+1}$ from $M_2(\varepsilon)$, move it to $M_3(\delta)$, load $P_{i+1}$ onto $M_3(\varepsilon)$, go to $M_1(2\delta)$, if necessary wait at $M_1(w_1^{i+2})$, unload $P_{i+2}$ from $M_1(\varepsilon)$, move it to $M_2(\delta)$, load $P_{i+2}$ onto $M_2(\varepsilon)$, go to $I/O(2\delta)$, then start a new cycle by picking up the part $P_{i+3}$. | $T^6_{I,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + w_1^{i+2} + w_2^{i+1} + w_3^i$<br>$w_1^{i+2} = \max\{0, a_{\sigma(i+2)} - w_2^{i+1} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_2^{i+1} = \max\{0, b_{\sigma(i+1)} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_3^i = \max\{0, c_{\sigma(i)} - w_1^{i+2} - 8\delta - 4\varepsilon\}$ |
| **Policy $\hat{S}_I^6$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_I^6$ is similar to $S_I^6$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the second operation, and machine $M_3$ assigned to the first operation. | $\hat{T}^6_{I,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + w_1^{i+2} + w_2^{i+1} + w_3^i$<br>$w_1^{i+2} = \max\{0, c_{\sigma(i+2)} - w_2^{i+1} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_2^{i+1} = \max\{0, b_{\sigma(i+1)} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_3^i = \max\{0, a_{\sigma(i)} - w_1^{i+2} - 8\delta - 4\varepsilon\}$ |
| **Policy $S_{II}^6$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{II}^6$ is similar to $S_I^6$ but in this policy machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the first, and machine $M_3$ is assigned to the third operation. | $T^6_{II,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + w_1^{i+2} + w_2^{i+1} + w_3^i$<br>$w_1^{i+2} = \max\{0, b_{\sigma(i+2)} - w_2^{i+1} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_2^{i+1} = \max\{0, a_{\sigma(i+1)} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_3^i = \max\{0, c_{\sigma(i)} - w_1^{i+2} - 8\delta - 4\varepsilon\}$ |
| **Policy $\hat{S}_{II}^6$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{II}^6$ is similar to $S_{II}^6$ but in this policy machine $M_1$ is assigned to the third operation, machine $M_2$ is assigned to the first operation, and machine $M_3$ is assigned to the second operation. | $\hat{T}^6_{II,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + w_1^{i+2} + w_2^{i+1} + w_3^i$<br>$w_1^{i+2} = \max\{0, c_{\sigma(i+2)} - w_2^{i+1} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_2^{i+1} = \max\{0, a_{\sigma(i+1)} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_3^i = \max\{0, b_{\sigma(i)} - w_1^{i+2} - 8\delta - 4\varepsilon\}$ |
| **Policy $S_{III}^6$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $S_{III}^6$ is similar to $S_I^6$ but in this policy machine $M_1$ is assigned to the first operation, machine $M_2$ is assigned to the third operation and, machine $M_3$ is assigned to the second operation. | $T^6_{III,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + w_1^{i+2} + w_2^{i+1} + w_3^i$<br>$w_1^{i+2} = \max\{0, a_{\sigma(i+2)} - w_2^{i+1} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_2^{i+1} = \max\{0, c_{\sigma(i+1)} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_3^i = \max\{0, b_{\sigma(i)} - w_1^{i+2} - 8\delta - 4\varepsilon\}$ |
| **Policy $\hat{S}_{III}^6$** | **Cycle Time:** |
| **Robot Movement:**<br>Robot movement in $\hat{S}_{III}^6$ is similar to $S_{III}^6$ but machine $M_1$ is assigned to the second operation, machine $M_2$ is assigned to the third operation, and machine $M_3$ is assigned to the first operation. | $\hat{T}^6_{III,\sigma(i)\sigma(i+1)\sigma(i+2)} = 12\delta + 8\varepsilon + w_1^{i+2} + w_2^{i+1} + w_3^i$<br>$w_1^{i+2} = \max\{0, b_{\sigma(i+2)} - w_2^{i+1} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_2^{i+1} = \max\{0, c_{\sigma(i+1)} - w_3^i - 8\delta - 4\varepsilon\}$<br>$w_3^i = \max\{0, a_{\sigma(i)} - w_1^{i+2} - 8\delta - 4\varepsilon\}$ |

**Table A7.** Computational results for identical part type problem under six categories.

| Problem Condition | Problem Instance | Category 1 | Category 2 | Category 3 | Category 4 | Category 5 | Category 6 | Best Cycle Time | Best Policy |
|---|---|---|---|---|---|---|---|---|---|
| $a_i \geq b_i \geq c_i$ | I01 | 156 | 90 | 114 | 114 | 84 | 78 | 78 | Category 6 |
| | I02 | 142 | 106 | 118 | 118 | 104 | 104 | 104 | Category 6 |
| | I03 | 175 | 110 | 125 | 125 | 104 | 77 | 77 | Category 6 |
| | I04 | 152 | 104 | 102 | 102 | 98 | 60 | 60 | Category 6 |
| | I05 | 141 | 104 | 122 | 122 | 102 | 102 | 102 | Category 6 |
| $a_i \geq c_i \geq b_i$ | I06 | 197 | 126 | 131 | 131 | 120 | 83 | 83 | Category 6 |
| | I07 | 110 | 79 | 92 | 92 | 77 | 77 | 77 | Category 6 |
| | I08 | 66 | 42 | 52 | 52 | 40 | 40 | 40 | Category 6 |
| | I09 | 192 | 109 | 121 | 121 | 107 | 107 | 107 | Category 6 |
| | I10 | 125 | 75 | 96 | 96 | 73 | 73 | 73 | Category 6 |
| $b_i \geq a_i \geq c_i$ | I11 | 120 | 73 | 69 | 77 | 69 | 59 | 59 | Category 6 |
| | I12 | 200 | 114 | 110 | 138 | 110 | 98 | 98 | Category 6 |
| | I13 | 161 | 100 | 96 | 105 | 96 | 73 | 73 | Category 6 |
| | I14 | 159 | 87 | 85 | 116 | 85 | 85 | 85 | Category 6 |
| | I15 | 214 | 141 | 137 | 143 | 137 | 85 | 85 | Category 6 |
| $b_i \geq c_i \geq a_i$ | I16 | 142 | 92 | 90 | 113 | 90 | 90 | 90 | $S^3_{III}$, $\hat{S}^3_{III}$, $S^5_{II}$, Category 6 |
| | I17 | 180 | 102 | 98 | 132 | 98 | 90 | 90 | Category 6 |
| | I18 | 187 | 104 | 102 | 119 | 102 | 102 | 102 | $S^3_{III}$, $\hat{S}^3_{III}$, $S^5_{II}$, Category 6 |
| | I19 | 155 | 98 | 96 | 115 | 96 | 96 | 96 | $S^3_{III}$, $\hat{S}^3_{III}$, $S^5_{II}$, Category 6 |
| | I20 | 157 | 92 | 88 | 112 | 88 | 77 | 77 | Category 6 |
| $c_i \geq a_i \geq b_i$ | I21 | 223 | 137 | 131 | 133 | 145 | 98 | 98 | Category 6 |
| | I22 | 188 | 113 | 107 | 109 | 118 | 87 | 87 | Category 6 |
| | I23 | 173 | 111 | 105 | 107 | 118 | 74 | 74 | Category 6 |
| | I24 | 159 | 108 | 106 | 61 | 128 | 106 | 61 | $S^5_I$, $\hat{S}^5_I$ |
| | I25 | 86 | 58 | 52 | 53 | 53 | 40 | 40 | Category 6 |
| $c_i \geq b_i \geq a_i$ | I26 | 205 | 134 | 128 | 130 | 137 | 83 | 83 | Category 6 |
| | I27 | 222 | 128 | 122 | 124 | 154 | 106 | 106 | Category 6 |
| | I28 | 170 | 102 | 96 | 98 | 125 | 80 | 80 | Category 6 |
| | I29 | 132 | 74 | 72 | 68 | 94 | 72 | 72 | $S^3_I$, $\hat{S}^3_I$, $S^3_{II}$, $\hat{S}^3_{II}$, Category 6 |
| | I30 | 239 | 162 | 156 | 158 | 159 | 89 | 89 | Category 6 |

**Table A8.** Computational results for deferent type parts problems under category 6.

| No. of Parts | Problem Instance | Problem Condition | $S_I^6$ | | $\hat{S}_I^6$ | | $S_{II}^6$ | | $\hat{S}_{II}^6$ | | $S_{III}^6$ | | $\hat{S}_{III}^6$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | OFV[a] | CPU Time[b] | OFV[a] | CPU Time | OFV[a] | CPU Time | OFV[a] | CPU Time | OFV[a] | CPU Time | OFV[a] | CPU Time |
| 5 | D01 | $a_i \geq b_i \geq c_i$ | 483 | $< 1$ | 483 | $< 1$ | 483 | $< 1$ | 483 | $< 1$ | 483 | $< 1$ | 483 | $< 1$ |
| | D02 | $a_i \geq c_i \geq b_i$ | 435 | $< 1$ | 435 | $< 1$ | 441 | $< 1$ | 441 | $< 1$ | 443 | $< 1$ | 443 | $< 1$ |
| | D03 | $b_i \geq a_i \geq c_i$ | 363 | $< 1$ | 363 | $< 1$ | 363 | $< 1$ | 363 | $< 1$ | 363 | $< 1$ | 363 | $< 1$ |
| | D04 | $b_i \geq c_i \geq a_i$ | 459 | $< 1$ | 459 | $< 1$ | 459 | $< 1$ | 459 | $< 1$ | 459 | $< 1$ | 459 | $< 1$ |
| | D05 | $c_i \geq a_i \geq b_i$ | 454 | $< 1$ | 454 | $< 1$ | 458 | $< 1$ | 458 | $< 1$ | 458 | $< 1$ | 458 | $< 1$ |
| | D06 | $c_i \geq b_i \geq a_i$ | 404 | $< 1$ | 404 | $< 1$ | 397 | $< 1$ | 397 | $< 1$ | 404 | $< 1$ | 404 | $< 1$ |
| | D07 | Unconditional case | 321 | $< 1$ | 321 | $< 1$ | 323 | $< 1$ | 323 | $< 1$ | 321 | $< 1$ | 321 | $< 1$ |
| 10 | D08 | $a_i \geq b_i \geq c_i$ | 754 | 1 | 754 | 2 | 754 | 1 | 754 | 1 | 754 | 1 | 754 | $< 1$ |
| | D09 | $a_i \geq c_i \geq b_i$ | 763 | $< 1$ | 763 | $< 1$ | 763 | 1 | 763 | 1 | 763 | 1 | 763 | $< 1$ |
| | D10 | $b_i \geq a_i \geq c_i$ | 910 | $< 1$ | 910 | 1 | 910 | 1 | 910 | $< 1$ | 910 | $< 1$ | 846 | $< 1$ |
| | D11 | $b_i \geq c_i \geq a_i$ | 825 | 1 | 825 | $< 1$ | 825 | $< 1$ | 825 | $< 1$ | 825 | 1 | 765 | 1 |
| | D12 | $c_i \geq a_i \geq b_i$ | 907 | $< 1$ | 907 | $< 1$ | 907 | $< 1$ | 907 | $< 1$ | 907 | $< 1$ | 907 | $< 1$ |
| | D13 | $c_i \geq b_i \geq a_i$ | 753 | 1 | 752 | $< 1$ | 753 | $< 1$ | 753 | 1 | 753 | $< 1$ | 753 | 1 |
| | D14 | Unconditional case | 739 | 232 | 739 | 186 | 734 | 211 | 734 | 241 | 728 | $< 1$ | 666 | 7 |
| 15 | D15 | $a_i \geq b_i \geq c_i$ | 1312 | $< 1$ | 1312 | $< 1$ | 1321 | 1 | 3121 | 1 | 1312 | $< 1$ | 1312 | $< 1$ |
| | D16 | $a_i \geq c_i \geq b_i$ | 1272 | $< 1$ | 1272 | 1 | 1272 | 1 | 1272 | 1 | 1272 | 3 | 1272 | 1 |
| | D17 | $b_i \geq a_i \geq c_i$ | 1212 | 1 | 1212 | 1 | 1212 | 1 | 1212 | 1 | 1212 | $< 1$ | 1212 | 1 |
| | D18 | $b_i \geq c_i \geq a_i$ | 1352 | $< 1$ | 1352 | 1 | 1352 | $< 1$ | 1352 | $< 1$ | 1352 | 1 | 1352 | 1 |
| | D19 | $c_i \geq a_i \geq b_i$ | 1331 | $< 1$ | 1331 | 1 | 1331 | 1 | 1331 | 1 | 1331 | 1 | 1331 | 1 |
| | D20 | $c_i \geq b_i \geq a_i$ | 1222 | 3 | 1222 | 1 | 1222 | 1 | 1222 | $< 1$ | 1223 | 7200[c] | 1223 | 7200[c] |
| | D21 | Unconditional case | 1086 | 7200[c] | 1095 | 7200[c] | 1095 | 7200[c] | 1099 | 7200[c] | 1075 | 7200[c] | 1072 | 7200[c] |

a: Objective Function Value (Cycle Time),

b: All times are in second,

c: Denotes that the Lingo interrupted after this time and the best achieved value was reported.