Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 34 (2007) 2463–2477
computers & operations research
www.elsevier.com/locate/cor

Scheduling in a three-machine robotic flexible manufacturing cell

Hakan Gultekin, M. Selim Akturk*, Oya Ekin Karasan

Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey

Available online 26 October 2005

Abstract

In this study, we consider a flexible manufacturing cell (FMC) processing identical parts on which the loading and unloading of machines are made by a robot. The machines used in FMCs are predominantly CNC machines and these machines are flexible enough for performing several operations provided that the required tools are stored in their tool magazines. Traditional research in this area considers a flowshop type system. The current study relaxes this flowshop assumption which unnecessarily limits the number of alternatives. In traditional robotic cell scheduling literature, the processing time of each part on each machine is a known parameter. However, in this study the processing times of the parts on the machines are decision variables. Therefore, we investigated the productivity gain attained by the additional flexibility introduced by the FMCs. We propose new lower bounds for the 1-unit and 2-unit robot move cycles (for which we present a completely new procedure to derive the activity sequences of 2-unit cycles in a three-machine robotic cell) under the new problem domain for the flowshop type robot move cycles. We also propose a new robot move cycle which is a direct consequence of process and operational flexibility of CNC machines. We prove that this proposed cycle dominates all 2-unit robot move cycles and present the regions where the proposed cycle dominates all 1-unit cycles. We also present a worst case performance bound of using this proposed cycle.

Keywords: Robotic cell; Cyclic scheduling; Automated manufacturing

1. Introduction

Industrial robots are used extensively in manufacturing companies performing tasks ranging from assembly to testing and inspection. Material handling is one of the important applications of robots. A manufacturing cell in which loading and unloading operations are made by robots is called a robotic cell. These kinds of robots are used extensively in chemical, electronic and metal cutting industries. In this study, we restrict ourselves to robotic cells used in machining applications. The machines used in such systems are predominantly CNC machines and they are capable of performing several different operations by fast and inexpensive tool changes. Consequently, we assume that, each part to be processed has a fixed number (p) of operations with identical operation times on the three machines (t_l for operation l) which can be performed in any order on the three machines. Furthermore, we assume that each operation can be assigned to any one of the machines.

Three different cell layouts are considered in the literature for robotic cells: robot-centered cells (where the robot movement is rotational), in-line robotic cells (where the robot moves linearly), and mobile-robot cells (generalization of in-line robotic cell and robot-centered cell) [1]. In this study we deal with the in-line robotic cell layout with three

* Corresponding author. Tel.: +90 312 2664126; fax: +90 312 266 4054.
E-mail address: akturk@bilkent.edu.tr (M.S. Akturk).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2005.09.015

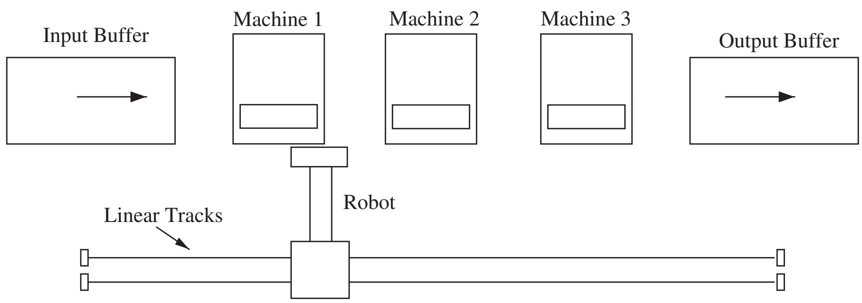

Fig. 1. Inline robotic cell layout.

machines as shown in Fig. 1. In these cells the robot moves on the linear tracks to transport the parts between the machines. After loading a part on one of the machines, the robot either waits in front of the machine to finish the processing, moves to unload another machine or moves to input buffer to take another part and load the first machine. At any time instant, a part is either on one of the machines, on the robot or at the input or output buffer. Crama et al. [2] state the following assumptions which can be referred to as *basic feasibility assumptions*:

(i) Robot never loads an already loaded machine.
(ii) Robot never unloads an already empty machine.

A state of the system can be specified by the position of the robot and whether the machines are loaded or empty. Consistent with the literature, we will deal with cyclic schedules in this study. A cyclic schedule is one in which the robot performs the same set of activities continuously. A cycle starts with an initial state of the system, the robot performs a set of activities and when the system returns back to the initial state, the cycle is completed. A cycle in which n parts are produced is called an n-unit cycle.

Various problems concerning the robotic cells arise in the literature. Crama et al. [2] and Dawande et al. [3] provide extensive surveys in this research area. Sethi et al. [4] developed the necessary framework and proved that for 2-machine identical parts case, the optimal solution is a 1-unit cycle and conjectured that 1-*unit cycles are optimal for m-machine robotic cells*. They also proved that the number of 1-unit cycles in the m-machine case is exactly $(m!)$. The validity of the conjecture of Sethi et al. [4] for 3-machine robotic flow-shops is established by Crama and Van de Klundert [5]. Brauner and Finke [6] simplified this proof. In a later study, Brauner and Finke [7] proved that 1-unit cycles do not necessarily yield optimal solutions for cells of size four or large. Hall et al. [8] showed that for multiple part types, even in 2-machines case there are instances for which 1-unit cycles are dominated. Aneja and Kamoun [9] proposed an $O(n \log n)$ time algorithm for the same problem which determines the optimal robot move cycle and the part input sequence simultaneously. Geismar et al. [10] developed a 1.5 approximation for the multi-unit cyclic solution. Akturk et al. [11] considered a robotic cell with two identical CNC machines which possess operational and process flexibility. For this problem, they proved that 1-unit cycles are not necessarily optimal and that a 2-unit cycle can also be optimal and presented the regions of optimality.

Current study deviates from the existing literature in that we assume the robotic cell to be a flexible manufacturing cell (FMC). An FMC is a production cell consisting of CNC machines, connected through an automated material handling system and controlled by a centralized computer. "Flexibility" is the key term which distinguishes FMCs from traditional ones. There are several types of flexibilities such as machine, material handling, operation, process and routing flexibilities. In this study we consider operation flexibility (the ability to change the ordering of several operations) and process flexibility (the ability of machines to perform multiple operations). Such flexibilities are achieved by considering alternative tool types for operations and loading multiple tools to the tool magazines of the machines.

Furthermore, in this study we will investigate the additional flexibility introduced by the CNC machines. A new robot move cycle which is a direct consequence of operational and process flexibilities will be proposed. We will compare the cycle times (throughput values) of the proposed cycle and the robot move cycles considered in the literature thus far and exhibit the regions of optimality.

In the following section, the notation and basic assumptions pertinent to this study will be introduced and a new cycle will be proposed. In Section 3, the solution methodology will be introduced and a worst case performance bound of using the proposed cycle will be derived. Section 4 is devoted to the concluding remarks and future research directions.

2. Notation and assumptions

In this section we present the basic assumptions and the notation to be used throughout this study as well as propose a new cycle. We shall resort to the following definition borrowed from [12] throughout the text.

Definition 1. A_i is the robot activity defined as; robot unloads machine i, transfers part from machine i to machine $i+1$, loads machine $i+1$. The input buffer is denoted as machine 0 and the output buffer is denoted as machine 4.

Using this definition we can list the six feasible 1-unit cycles of 3-machine robotic cells as follows:

$$S1 = (A_0A_1A_2A_3), \quad S2 = (A_0A_2A_1A_3), \quad S3 = (A_0A_1A_3A_2),$$

$$S4 = (A_0A_3A_1A_2), \quad S5 = (A_0A_2A_3A_1), \quad S6 = (A_0A_3A_2A_1).$$

The above definition of robot activities is necessary and enough for the traditional research in this area where the system is assumed to be a flowshop. In such systems, all of the parts go through the same sequence of machines, namely 1, 2 and 3. However, in the settings of this study, each part may have an alternative route through the machines. This flexibility results in new cycles and in order to represent these we need the following definition:

Definition 2. A_{ij} is the robot activity defined as; robot unloads machine i, transfers part from machine i to machine j and loads machine j, where $j > i$.

With the assumptions of this study we propose a new cycle which can be represented with the sequence of activities; $A_{01}A_{02}A_{03}A_{14}A_{24}A_{34}$. In words, the robot first loads machines 1, 2 and 3 in respective order, all of the operations of the parts are performed by a single machine, then the robot unloads machines 1, 2, and 3 and drops the parts to the output buffer in respective order. Note that three parts are produced by a single repetition of this cycle, hence it is a 3-unit cycle. The derivation of the cycle time for this proposed cycle is shown in Appendix A. (The animated views of the proposed cycle and the 1-unit cycles considered in the literature so far can be found at the web site http://www.ie.bilkent.edu.tr/∼robot.)

Let us first pinpoint the differences of this study from the classical ones. In traditional robotic cell scheduling literature, the processing time of each part on each machine is a known parameter. However, in this study the processing times of the parts on the machines are decision variables. The identical parts have a number of operations to be completed on three machines and the individual operation times are known and identical for all machines. Let $O = \{1, 2, \ldots, p\}$ be the set of all operations. The processing times of a part on each of the three machines depend on the allocation of these operations to the machines. An allocation of operations to the three machines means partitioning set O into three subsets; O_1, O_2 and O_3, where O_i is the set of operations allocated to machine i. Consequently, by finding the optimal allocation of the operations to the machines we can minimize the cycle time. Moreover, the allocation of the operations to the machines need not be the same for all parts. The operations of each part can be allocated differently. Since during one repetition of the cycle more than one part can be processed on different machines simultaneously, having different allocations for these parts is an opportunity to minimize the cycle time. However, since we consider cyclic production, that is, the robot performs the same set of activities repeatedly, after some point the allocation of the operations of a part, say the $(k+1)$st part $k = 1, 2, \ldots,$ becomes identical with the first part. Then, the allocation of the operations of the parts 1 through k is used in the same order repeatedly for the remaining parts. That is, k is the period of the allocation types. Note that, for a specific k, there are a finite (though large) number of different ways to allocate the operations of the parts. For the clarity of the consequent discussion, we need the following definition and notation.

Definition 3. Let $\Pi_k = [\pi_{ij}]$ denote a specific allocation matrix with k different allocation types. The (i, j)th entry, π_{ij}, $i = [1, 2, \ldots, k]$ and $j = [1, 2, 3]$, of this matrix corresponds to the set of operations allocated to the jth machine for every $(rk+i)$th part in the infinite sequence where $r=0,1,2,\ldots$. For this matrix we have:

- Each row corresponds to a proper 3-partitioning of the operation set O. With our notation, for any i, $\pi_{i1}\cup\pi_{i2}\cup\pi_{i3}=O$ and $\pi_{i1}\cap\pi_{i2}=\emptyset$ and $\pi_{i1}\cap\pi_{i3}=\emptyset$ and $\pi_{i2}\cap\pi_{i3}=\emptyset$.
- No two rows are identical.

We also let Π_k^* denote the optimal allocation of operations when a total of k different allocation types is used.

For example, for a cycle for which specific two different allocation types are used, the allocations of the operations are represented as follows:

$$\Pi_2=\begin{bmatrix}\pi_{11} & \pi_{12} & \pi_{13}\\ \pi_{21} & \pi_{22} & \pi_{23}\end{bmatrix}.$$

That is, there are two distinct 3-partitions of operations to the machines which are used alternately. Before we proceed with a numerical example let us list the remaining notation to be used throughout the text.

- t_l: Processing time of operation l. Note that the processing time of operation l on all three machines are equal, $\forall l=1,2,\ldots,p$.
- P: Total processing time of the operations that will be allocated to the machines, $P=\sum_{l=1}^{p}t_l$. Note that the sum of the processing times on three machines corresponding to each row of the allocation matrix is also equivalent to P.
- P_{ij}: Total processing time on machine j for the part which corresponds to the ith row of the specific allocation matrix Π. That is, $P_{ij}=\sum_{l\in\pi_{ij}}t_l$. Also, we let $P_\pi=[P_{ij}]$.
- w_{ij}: Waiting time of the robot for machine j to finish the processing of the part which is produced according to the ith row of the allocation matrix Π. Note that, if the processing of the part is already finished when the robot arrived to machine j, then the waiting time is 0.
- ε: The load and unload time of machines by the robot. Consistent with the literature we assume that loading/unloading times for all machines are the same.
- δ: Time taken by the robot to travel between two consecutive machines.
- $T_{Sj(\Pi_k)}$: Cycle time, i.e., the long run average time that is required to produce one part using robot move cycle Sj and the specific allocation matrix Π_k.

Example 1. Let us assume that each part has 5 operations to be performed on the three machines with corresponding operation times $t_1=30$, $t_2=25$, $t_3=35$, $t_4=30$ and $t_5=15$. Thus, total processing time of each part is $P=135$. Let us also assume that $\varepsilon=2$ and $\delta=4$. Now consider the robot move cycle $S6$ which is defined by the following activity sequence $A_0A_3A_2A_1$. In our study, the cycle time derived by Sethi et al. [4] corresponds to the case where the allocations of the operations of all parts are identical. Let Π_1 be a specific allocation. Then, the cycle time for this case is the following:

$$T_{S6(\Pi_1)}=8\varepsilon+12\delta+\max\{0,\,P_{11}-4\varepsilon-8\delta,\,P_{12}-4\varepsilon-8\delta,\,P_{13}-4\varepsilon-8\delta\}.$$

The optimal allocation in this case is: $\Pi_{11}^*=\{1,5\}$ with $P_{11}^*=45$, $\Pi_{12}^*=\{2,4\}$ with $P_{12}^*=55$, and $\Pi_{13}^*=\{3\}$ with $P_{13}^*=35$. The corresponding cycle time is

$$T_{S6(\Pi_1^*)}=64+\max\{0,45-40,55-40,35-40\}=79.$$

Now let us assume that two different allocation types are used repeatedly. That is, a specific allocation is now represented by Π_2. The new cycle time to produce one part for this case is the following:

$$\begin{aligned}T_{S6}(\Pi_2)=8\varepsilon+12\delta&+\tfrac{1}{2}\max\{0,\,P_{11}-4\varepsilon-8\delta,\,P_{12}-4\varepsilon-8\delta,\,P_{13}-4\varepsilon-8\delta\}\\&+\tfrac{1}{2}\max\{0,\,P_{21}-4\varepsilon-8\delta,\,P_{22}-4\varepsilon-8\delta,\,P_{23}-4\varepsilon-8\delta\}.\end{aligned}$$

The optimal allocations of the operations for this case are, in the first allocation type, $\Pi_{11}^*=\{1,2\}$, $\Pi_{12}^*=\{3\}$, $\Pi_{13}^*=\{4,5\}$. In other words, $P_{11}^*=55$, $P_{12}^*=35$ and $P_{13}^*=45$. As for the second allocation type, $\Pi_{21}^*=\{4,5\}$ with

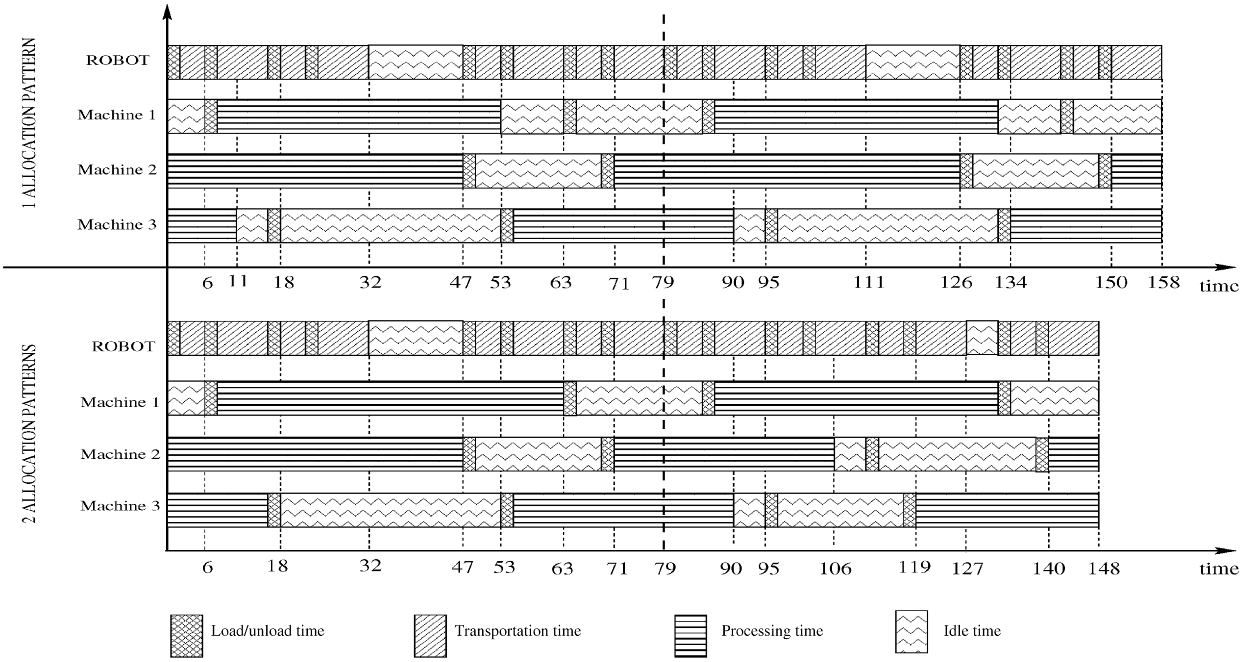

Fig. 2. Gantt chart for Example 1.

$P^*_{21} = 45$, $\Pi^*_{22} = \{1, 2\}$ with $P^*_{22} = 55$, and finally $\Pi^*_{23} = \{3\}$ with $P^*_{23} = 35$. Then the corresponding cycle time is the following:

$$T_{S6}(\Pi^*_2) = 64 + \tfrac{1}{2}\max\{0, 55-40, 55-40, 45-40\} + \tfrac{1}{2}\max\{0, 45-40, 35-40, 35-40\} = 74.$$

The Gantt chart in Fig. 2 compares these two cases. In order to see the difference, the Gantt chart of one allocation case is drawn for two repetitions of the cycle. One can observe that the completion times of the first repetition of both cycles (bold dashed line in the figure) are the same but the completion times of the second repetition of the robot activities are different. In one allocation case the second repetition is exactly the same as the first repetition (which means the processing times on the machines are the same). However, for two different allocations case, the time of the second repetition is less than the first repetition because the total waiting time of the robot in front of the machines is reduced by 10 units. Then, in order to produce 1 part, this makes 5 units or $5/79 = 6.3\%$ decrease in between the cycle times of these two cases.

In fact, in this example using three different types of allocations is optimal. The optimal allocation matrix and the corresponding processing times are

$$\Pi^*_3 = \begin{bmatrix} \{1,2\} & \{4,5\} & \{3\} \\ \{3\} & \{1,2\} & \{4,5\} \\ \{4,5\} & \{3\} & \{1,2\} \end{bmatrix} \Rightarrow P^*_{\Pi} = \begin{bmatrix} 55 & 45 & 35 \\ 35 & 55 & 45 \\ 45 & 35 & 55 \end{bmatrix}.$$

The cycle time for this case is, $T_{S6(\Pi^*_3)} = 70.67$. The cycle time of the proposed cycle $A_{01}A_{02}A_{03}A_{14}A_{24}A_{34}$, with given parameters is $T = 69$, which is optimal for this example. This corresponds to $\frac{10}{79} = 12.7\%$ decrease from the best cycle time that can be found using the results reported in the literature. It is important to note that this significant decrease in cycle time can be obtained with no additional cost, just by capturing the inherent flexibility of the CNC machines.

At this point one can conjecture that if the whole processing of a part can be done on a single machine, there is no reason for performing a portion of it on a machine and the rest on another. Thus, in this way some load/unload time will be saved and the proposed cycle will always be optimal and there is no reason to consider the robot move cycles derived under the assumption of a flowshop type system. As we will see later in this study, this conjecture holds when the robot is the bottleneck. That is, when the total processing time of the parts is small with respect to the load/unload time, ε, and transportation time, δ. However, when the machines are bottleneck instead of the robot, that is, total processing time might become greater with respect to ε and δ, the proposed cycle may result in higher cycle time values. In other words, if the processing time exceeds some threshold then the average idle time of the machines waiting for some part to be loaded is greater in the proposed cycle. The following is a concrete example to this nature.

Example 2. Consider 6 operations with $t_1 = 40$, $t_2 = 45$, $t_3 = 50$, $t_4 = 60$, $t_5 = 50$, $t_6 = 55$ so the total processing time of each part is $P = 300$. Also let $\varepsilon = 2$ and $\delta = 10$. In this case for cycle $S6$ let Π_1 be a specific allocation matrix in which $\Pi_{11} = \{1, 4\}$, $\Pi_{12} = \{2, 6\}$ and $\Pi_{13} = \{3, 5\}$ resulting in $P_{11} = 100$, $P_{12} = 100$ and $P_{13} = 100$. The cycle time in this case is $T_{S6(\Pi_1)} = 148$. On the other hand the cycle time of the proposed cycle is 152.

In the next section we will present the solution procedure for this problem and list the regions of optimality for the proposed cycle.

3. Solution procedure

In this section we will compare the proposed cycle with the flowshop type robot move cycles presented in the literature and find the regions of optimality for each of these cycles. Recall that the flowshop type robot move cycles are those that are considered thus far in the research on this area except in Akturk et al. [11]. In these cycles each part to be processed follows the same sequence of machines. Geismar et al. [10] provide a lower bound for the cycle time of such classical robot move cycles when there are m machines. With Theorem 1, we extend this result to our case involving the operation allocation problem as well. With Theorem 2, we are able to find the regions where the proposed cycle dominates the flowshop type robot move cycles. Next, for the regions where the flowshop type robot move cycles are not dominated; in Lemma 1 we will prove that the proposed cycle dominates all 1-unit cycles except $S6$ and in Lemma 3 we will prove that the proposed cycle dominates all 2-unit cycles. Theorem 3 derives a worst case performance bound for using the proposed cycle.

In the following theorem we will derive a lower bound for the cycle time of the flowshop type robot move cycles. Let $\underline{T_{\text{flowshop}}}$ represent the lower bound for these cycles. Note that, $\underline{T_{\text{flowshop}}} \leqslant \min_{Sj,k}\{T_{Sj(\Pi_k^*)}\}$. That is, the lower bound is over all flowshop type robot move cycles, Sj's, and for the optimal allocations, Π_k^*, over all possible allocation periods, k.

The following result will prove useful when comparing the proposed cycle's times with those of the flowshop type robot move cycles. The fact that the result is allocation independent is further going to ease our analysis.

Theorem 1. *For a 3 machine robotic cell, the cycle time of any n-unit flowshop type robot move cycle with any allocation matrix Π_k is no less than*

$$\underline{T_{\text{flowshop}}} = \max\{8(\varepsilon + \delta) + \min\{P, \delta\}, 4\varepsilon + 4\delta + (P/3)\}. \tag{1}$$

Proof. Let us consider any specific allocation matrix Π_k. The reasoning behind the first term in the max function is as follows: the robot loads and unloads all 3 machines exactly once, (6ε), also takes a part from the input buffer, (ε), and drops a part to the output buffer, (ε), in every cycle. Then the total load and unload time is exactly 8ε. As the forward movement, the robot travels all the way from the input buffer to the output buffer in some sequence of robot activities, which takes at least 4δ and in order to return back to the initial state the robot must travel back to the input buffer which again takes at least 4δ. Thus the travel time is at least 8δ. On the other hand, after loading a part to machine j, the robot has two options: it either waits in front of the machine, (P_{ij}) or travels to another machine to make some other activities which at least takes δ time units. Then, for all allocation types and for all machines, we have $\sum_{i=1}^{k}\sum_{j=1}^{3} \min\{P_{ij}, \delta\}/k$. We divide by k since we defined the cycle time as the time required to produce one part.

Furthermore, since $\min\{P,\delta\} \leqslant \sum_{j=1}^{3} \min\{P_{ij},\delta\}$, we have, $(1/k)\sum_{i=1}^{k} \min\{P,\delta\} = \min\{P,\delta\}$ and the first term of the max function becomes,

$$8(\varepsilon+\delta)+\min\{P,\delta\}.$$

The reasoning behind the second term of the max function is the following: the cycle time of any cycle is greater than the time between two consecutive loadings of a machine for which the consecutive loading time is the greatest. But in order to make a consecutive loading, the robot must at least perform the following activities: after loading a part to some machine j, the robot either waits in front of the machine or makes some other activities which takes at least P_{ij} amount of time; then, the robot unloads machine j, (ε); transports the part to machine $(j+1)$, (δ); loads it, (ε); returns back to machine $(j-1)$, (2δ); unloads it, (ε); transports the part to machine j, (δ); and loads it, (ε). This in total makes $4\varepsilon+4\delta+P_{ij}$. In order to find the greatest consecutive loading time we take $\max_{ij}\{P_{ij}\}$. However, since $\sum_{j=1}^{3} P_{ij} = P\ \forall i$, we have, $P/3 \leqslant \max_{i,j}\{P_{ij}\}$. Then the second term of the max function becomes $4\varepsilon+4\delta+P/3$. This completes the proof. □

The forthcoming theorem compares the cycle time of the proposed cycle with the lower bound of the traditional robot move cycles and provides the regions where the proposed cycle is optimal.

Theorem 2. *If $\delta \leqslant 2\varepsilon$ or $P \leqslant 16\varepsilon+13\delta$, then the proposed cycle gives the minimum cycle time.*

Proof. In order to prove this theorem we will compare the cycle time of the proposed cycle given in (A.1) and the lower bound of the cycle times of the flowshop type move cycles given in (1). When we consider $\underline{T_{\text{flowshop}}}$, there are two breakpoints with respect to P where the cycle time has a different form. These are: $(P=\delta)$ and $(P=12\varepsilon+15\delta)$. The first one of these comes from the two arguments of the min term inside the max and the second breakpoint comes from the comparison of the two arguments of the max term. In the same way T_{proposed} also has a breakpoint with respect to P, which is: $(P=4\varepsilon+10\delta)$. Comparing this with the ones we get from $\underline{T_{\text{flowshop}}}$, one can easily show that $\delta < 4\varepsilon+10\delta < 12\varepsilon+15\delta$. In the following cases we will consider each of these regions:

1. If $P \leqslant \delta$, then the cycle time of the proposed cycle is

   $$T_{\text{proposed}} = 4\varepsilon+8\delta.$$

   The lower bound of the cycle times of flowshop type robot move cycles is

   $$\underline{T_{\text{flowshop}}} = 8\varepsilon+8\delta+P \quad\Rightarrow\quad T_{\text{proposed}} < \underline{T_{\text{flowshop}}}.$$

2. If $\delta < P \leqslant 4\varepsilon+10\delta$, then $T_{\text{proposed}} = 4\varepsilon+8\delta < 8\varepsilon+9\delta = \underline{T_{\text{flowshop}}}$.
3. If $4\varepsilon+10\delta < P \leqslant 12\varepsilon+15\delta$, then $T_{\text{proposed}} = \frac{1}{3}(8\varepsilon+14\delta+P)$ and $\underline{T_{\text{flowshop}}} = 8\varepsilon+9\delta$. Comparing these two, one can show that the proposed cycle dominates the flowshop type robot move cycles if $P \leqslant 16\varepsilon+13\delta$ and no dominance relations exist otherwise. On the other hand, if $\delta \leqslant 2\varepsilon$ then $16\varepsilon+13\delta \geqslant 12\varepsilon+15\delta$. This means that for $\delta \leqslant 2\varepsilon$ the breakpoint is outside the region of consideration which means the proposed cycle dominates all of the flowshop type robot move cycles.
4. If $P > 12\varepsilon+15\delta$, then $T_{\text{proposed}} = \frac{1}{3}(8\varepsilon+14\delta+P)$ and $\underline{T_{\text{flowshop}}} = 4\varepsilon+4\delta+P/3$. Comparing these two, one can show that for $\delta \leqslant 2\varepsilon$ the proposed cycle dominates the flowshop type robot move cycles but for $\delta > 2\varepsilon$ no dominance relations exist. □

Now let us consider the region where the lower bound of the flowshop type robot move cycles is less than the cycle time of the proposed robot move cycle. That is, $\delta > 2\varepsilon$ and $P > 16\varepsilon+13\delta$. First we concentrate on the 1-unit robot move cycles since they are simple, practical, easy to understand and provably optimal for 3-machine flowshop type systems. The following lemma is very useful in reducing the number of potentially optimal robot move cycles.

Lemma 1. *The proposed cycle dominates all flowshop type* 1-*unit cycles except* $S6$.

Proof. Let us consider each 1-unit cycle one by one:

$S1$: For the cycle $S1$, whatever the allocation of the operations is, the cycle time is the same. The cycle time derived by Sethi et al. [4] is

$$T_{S1(\Pi_k)} = 8\varepsilon + 8\delta + P.$$

As it is seen, the cycle time does not depend on the allocation. When we compare this cycle time with the cycle time of the proposed cycle given in (A.1), $T_{\text{proposed}} < T_{S1(\Pi_k)}$. Thus we conclude that the proposed cycle dominates $S1$.

$S2$: Let us derive the cycle time of the cycle $S2$ considering the assumptions of this study. Consider an arbitrary allocation matrix Π_k and the ith repetition of this cycle. Initially the second machine is loaded with a part having allocation type $(i-1)$ and the robot is in front of the input buffer. The robot takes a part from the input buffer and loads it to the first machine, $(2\varepsilon+\delta)$, moves to second machine, waits if necessary for the machine to finish the processing of the part with allocation type $(i-1)$, $(\delta + w_{(i-1)2})$, unloads the second machine and loads the third machine, $(2\varepsilon+\delta)$, moves to the first machine, waits if necessary for the machine to finish the processing of the part with allocation type i, $(2\delta + w_{i1})$, unloads the first machine and loads the second machine, $(2\varepsilon+\delta)$, moves to the third machine and waits if necessary for the part with allocation type $(i-1)$, $(\delta + w_{(i-1)3})$, unloads the machine and drops the part to the output buffer, $(2\varepsilon+\delta)$, returns back to input buffer, (4δ). Hence the time for the ith repetition of the cycle $S2$ with allocation matrix Π_k becomes

$$8\varepsilon + 12\delta + w_{i1} + w_{(i-1)2} + w_{(i-1)3}.$$

Let us denote $\max\{0, a\}$ as $(a)^+$. With this notation, $w_{i1} = (P_{i1} - 2\varepsilon - 4\delta - w_{(i-1)2})^+$, $w_{(i-1)2} = (P_{(i-1)2} - 4\varepsilon - 8\delta - w_{(i-1)3})^+$ and $w_{(i-1)3} = (P_{(i-1)3} - 2\varepsilon - 4\delta - w_{i1})^+$ are the waiting times in front of the machines 1, 2 and 3, respectively. For all k repetitions, we have the following:

$$T_{S2(\Pi_k)} = 8\varepsilon + 12\delta + 1/k\left(\sum_{i=1}^{k}(w_{i1} + w_{(i-1)2} + w_{(i-1)3})\right).$$

Using the fact that $a \leqslant (a)^+$ we get

$$\sum_{i=1}^{k}(w_{i1} + w_{(i-1)2} + w_{(i-1)3}) \geqslant \sum_{i=1}^{k}(P_{i1} + P_{(i-1)2} + P_{(i-1)3} - 8\varepsilon - 16\delta) - \sum_{i=1}^{k}(w_{i1} + w_{(i-1)2} + w_{(i-1)3})$$

Since the allocation of every $(rk+i)$th part in the infinite sequence is identical, then

$$\sum_{i=1}^{k}(P_{i1} + P_{(i-1)2} + P_{(i-1)3} - 8\varepsilon - 16\delta) = k(P - 8\varepsilon - 16\delta)$$

and

$$\sum_{i=1}^{k}(w_{i1} + w_{(i-1)2} + w_{(i-1)3}) = \sum_{i=1}^{k}(w_{i1} + w_{(i-1)2} + w_{(i-1)3}) = W,$$

where W is defined to be the total waiting time in front of the three machines for the k parts produced according to the allocation matrix Π_k. This yields

$$2W \geqslant k(P - 8\varepsilon - 16\delta) \quad\Rightarrow\quad W \geqslant k/2(P - 8\varepsilon - 16\delta).$$

Thus for $S2$ we have

$$T_{S2(\Pi_k)} \geqslant 8\varepsilon + 12\delta + 1/k(k/2(P - 8\varepsilon - 16\delta)) = \tfrac{1}{2}(8\varepsilon + 8\delta + P). \tag{2}$$

$S3$: Using the above procedure, the time for the ith repetition of $S3$ is

$$8\varepsilon + 10\delta + P_{i1} + w_{i2} + w_{(i-1)3},$$

where $w_{i2} = (P_{i2} - 2\varepsilon - 4\delta - w_{(i-1)3})^+$ and $w_{(i-1)3} = (P_{(i-1)3} - 4\varepsilon - 6\delta - P_{i1})^+$ are the waiting times in front of machines 2 and 3, respectively. Then we have the following:

$$P_{i1} + w_{i2} + w_{(i-1)3} \geqslant P_{i1} + P_{i2} - 2\varepsilon - 4\delta - w_{(i-1)3} + P_{(i-1)3} - 4\varepsilon - 6\delta - P_{i1}.$$

This yields

$$2(P_{i1} + w_{i2} + w_{(i-1)3}) \geqslant 2(P_{i1} + w_{(i-1)3}) + w_{i2} \geqslant P_{i1} + P_{i2} + P_{(i-1)3} - 6\varepsilon - 10\delta.$$

For all k repetitions we have the following:

$$2\sum_{l=1}^{k}(P_{i1} + w_{i2} + w_{(i-1)3}) \geqslant \sum_{l=1}^{k}(P_{i1} + P_{i2} + P_{(i-1)3} - 6\varepsilon - 10\delta).$$

Let W be as defined previously. Then we can write the following:

$$2W \geqslant P - 6\varepsilon - 10\delta \Rightarrow W \geqslant \tfrac{1}{2}(P - 6\varepsilon - 10\delta).$$

Then for $S3$ we have the following:

$$T_{S3(\Pi_k)} \geqslant 8\varepsilon + 10\delta + \tfrac{1}{2}(P - 6\varepsilon - 10\delta) = \tfrac{1}{2}(P + 10\varepsilon + 10\delta). \tag{3}$$

$S4$: Total time for the ith repetition of $S4$ is the following:

$$8\varepsilon + 12\delta + w_{i1} + P_{i2} + w_{(i-1)3},$$

where $w_{i1} = (P_{i1} - 2\varepsilon - 6\delta - w_{(i-1)3})^+$ and $w_{(i-1)3} = (P_{(i-1)3} - 2\varepsilon - 6\delta)^+$ are the waiting times in front of machines 1 and 3, respectively. A similar procedure that we used for $S3$ yields, $W \geqslant \frac{1}{2}(P - 4\varepsilon - 12\delta)$ and

$$T_{S4(\Pi_k)} \geqslant \tfrac{1}{2}(12\varepsilon + 12\delta + P). \tag{4}$$

$S5$: Total time for the ith repetition of $S5$ is the following:

$$8\varepsilon + 10\delta + w_{i1} + w_{(i-1)2} + P_{(i-1)3},$$

where $w_{i1} = (P_{i1} - 4\varepsilon - 6\delta - w_{(i-1)2} - P_{(i-1)3})^+$ and $w_{(i-1)2} = (P_{(i-1)2} - 2\varepsilon - 4\delta)^+$ are the waiting times in front of the machines 1 and 2, respectively. From here we get, $W \geqslant \frac{1}{2}(P - 6\varepsilon - 10\delta)$ and the lower bound for $S5$ becomes

$$T_{S5(\Pi_k)} \geqslant \tfrac{1}{2}(10\varepsilon + 10\delta + P). \tag{5}$$

Comparing the lower bounds for the cycles $S2$, $S3$, $S4$ and $S5$, given in Eqs. (2), (3), (4) and (5), respectively, we get the following:

$$\underline{T_{S2(\Pi_k)}} < \underline{T_{S3(\Pi_k)}} = \underline{T_{S5(\Pi_k)}} < \underline{T_{S4(\Pi_k)}},$$

where $\underline{T_{Sj(\Pi_k)}} = \min_{\Pi_k}\{T_{Sj(\Pi_k)}\}$. Let us compare $\underline{T_{S2(\Pi_k)}} = \frac{1}{2}(P + 8\varepsilon + 8\delta)$ with the cycle time of the proposed cycle given in (A.1):

$$\tfrac{1}{2}(8\varepsilon + 8\delta + P) = \tfrac{1}{2}(\tfrac{1}{3}(24\varepsilon + 24\delta + 3P)).$$

Since in this region $P > 16\varepsilon + 13\delta$,

$$\frac{1}{6}(24\varepsilon + 24\delta + 2P + P) > \frac{1}{6}(40\varepsilon + 37\delta + 2P) = \frac{1}{3}(20\varepsilon + (18.5)\delta + P) > \frac{1}{3}(8\varepsilon + 14\delta + P) = T_{\text{proposed}}.$$

Thus we can conclude that the proposed cycle dominates $S2$, $S3$, $S4$ and $S5$.

$S6$: Example 2 shows that $S6$ cannot be dominated by the proposed robot move cycle. □

1-unit cycles are important because they are simple, practical and easy to understand. Also if the system is assumed to be a flowshop then 1-unit cycles are provably optimal for 2-machine cells [4] and 3-machine cells [5]. However, Akturk et al. [11] proved that with the assumption of operational flexibility, even in 2-machines case, a 2-unit cycle can result in smaller cycle times than the 1-unit robot move cycles for some parameter ranges. This motivates us to consider the 2-unit cycles. Hall et al. [8] derived the activity sequences of all feasible 2-unit cycles in a 3-machine robotic cell. In Appendix B we present a completely new procedure to derive the activity sequences of these cycles and list them. This new procedure utilizes the fact that all 2-unit cycles are made up from two 1-unit cycles. That is, let Si and Sj be two different 1-unit cycles. Then, in a 2-unit cycle, S_{ij} is simply a combination of Si and Sj; during some part of the cycle the robot follows the activity sequence of Si and during the remaining part of the cycle the robot follows the activity sequence of Sj.

The following lemma derives a general lower bound, $\underline{T_2}$, for all of the 2-unit robot move cycles with any allocation matrix Π_k.

Lemma 2. $\underline{T_2} = \frac{1}{2}(P + 8\varepsilon + 8\delta)$ *where* $\underline{T_2} \leqslant \min_{S_{ij},k}\{T_{S_{ij(\Pi_k^*)}}\}$.

Proof. For the clarity of the presentation, we refer the reader to Appendix C for the proof. □

The following lemma proves that the proposed cycle dominates all flowshop type 2-unit robot move cycles.

Lemma 3. *The proposed cycle dominates all flowshop type* 2-*unit cycles.*

Proof. With Theorem 2 we assert that the proposed cycle gives the minimum cycle time for $P \leqslant 16\varepsilon + 13\delta$. Now let us consider the region where $P > 16\varepsilon + 13\delta$. In this region the cycle time of the proposed cycle given in (A.1) becomes $4\varepsilon + 8\delta + \frac{1}{3}(P - 4\varepsilon - 10\delta)$ which can be rewritten as $\frac{1}{3}(P + 8\varepsilon + 14\delta)$. When we compare this cycle time with the lower bound we found in Lemma 2, we have the following:

$$\underline{T_2} = \frac{1}{2}(P + 8\varepsilon + 8\delta) = \frac{1}{6}(3P + 24\varepsilon + 24\delta) \geqslant \frac{1}{6}(2P + 40\varepsilon + 37\delta) = \frac{1}{3}(P + 20\varepsilon + (18.5)\delta) > \frac{1}{3}(P + 8\varepsilon + 14\delta) = T_{\text{proposed}}.$$

This completes the proof. □

Until now we considered all the 1 and 2-unit cycles and showed that the proposed cycle dominates all except the 1-unit cycle $S6$. Knowing that the proposed cycle dominates all 2-unit cycles, one can conjecture that it also dominates 3 and higher unit cycles. Proving or disproving this conjecture is not so simple because the number of feasible robot move cycles increases drastically as n, the number of units produced in one cycle increases and deriving and comparing these cycles with the proposed cycle become quite complex. Additionally, the proposed cycle is simple, practical and easy to implement. Furthermore, there is no allocation problem to be solved for this cycle. More importantly, the following theorem derives a worst case bound for using the proposed cycle. As a result of these observations, we conclude that what little improvement we might attain (if any) by considering 3 and higher unit cycles will not be sufficient enough to justify the effort that will be spent for this purpose.

Theorem 3. *The worst case performance bound of using the proposed cycle is* 1.08.

Proof. Let us consider the region $\delta > 2\varepsilon$ and $P > 16\varepsilon + 13\delta$ since otherwise the proposed cycle is optimal as stated in Theorem 2. Let T^* represent the cycle time of the optimal robot move cycle with optimal allocation matrix. We know from (1) that $T^* \geqslant \frac{1}{3}(12\varepsilon + 12\delta + P)$. Thus we have

$$\frac{T_{\text{proposed}}}{T^*} \leqslant \frac{\frac{1}{3}(8\varepsilon + 14\delta + P)}{\frac{1}{3}(12\varepsilon + 12\delta + P)} = 1 + \frac{2\delta - 4\varepsilon}{12\varepsilon + 12\delta + P}.$$

Since $P > 16\varepsilon + 13\delta$,

$$\frac{T_{\text{proposed}}}{T^*} < 1 + \frac{2\delta - 4\varepsilon}{25\delta + 28\varepsilon}.$$

Let $\delta = \alpha\varepsilon$ where $\alpha \geqslant 2$;

$$\frac{T_{\text{proposed}}}{T^*} < 1 + \frac{2\alpha - 4}{25\alpha + 28}.$$

In order to find the worst case performance take the limit as ($\alpha \to \infty$);

$$\lim_{\alpha \to \infty} \left(1 + \frac{2\alpha - 4}{25\alpha + 28}\right) = 1.08. \qquad \square$$

α being infinity means that the load/unload time is negligibly small. One can also assume it to be infinity if the robot travel time is extremely high. However, the latter case does not have much practical relevance.

4. Conclusion

In this study we considered a 3-machine FMC in which the loading and unloading of the machines are made by a robot. The machines in an FMC are CNC machines and they possess different kinds of flexibilities. In this paper we focused on the process and operational flexibilities. In existence of such flexibilities, we assumed identical parts, each having a number of operations to be performed on these three machines. We also assumed all machines to be able to perform all the required operations in order to finish the processing of a part. We investigated the productivity gain attained by the additional flexibility introduced by the FMCs.

We proposed a new robot move cycle which is a direct consequence of the assumption of process and operational flexibilities. In Theorem 1, we derived a lower bound for the flowshop type robot move cycles considered in the literature and compared this lower bound with the cycle time of the proposed cycle. In Theorem 2, we found the regions of optimality for the proposed cycle. We then considered the 1-unit and 2-unit cycles individually and showed that the proposed cycle dominates all of the 1-unit robot move cycles except $S6$ and dominates all of the 2-unit robot move cycles.

Options for future research includes extending the results of this study to the m-machine general case, assuming non-identical machines so that the processing time of the tasks on different machines are different, assuming multiple parts instead of identical parts so that each of them has a number of operations to be allocated to the three machines, and assuming the machines to have limited tool magazine capacity so that they can hold a limited number of tools thus preventing them to perform all the operations that a part requires.

Appendix A. Cycle time calculation of the proposed cycle

We will derive the cycle time of the proposed cycle (the animated view can be found at http://www.ie.bilkent.edu.tr/~robot). Initially the robot is in front of the input buffer and all of the machines are empty. The robot takes a part from the input buffer, transports it to the first machine and loads it to the first machine, $(\varepsilon + \delta + \varepsilon)$; returns back to the input buffer, takes another part, transports it to the second machine, loads the second machine, $(\delta + \varepsilon + 2\delta + \varepsilon)$; returns back to the input buffer, takes the third part, transports to the third machine and loads this machine, $(2\delta + \varepsilon + 3\delta + \varepsilon)$; returns back to the first machine in order to unload the machine, (2δ); if the processing of the part is already finished, unloads the machine without waiting, otherwise waits in front of the machine in order to finish the processing, $(w1)$; unloads the machine, transports the part to the output buffer, drops the part, returns back to the second machine in order to unload the machine, $(\varepsilon + 3\delta + \varepsilon + 2\delta)$; waits if necessary, $(w2)$; unloads the machine, transports the part to the output buffer, drops the part, returns back to the third machine, $(\varepsilon + 2\delta + \varepsilon + \delta)$; waits if necessary, $(w3)$; unloads the machine, transports the part to the output buffer, drops the part, returns back to the input buffer so that the initial and the final states of the cycle will be identical, $(\varepsilon + \delta + \varepsilon + 4\delta)$. Then the total time to produce three parts is

$$12\varepsilon + 24\delta + w1 + w2 + w3,$$

where $w1 = \max\{0, P - 4\varepsilon - 10\delta\}$, $w2 = \max\{0, P - 4\varepsilon - 12\delta - w1\}$ and $w3 = \max\{0, P - 4\varepsilon - 10\delta - w1 - w2\}$.

From here,

$$\begin{aligned} w1 + w2 &= \max\{w1, P - 4\varepsilon - 12\delta\} = \max\{0, P - 4\varepsilon - 10\delta, P - 4\varepsilon - 12\delta\} \\ &= \max\{0, P - 4\varepsilon - 10\delta\} \end{aligned}$$

and

$$\begin{aligned} w1 + w2 + w3 &= w1 + w2 + \max\{0, P - 4\varepsilon - 10\delta - w1 - w2\} = \max\{w1 + w2, P - 4\varepsilon - 10\delta\} \\ &= \max\{0, P - 4\varepsilon - 10\delta, P - 4\varepsilon - 10\delta\} = \max\{0, P - 4\varepsilon - 10\delta\}. \end{aligned}$$

Then, the long run average cycle time to produce one part is

$$T_{\text{proposed}} = 4\varepsilon + 8\delta + \tfrac{1}{3}\max\{0, P - 4\varepsilon - 10\delta\}. \tag{A.1}$$

Appendix B. Derivation of the 2-unit robot move cycles

Let us present the procedure for deriving the activity sequences for the 2-unit robot move cycles. In a 2-unit cycle, each activity is made exactly twice. A 2-unit robot move cycle is in fact a combination of two 1-unit cycles. At some part of the 2-unit cycle it follows the activity sequence of one of the 1-unit cycles and in the remaining part it follows the activity sequence of the other 1-unit cycle. Then, in order to follow the activity sequences of two 1-unit robot move cycles, there must be a transition state from one of them to the other and later another transition from the latter one to the initial. This requires the two 1-unit robot move cycles to have at least one common state. In the context of this study, any state of the system can be defined as a triplet $(x_1x_2x_3)$, $x_i \in \{0, 1\}$ where $x_i = 0$ indicates that machine i is empty and $x_i = 1$ indicates that machine i is loaded. For example, (011) is a state in which machine 1 is empty and machines 2 and 3 are loaded. The following lists the activity sequences and the states of all 1-unit robot move cycles:

$S1 : A_0A_1A_2A_3 : (000) \rightarrow (100) \rightarrow (010) \rightarrow (001)$
$S2 : A_0A_2A_1A_3 : (010) \rightarrow (110) \rightarrow (101) \rightarrow (011)$
$S3 : A_0A_1A_3A_2 : (001) \rightarrow (101) \rightarrow (011) \rightarrow (010)$
$S4 : A_0A_3A_1A_2 : (001) \rightarrow (101) \rightarrow (100) \rightarrow (010)$
$S5 : A_0A_2A_3A_1 : (010) \rightarrow (110) \rightarrow (101) \rightarrow (100)$
$S6 : A_0A_3A_2A_1 : (011) \rightarrow (111) \rightarrow (110) \rightarrow (101)$

In the above list all cycles have a common state with each other except cycles $S1$ and $S6$. This means that we will have $C(6, 2) - 1 = 14$, 2-unit robot move cycles. Let us give an example of constructing a 2-unit robot move cycle from two 1-unit robot move cycles. Let S_{ij} be defined as the 2-unit cycle made up from 1-unit cycles Si and Sj. Let us consider $S1$ and $S2$ for this example. The common state for these two cycles is (010). Thus, without loss of generality, we may start with the activities of $S1$ and follow them until we reach state (010); $A_0A_1 \ldots$. At the common state we start following the activity sequence of $S2$ until we reach to that common state again; $A_0A_1A_0A_2A_1A_3 \ldots$. Finally, we end up with the remaining activities of $S1$; $A_0A_1A_0A_2A_1A_3A_2A_3$.

The robot activity sequences for each of the 14 2-unit robot move cycles over 3-machines are listed below.

$S_{12} = A_0A_1A_0A_2A_1A_3A_2A_3$	$S_{26} = A_0A_2A_1A_0A_3A_2A_1A_3$
$S_{13} = A_0A_1A_2A_0A_1A_3A_2A_3$	$S_{34} = A_0A_1A_3A_2A_0A_3A_1A_2$
$S_{14} = A_0A_1A_2A_0A_3A_1A_2A_3$	$S_{35} = A_0A_1A_3A_0A_2A_3A_1A_2$
$S_{15} = A_0A_1A_0A_2A_3A_1A_2A_3$	$S_{36} = A_0A_1A_0A_3A_2A_1A_3A_2$
$S_{23} = A_0A_1A_3A_0A_2A_1A_3A_2$	$S_{45} = A_0A_2A_3A_1A_2A_0A_3A_1$
$S_{24} = A_0A_2A_1A_3A_2A_0A_3A_1$	$S_{46} = A_0A_1A_0A_3A_2A_3A_1A_2$
$S_{25} = A_0A_2A_1A_3A_0A_2A_3A_1$	$S_{56} = A_0A_2A_1A_0A_3A_2A_3A_1$

Appendix C. Derivation of lower bounds for the 2-unit robot move cycles

Let $\underline{T_{S_{ij(\Pi_k^*)}}}$ be the lower bound of the cycle time of the 2-unit cycle S_{ij} with any allocation matrix Π_k. Note that $\underline{T_{S_{ij(\Pi_k^*)}}} \leqslant \min_{\Pi_k}\{T_{S_{ij(\Pi_k)}}\}$. We will show that $\underline{T_2} \leqslant \underline{T_{S_{ij(\Pi_k^*)}}}$. We will consider each 2-unit cycle, S_{ij}, one at a time and derive a lower bound, $\underline{T_{S_{ij(\Pi_k^*)}}}$, for each of them. For each of these cycles, we will consider one repetition of the cycle where we assume w.l.o.g. that the cycle starts with the state that the robot is in front of the input buffer just taking a part with ith allocation type. We will find a lower bound for the total time of this particular repetition and show that this lower bound does not depend on i, which means that the lower bound for the total time of this repetition of the cycle is also a lower bound for the total time of all repetitions of the cycle. Now let us consider each 2-unit cycle one at a time.

S_{12}: Using a similar procedure used in Appendix A, one can calculate the total time for one repetition of this cycle starting with loading a part with ith allocation type as $16\varepsilon + 20\delta + P_{i1} + w_{i2} + w_{(i+1)1} + w_{i3} + w_{(i+1)2} + P_{(i+1)3}$, where $w_{i2} = (P_{i2} - 2\varepsilon - 4\delta)^+$, $w_{(i+1)1} = (P_{(i+1)1} - 2\varepsilon - 4\delta - w_{i2})^+$, $w_{i3} = (P_{i3} - 2\varepsilon - 4\delta - w_{(i+1)1})^+$ and $w_{(i+1)2} = (P_{(i+1)2} - 2\varepsilon - 4\delta - w_{i3})^+$.

Since we have $P_{i2} - 2\varepsilon - 4\delta \leqslant w_{i2}$, $P_{i3} - 2\varepsilon - 4\delta - w_{(i+1)1} \leqslant w_{i3}$, $0 \leqslant w_{(i+1)2}$ and $0 \leqslant P_{(i+1)3}$, we have $16\varepsilon + 20\delta + P_{i1} + P_{i2} - 2\varepsilon - 4\delta + w_{(i+1)1} + P_{i3} - 2\varepsilon - 4\delta - w_{(i+1)1} \leqslant 16\varepsilon + 20\delta + P_{i1} + w_{i2} + w_{(i+1)1} + w_{i3} + w_{(i+1)2} + P_{(i+1)3}$. Since in a 2-unit cycle two parts are produced in one repetition, a lower bound to produce one part can be found as

$$\underline{T_{S_{12(\Pi_k^*)}}} = \tfrac{1}{2}(P + 12\varepsilon + 12\delta) > \underline{T_2}.$$

S_{13}: Total time of one repetition of this cycle is $16\varepsilon + 18\delta + P_{i1} + P_{i2} + P_{(i+1)1} + w_{i3} + w_{(i+1)2} + P_{(i+1)3}$, where $w_{i3} = (P_{i3} - 4\varepsilon - 6\delta - P_{(i+1)1})^+$ and $w_{(i+1)2} = (P_{(i+1)2} - 2\varepsilon - 4\delta - w_{i3})^+$.

Since $P_{(i+1)2} - 2\varepsilon - 4\delta - w_{i3} \leqslant w_{(i+1)2}$, we have $16\varepsilon + 18\delta + P_{(i+1)1} + w_{i3} + P_{(i+1)2} - 2\varepsilon - 4\delta - w_{i3} + P_{(i+1)3} \leqslant 16\varepsilon + 18\delta + P_{i1} + P_{i2} + P_{(i+1)1} + w_{i3} + w_{(i+1)2} + P_{(i+1)3}$. Thus, a lower bound for the time to produce one part can be found as

$$\underline{T_{S_{13(\Pi_k^*)}}} = \tfrac{1}{2}(P + 14\varepsilon + 14\delta) > \underline{T_2}.$$

S_{14}: Total time of one repetition of this cycle is $16\varepsilon + 20\delta + P_{i1} + P_{i2} + w_{(i+1)1} + w_{i3} + P_{(i+1)2} + P_{(i+1)3}$, where $w_{i3} = (P_{i3} - 2\varepsilon - 6\delta)^+$ and $w_{(i+1)1} = (P_{(i+1)1} - 2\varepsilon - 6\delta - w_{i3})^+$.

Since $P_{(i+1)1} - 2\varepsilon - 6\delta - w_{i3} \leqslant w_{(i+1)1}$, we have $16\varepsilon + 20\delta + P_{(i+1)1} - 2\varepsilon - 6\delta - w_{i3} + w_{i3} + P_{(i+1)2} + P_{(i+1)3} \leqslant 16\varepsilon + 20\delta + P_{i1} + P_{i2} + w_{(i+1)1} + w_{i3} + P_{(i+1)2} + P_{(i+1)3}$. Thus, a lower bound for the time to produce one part can be found as

$$\underline{T_{S_{14(\Pi_k^*)}}} = \tfrac{1}{2}(P + 14\varepsilon + 14\delta) > \underline{T_2}.$$

S_{15}: Total time of one repetition of this cycle is $16\varepsilon + 18\delta + P_{i1} + w_{i2} + w_{(i+1)1} + P_{i3} + P_{(i+1)2} + P_{(i+1)3}$, where $w_{i2} = (P_{i2} - 2\varepsilon - 4\delta)^+$ and $w_{(i+1)1} = (P_{(i+1)1} - 4\varepsilon - 6\delta - w_{i2} - P_{i3})^+$.

Since $P_{i2} - 2\varepsilon - 4\delta \leqslant w_{i2}$ and $0 \leqslant w_{(i+1)1}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{15(\Pi_k^*)}}} = \tfrac{1}{2}(P + 14\varepsilon + 14\delta) > \underline{T_2}.$$

S_{23}: Total time of one repetition of this cycle is $16\varepsilon + 22\delta + P_{i1} + w_{(i-1)3} + w_{i2} + w_{(i+1)1} + w_{i3} + w_{(i+1)2}$, where $w_{(i-1)3} = (P_{(i-1)3} - 4\varepsilon - 6\delta - P_{i1})^+$, $w_{i2} = (P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3})^+$, $w_{(i+1)1} = (P_{(i+1)1} - 2\varepsilon - 4\delta - w_{i2})^+$, $w_{i3} = (P_{i3} - 2\varepsilon - 4\delta - w_{(i+1)1})^+$ and $w_{(i+1)2} = (P_{(i+1)2} - 2\varepsilon - 6\delta - w_{i3})^+$.

Since $P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3} \leqslant w_{i2}$, $P_{i3} - 2\varepsilon - 4\delta - w_{(i+1)1} \leqslant w_{i3}$ and $0 \leqslant w_{(i+1)2}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{23(\Pi_k^*)}}} = \tfrac{1}{2}(P + 10\varepsilon + 10\delta) > \underline{T_2}.$$

S_{24}: Total time of one repetition of this cycle is $16\varepsilon + 24\delta + w_{(i-1)2} + w_{i1} + w_{(i-1)3} + w_{i2} + w_{i3} + w_{(i+1)1}$, where $w_{(i-1)2} = (P_{(i-1)2} - 2\varepsilon - 4\delta)^+$, $w_{i1} = (P_{i1} - 2\varepsilon - 4\delta - w_{(i-1)2})^+$, $w_{(i-1)3} = (P_{(i-1)3} - 2\varepsilon - 4\delta - w_{i1})^+$, $w_{i2} = (P_{i2} - 2\varepsilon - 4\delta - w_{(i-1)3})^+$, $w_{i3} = (P_{i3} - 2\varepsilon - 6\delta)^+$ and $w_{(i+1)1} = (P_{(i+1)1} - 2\varepsilon - 6\delta - w_{i3})^+$.

Since $P_{i1} - 2\varepsilon - 4\delta - w_{(i-1)2} \leqslant w_{i1}$, $P_{i2} - 2\varepsilon - 4\delta - w_{(i-1)3} \leqslant w_{i2}$, $P_{i3} - 2\varepsilon - 6\delta \leqslant w_{i3}$ and $0 \leqslant w_{(i+1)1}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{24(\Pi_k^*)}}} = \tfrac{1}{2}(P + 10\varepsilon + 10\delta) > \underline{T_2}.$$

S_{25}: Total time of one repetition of this cycle is $16\varepsilon + 22\delta + w_{(i-1)2} + w_{i1} + w_{(i-1)3} + w_{i2} + P_{i3} + w_{(i+1)1}$, where $w_{(i-1)2} = (P_{(i-1)2} - 2\varepsilon - 4\delta)^+$, $w_{i1} = (P_{i1} - 2\varepsilon - 4\delta - w_{(i-1)2})^+$, $w_{(i-1)3} = (P_{(i-1)3} - 2\varepsilon - 4\delta - w_{i1})^+$, $w_{i2} = (P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3})^+$ and $w_{(i+1)1} = (P_{(i+1)1} - 4\varepsilon - 6\delta - w_{i2} - P_{i3})^+$.

Since, $P_{i1} - 2\varepsilon - 4\delta - w_{(i-1)2} \leqslant w_{i1}$, $P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3} \leqslant w_{i2}$ and $0 \leqslant w_{(i+1)1}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{25(\Pi_k^*)}}} = \tfrac{1}{2}(P + 10\varepsilon + 10\delta) > \underline{T_2}.$$

S_{26}: Total time of one repetition of this cycle is $16\varepsilon + 24\delta + w_{(i-1)2} + w_{i1} + w_{(i-1)3} + w_{i2} + w_{i3} + w_{(i+1)1}$, where $w_{(i-1)2} = (P_{(i-1)2} - 4\varepsilon - 8\delta - w_{i3})^+$, $w_{i1} = (P_{i1} - 2\varepsilon - 4\delta - w_{(i-1)2})^+$, $w_{(i-1)3} = (P_{(i-1)3} - 4\varepsilon - 8\delta - w_{i1})^+$, $w_{i2} = (P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3})^+$, $w_{(i+1)1} = (P_{(i+1)1} - 4\varepsilon - 8\delta - w_{(i-1)3} - w_{i2})^+$ and $w_{i3} = (P_{i3} - 2\varepsilon - 4\delta - w_{(i+1)1})^+$.

Since $P_{i1} - 2\varepsilon - 4\delta - w_{(i-1)2} \leqslant w_{i1}$, $P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3} \leqslant w_{i2}$ and $P_{i3} - 2\varepsilon - 4\delta - w_{(i+1)1} \leqslant w_{i3}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{26(\Pi_k^*)}}} = \tfrac{1}{2}(P + 8\varepsilon + 8\delta) = \underline{T_2}.$$

S_{34}: Total time of one repetition of this cycle is $16\varepsilon + 22\delta + P_{i1} + w_{(i-1)3} + w_{i2} + w_{i3} + w_{(i+1)1} + P_{(i+1)2}$, where $w_{(i-1)3} = (P_{(i-1)3} - 4\varepsilon - 6\delta - P_{i1})^+$, $w_{i2} = (P_{i2} - 2\varepsilon - 4\delta - w_{(i-1)3})^+$, $w_{i3} = (P_{i3} - 2\varepsilon - 6\delta)^+$ and $w_{(i+1)1} = (P_{(i+1)1} - 2\varepsilon - 6\delta - w_{i3})^+$.

Since $P_{i2} - 2\varepsilon - 4\delta - w_{(i-1)3} \leqslant w_{i2}$, $P_{i3} - 2\varepsilon - 6\delta \leqslant w_{i3}$ and $0 \leqslant w_{(i+1)1}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{34(\Pi_k^*)}}} = \tfrac{1}{2}(P + 12\varepsilon + 12\delta) > \underline{T_2}.$$

S_{35}: Total time of one repetition of this cycle is $16\varepsilon + 20\delta + P_{i1} + w_{(i-1)3} + w_{i2} + P_{i3} + w_{(i+1)1} + P_{(i+1)2}$, where $w_{(i-1)3} = (P_{(i-1)3} - 4\varepsilon - 6\delta - P_{i1})^+$, $w_{i2} = (P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3})^+$ and $w_{(i+1)1} = (P_{(i+1)1} - 4\varepsilon - 6\delta - w_{i2} - P_{i3})^+$.

Since $P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3} \leqslant w_{i2}$ and $0 \leqslant w_{(i+1)1}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{35(\Pi_k^*)}}} = \tfrac{1}{2}(P + 12\varepsilon + 12\delta) > \underline{T_2}.$$

S_{36}: Total time of one repetition of this cycle is $16\varepsilon + 22\delta + P_{i1} + w_{(i+1)1} + w_{(i-1)3} + w_{i2} + w_{i3} + w_{(i+1)2}$, where $w_{(i-1)3} = (P_{(i-1)3} - 6\varepsilon - 10\delta - P_{i1})^+$, $w_{i2} = (P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3})^+$, $w_{(i+1)1} = (P_{(i+1)1} - 4\varepsilon - 8\delta - w_{(i-1)3} - w_{i2})^+$, $w_{i3} = (P_{i3} - 2\varepsilon - 4\delta - w_{(i+1)1})^+$ and $w_{(i+1)2} = (P_{(i+1)2} - 2\varepsilon - 4\delta - w_{i3})^+$.

Since $P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3} \leqslant w_{i2}$, $P_{i3} - 2\varepsilon - 4\delta - w_{(i+1)1} \leqslant w_{i3}$ and $0 \leqslant w_{(i+1)2}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{36(\Pi_k^*)}}} = \tfrac{1}{2}(P + 10\varepsilon + 10\delta) > \underline{T_2}.$$

S_{45}: Total time of one repetition of this cycle is $16\varepsilon + 22\delta + w_{(i-1)2} + P_{(i-1)3} + w_{i1} + P_{i2} + w_{i3} + w_{(i+1)1}$, where $w_{(i-1)2} = (P_{(i-1)2} - 2\varepsilon - 4\delta)^+$, $w_{i1} = (P_{i1} - 4\varepsilon - 6\delta - w_{(i-1)2} - P_{(i-1)3})^+$, $w_{i3} = (P_{i3} - 2\varepsilon - 6\delta)^+$ and $w_{(i+1)1} = (P_{(i+1)1} - 2\varepsilon - 6\delta - w_{i3})^+$.

Since $P_{i1} - 4\varepsilon - 6\delta - w_{(i-1)2} - P_{(i-1)3} \leqslant w_{i1}$, $P_{i3} - 2\varepsilon - 6\delta \leqslant w_{i3}$ and $0 \leqslant w_{(i+1)1}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{45(\Pi_k^*)}}} = \tfrac{1}{2}(P + 10\varepsilon + 10\delta) > \underline{T_2}.$$

S_{46}: Total time of one repetition of this cycle is $16\varepsilon + 20\delta + P_{i1} + w_{(i-1)3} + w_{i2} + P_{i3} + w_{(i+1)1} + w_{(i+1)2}$, where $w_{(i-1)3} = (P_{(i-1)3} - 6\varepsilon - 10\delta - P_{i1})^+$, $w_{i2} = (P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3})^+$ and $w_{(i+1)1} = (P_{(i+1)1} - 6\varepsilon - 10\delta - w_{(i-1)3} - w_{i2} - P_{i3})^+$.

Since $P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3} \leqslant w_{i2}$ and $0 \leqslant w_{(i+1)1}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{46(\Pi_k^*)}}} = \tfrac{1}{2}(P + 12\varepsilon + 12\delta) > \underline{T_2}.$$

S_{56}: Total time of one repetition of this cycle is $16\varepsilon + 22\delta + w_{(i-1)2} + w_{i1} + w_{(i-1)3} + w_{i2} + P_{i3} + w_{(i+1)1}$, where $w_{(i-1)2} = (P_{(i-1)2} - 2\varepsilon - 4\delta)^+$, $w_{i1} = (P_{i1} - 2\varepsilon - 4\delta - w_{(i-1)2})^+$, $w_{(i-1)3} = (P_{(i-1)3} - 4\varepsilon - 8\delta - w_{i1})^+$, $w_{i2} = (P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3})^+$ and $w_{(i+1)1} = (P_{(i+1)1} - 6\varepsilon - 10\delta - w_{(i-1)3} - w_{i2} - P_{i3})^+$.

Since $P_{i1} - 2\varepsilon - 4\delta - w_{(i-1)2} \leqslant w_{i1}$, $P_{i2} - 4\varepsilon - 8\delta - w_{(i-1)3} \leqslant w_{i2}$ and $0 \leqslant w_{(i+1)1}$, a lower bound for the total time to produce one part can be found as

$$\underline{T_{S_{56(\Pi_k^*)}}} = \tfrac{1}{2}(P + 10\varepsilon + 10\delta) > \underline{T_2}.$$

Thus, for any 2-unit cycle, S_{ij}, we showed that $\underline{T_2} \leqslant \underline{T_{S_{ij(\Pi_k^*)}}}$. Consequently, we can use $\underline{T_2}$ as a global lower bound for the 2-unit robot move cycles.

References

[1] Logendran R, Sriskandarajah C. Sequencing of robot activities and parts in two-machine robotic cells. International Journal of Production Research 1996;34:3447–63.

[2] Crama Y, Kats V, Van de Klundert J, Levner E. Cyclic scheduling in robotic flowshops. Annals of Operations Research 2000;96:97–124.

[3] Dawande M, Geismar HN, Sethi S, Sriskandarajah C. Sequencing and scheduling in robotic cells: recent developments. Journal of Scheduling 2005;8:387–426.

[4] Sethi SP, Sriskandarajah C, Sorger G, Blazewicz J, Kubiak W. Sequencing of parts and robot moves in a robotic cell. International Journal of Flexible Manufacturing Systems 1992;4:331–58.

[5] Crama Y, Van de Klundert J. Cyclic scheduling in 3-machine robotic flow shops. Journal of Scheduling 1999;4:35–54.

[6] Brauner N, Finke G. On a conjecture about robotic cells: new simplified proof for the three-machine case. INFOR 1999;37:20–36.

[7] Brauner N, Finke G. Cycles and permutations in robotic cells. Mathematical and Computer Modeling 2001;34:565–91.

[8] Hall NG, Kamoun H, Sriskandarajah C. Scheduling in robotic cells: classification, two and three machine cells. Operations Research 1997;45:421–39.

[9] Aneja YP, Kamoun H. Scheduling of parts and robot activities in two-machine robotic cell. Computers & Operations Research 1999;26:297–312.

[10] Geismar HN, Dawande M, Sethi S, Sriskandarajah C. Approximation algorithms for *k*-unit cyclic solutions in robotic cells. European Journal of Operational Research 2002;162:291–309.

[11] Akturk MS, Gultekin H, Karasan OE. Robotic cell scheduling with operational flexibility. Discrete Applied Mathematics 2005;145:334–48.

[12] Crama Y, Van de Klundert J. Cyclic scheduling of identical parts in a robotic cell. Operations Research 1997;45:952–65.