# SCHEDULING IN A THREE-MACHINE FLEXIBLE ROBOTIC CELL

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

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

Abstract: In this study, a three-machine flexible robotic manufacturing cell in which the CNC machines are used is considered. These machines are highly flexible and are capable of performing several different operations. Each machine is assumed to be capable of performing all of the required operations of each part. As a consequence of this assumption, a new class of cycles is defined and three simple and widely used cycles among this class is proposed. The regions of optimality for these cycles as well as the worst case performances are derived. 

Keywords: Flexible manufacturing systems, CNC, industrial robots, cellular automation, optimization, production control.

## 1. INTRODUCTION

A manufacturing cell which consists of a number of machines and a material handling robot is called a robotic cell. Such manufacturing cells are used extensively in chemical, electronic and metal cutting industries. In this study, we will restrict ourselves with the metal cutting applications in an environment in which the machines are predominantly CNC machines so that the machines and the robot can communicate in a real time basis. These machines are highly flexible and capable of performing several different operations by fast and inexpensive tool changes as long as the required tools are loaded in their tool magazines. There are no buffers at or between the machines. As a consequence, at any time instant, a part is either on one of the machines, on the robot or at the input or output buffer. Each of the identical parts to be produced is assumed to have a number of operations to be performed on the machines. As a consequence of the flexibility of the machines, these operations can be performed in any order on the three machines. Furthermore, each operation can be assigned to any one of the machines. In order to use such systems efficiently, problems including the scheduling of the robot moves and the allocation of the operations should be solved. Throughout this study, these problems will be tackled with the objective of minimizing the cycle time (long run average time to produce one part).

There is an extensive literature on robotic cell scheduling problems such as the surveys of (Crama *et al.*, 2000) and (Dawande *et al.*, 2005). (Sethi *et al.*, 1992) developed the necessary framework for these scheduling problems and proved that for two-machines producing identical parts, the optimal solution is a 1-unit cycle, where an $n$-unit cycle is defined to be a robot move cycle in which starting with an initial state, the robot loads and unloads all of the machines exactly $n$ times and returns back to the initial state. Note that, in an $n$-unit cycle exactly $n$ parts are produced. A similar result for three-machine case was proved by (Crama and de Klundert, 1999). However, the optimal solution is not necessarily a 1-unit cycle when the number of machines is greater than three (Brauner and Finke, 2001). Flexible robotic cells have recently been a topic of research. For example, in (Akturk *et al.*, 2005) a robotic cell with two identical CNC machines possessing operational and process flexibility was considered. For this problem, they proved that the optimal solution is either one of the two 1-unit cycles or a 2-unit cycle. A similar result is proved to hold in a more general setting where the tooling issues are also considered (Gultekin *et al.*, 2005*a*). In this study, the sufficient conditions for optimality for each robot move cycle are derived.

CNC machines possess several types of flexibilities such as the operational flexibility which is the ability to change the ordering of several operations and process flexibility which is 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. This study will focus on the consequences of introducing such machine flexibilities to our system. A new class of robot move cycles which are direct consequences of operational and process flexibilities will be defined. We will consider three specific cycles among this huge class and derive the worst case performance of these three cycles. We will also compare these cycles with the classical flowshop type robot move cycles and show that the proposed cycles dominate all flowshop type robot move cycles except a small region.

In the following section the notation and basic assumptions pertinent to this study will be introduced and the operation allocation problem will be defined. In Section 3, a new class of robot move cycles will be proposed. Three simple and widely used robot move cycles from this class will be compared with the rest of such cycles and classical flowshop type robot move cycles. A worst case performance bound of using these three proposed cycles will also be derived. Section 4 is devoted to the concluding remarks and future research directions.

## 2. PROBLEM DEFINITION

Let us first highlight the differences of a classical flowshop type robotic cell and a flexible robotic cell. In the former one, the processing time of each part on each machine is a known parameter and the problem is to find the robot move sequence that minimizes the cycle time. However, in a flexible robotic cell the sequence of the robot moves as well as the processing times of the parts on the machines are decision variables. More specifically, the identical parts have a number of operations to be completed on the 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 machines depend on the allocation of these operations to the machines. An allocation of operations to the $m$ machines entails partitioning set $O$ into $m$ subsets; $O_1, O_2, \ldots, O_m$, 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. Since during one repetition of the cycle more than one part can be processed on different machines at the same time, having different allocations for the 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 where $k = 1, 2, \ldots$, becomes identical with the first part. Hence, 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. The following definition and notations will be used throughout the paper.

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

Note that each row of $\Pi_k$ 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 $\pi_{i2} \cap \pi_{i3} = \emptyset$. Furthermore, no two rows are identical. We also let $\Pi_k^*$ denote the optimal allocation of operations when a total of $k$ different allocation types is used.

In particular, for a cycle in which a 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 alternatingly. 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, $P = \sum_{l=1}^{p} t_l$.

$P_{ij}$ : Total processing time on machine $j$ for the part which corresponds to the $i^{th}$ row of the specific allocation matrix $\Pi$. That is, $P_{ij} = \sum_{l \in \pi_{ij}} t_l$. Also, we let $P_\pi = [P_{ij}]$.

$\epsilon$ : 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.

$\delta$ : 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 $\Pi_k$.

The following definition borrowed from (Crama and de Klundert, 1997) is used to define the flowshop type robot move cycles.

*Definition 2.* Robot activity $A_i$ consists of the following moves of the robot: unload a part from machine $i$, transport it to machine $i+1$, and load machine $i+1$.

As already mentioned, for two- (Sethi *et al.*, 1992) and three-machine cells (Crama and de Klundert, 1999) producing identical parts, the optimal solution is a 1-unit cycle. However, in these studies, the processing times are assumed to be fixed on each machine for each part. With operation and process flexibilities, this assumption must be relaxed. The following example is crucial in understanding the difference of this study from the classical robotic cell scheduling literature.

*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 $\epsilon = 2$ 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 as in (Sethi *et al.*, 1992) corresponds to the case where the allocations of the operations of all parts are identical. Let $\Pi_1$ be a specific allocation. Then, the cycle time for this case is the following:

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

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 $\Pi_2$. The new cycle time to produce one part for this case is the following:

$$T_{S6}(\Pi_2) = 8\epsilon + 12\delta$$
$$+\frac{1}{2}max\{0, P_{11} - 4\epsilon - 8\delta, P_{12} - 4\epsilon - 8\delta, P_{13} - 4\epsilon - 8\delta\}$$
$$+\frac{1}{2}max\{0, P_{21} - 4\epsilon - 8\delta, P_{22} - 4\epsilon - 8\delta, P_{23} - 4\epsilon - 8\delta\}$$

The optimal allocations of the operations, in the first allocation type are, $\pi_{11}^* = \{1, 2\}$, $\pi_{12}^* = \{3\}$, $\pi_{13}^* = \{4, 5\}$. In other words, $P_{11}^* = 55$, $P_{12}^* = 35$ and $P_{13}^* = 45$. As for the second allocation type, $\pi_{21}^* = \{4, 5\}$ with $P_{21}^* = 45$, $\pi_{22}^* = \{1, 2\}$ with $P_{22}^* = 55$, and finally $\pi_{23}^* = \{3\}$ with $P_{23}^* = 35$. Then the corresponding cycle time becomes the following:

$$T_{S6}(\Pi_2^*) = 64 + \frac{1}{2}max\{0, 55 - 40, 55 - 40, 45 - 40\}$$
$$+\frac{1}{2}max\{0, 45 - 40, 35 - 40, 35 - 40\} = 74$$

The Gantt chart in Figure 1 compares these two cases. In order to make a valid comparison, 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 difference between the cycle times of these two cases.

The following theorem for which a detailed proof can be found in (Gultekin *et al.*, 2005*b*), derives a lower bound for the cycle times of the flowshop type robot move cycles.

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

$$\underline{T_{flowshop}} = max\{8(\epsilon + \delta) + min\{P, \delta\}, 4\epsilon + 4\delta + (P/3)\}$$

With the assumption of process and operation flexibilities, even in two-machine cells, the optimal solution may not be a 1-unit cycle as shown in (Gultekin *et al.*, 2005*a*). In this particular study, the authors assumed that the machines are capable of performing a set of different operations since they are loaded with the required tools. However, in most practical applications the number of required cutting tools for the CNC machines to

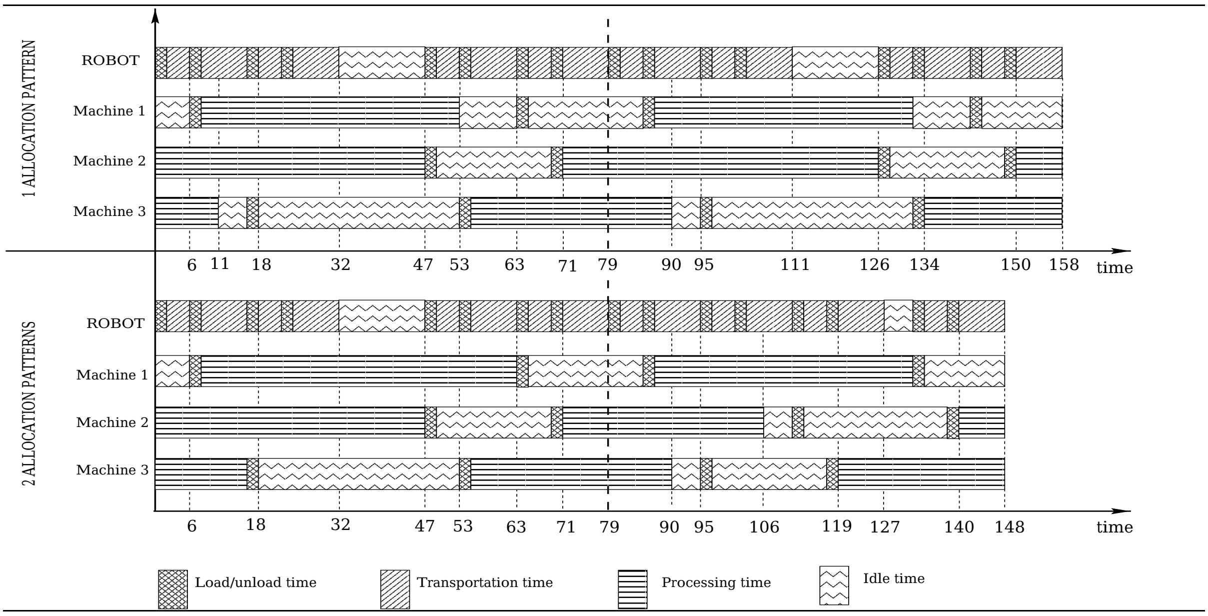


Fig. 1. Gantt chart for example 1

perform all operations of a part exceeds the tool magazine capacity. Additionally, though duplicating the tools increases flexibility, duplicating all of them may not be economically justifiable. Hence, they assumed that some of the required tools are loaded only on the first machine and some others are loaded only on the second machine. A third set of tools are duplicated and loaded on both machines. As a result of this, an operation can either be processed only on the first machine, only on the second machine or on either machine. The problem is not only sequencing the robot's activities but also partitioning the set of flexible operations into two machines with the objective of minimizing the cycle time. It is proved that the optimal solution is either one of the two 1-unit cycles; $S1$ defined by the activity sequence $A_0A_1A_2$ or $S2$ defined as $A_0A_2A_1$ or the only 2-unit cycle $S_{12}S_{21}$ defined as $A_0A_1A_0A_2A_1A_2$. The regions of optimality which depend on the allocation of the operations for the $S2$ cycle are also presented followed by a sensitivity analysis on parameters such as the loading/unloading time, $\epsilon$ and robot travel time, $\delta$.

Note that this general problem has two special cases. The first one is considered by (Sethi *et al.*, 1992), which assumes that specific operations are performed on each machine so that operation allocation problem vanishes. In this case the optimal solution appears to be one of the 1-unit cycles. The second special case assumes that each machine is capable of performing all of the required operations of a part (Akturk *et al.*, 2005). In this case, similar to the general problem, the optimal solution is either one of the 1-unit cycles or a 2-unit cycle. Note that for the $S1$ cycle, the allocation of the operations does not affect the cycle time whereas for $S2$ and $S_{12}S_{21}$, the allocation of the operations affects the cycle time.

## 3. PURE CYCLES

In this section, we will define new robot move cycles which are direct consequences of the operation and process flexibilities. In the sequel, we will define these cycles, compare them with the classical flowshop type robot move cycles and also with themselves to determine the sufficient conditions for optimality. In order to represent these cycles we need the following definitions:
$A_{0i}$ = The robot activity in which the robot takes a part from the input buffer and loads machine $i = 1, 2, \ldots, m$.
$A_{i(m+1)}$ = The robot activity in which the robot unloads machine $i$ and drops the part to the output buffer where $i = 1, 2, \ldots, m$.

In an $m$-machine robotic cell there are exactly $2m$ activities. By using these activities we can define new cycles as follows:

*Definition 4.* Under a **pure cycle**, starting with an initial state, the robot performs each of the $2m$ activities $(A_{0i}, A_{i(m+1)},\ \ i = 1, \ldots, m)$ exactly once and the final state of the system is identical with the initial state.

According to this definition, the pure cycles can be represented by the permutations the activities $A_{0i}$ and $A_{i(m+1)}$, $i = 1, 2, \ldots, m$ and each permutation yields a different feasible pure cycle. After eliminating different representations of the same cycles, in an $m$-machine cell there are a total of $(2m-1)!$ pure cycles. The number of pure cycles increases drastically as the number of machines in a cell increases. In this study, we will consider 3-machine cells in which there are a total of 120 pure cycles. With this many different pure cycles, finding the best and later comparing it against all the classical flowshop type robot move cycles is extremely cumbersome and hence omitted from the scope of the current paper. Instead, we focus on the three simplest and most widely used pure cycles.

Let $C1$ denote the pure cycle which can be represented as $A_{01}A_{02}A_{03}A_{14}A_{24}A_{34}$. 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. Also let $C2$ denotes the pure cycle represented as $A_{01}A_{34}A_{03}A_{24}A_{02}A_{14}$ and $C3$ denote $A_{01}A_{24}A_{02}A_{34}A_{03}A_{14}$. Note that three parts are produced by a single repetition of these cycle, hence they are called 3-unit cycles. (The animated views of some of the pure and flowshop type robot move cycles can be found at the web site http://www.ie.bilkent.edu.tr/∼robot).

The cycle times for these can be derived to be as follows:

$$T_{C1} = \frac{1}{3}(4\epsilon + 8\delta + max\{0, P - 4\epsilon - 10\delta)\}$$

$$T_{C2} = \frac{1}{3}(12\epsilon + 28\delta + max\{0, P - 8\epsilon - 20\delta)\}$$

$$T_{C3} = \frac{1}{3}(12\epsilon + 28\delta + max\{0, P - 8\epsilon - 20\delta)\}$$

As it is obvious, $T_{C2} = T_{C3}$. Hence, from now on we will not consider cycle $C3$, which performs equally well as cycle $C2$. The following theorem compares cycles $C1$ and $C2$ with each other.

*Theorem 5.* If $P < 4\epsilon + 14\delta$, $C1$ dominates $C2$; else if $P > 4\epsilon+14\delta$, $C2$ dominates $C1$. If $P = 4\epsilon+14\delta$ both cycles perform equally well.

**PROOF.** We will compare the cycle times of these two cycles in the following cases:

1. If $P \leq 4\epsilon + 10\delta$,
   $T_{C1} = 4\epsilon + 8\delta \leq 4\epsilon + (28/3)\delta = T_{C2}$
2. If $4\epsilon + 10\delta < P \leq 8\epsilon + 20\delta$,
   $T_{C1} = 1/3(8\epsilon + 14\delta + P)$. If $P = 4\epsilon + 14\delta$, $1/3(8\epsilon + 14\delta + P) = 4\epsilon + (28/3)\delta = T_{C2}$. Hence, if $P < 4\epsilon + 14\delta$, $T_{C1} < T_{C2}$. Else If $P > 4\epsilon + 14\delta$, $T_{C1} > T_{C2}$.
3. If $P > 8\epsilon + 20\delta$,
   $T_{C1} = 1/3(8\epsilon + 14\delta + P) \geq 1/3(4\epsilon + 8\delta + P) = T_{C2}$

This completes the proof. □

Now we will determine the performance of the selected pure cycles with respect to other pure cycles. In the following theorem we determine a lower bound for the pure cycles.

*Theorem 6.* For a three-machine robotic cell, the cycle time of any pure cycle is no less than

$$\underline{T_{pure}} = max\{4\epsilon + 8\delta, \frac{4\epsilon + 8\delta + P}{3}\}$$

**PROOF.** The first argument results from the following observation: any part to be produced with one of the pure cycles is taken from the input buffer ($\epsilon$), loaded to one of the machines and unloaded after the processing is completed ($2\epsilon$) and dropped to the output buffer ($\epsilon$), which makes a total of $4\epsilon$. Also for each part, the robot travels from the input buffer to output buffer and returns back either to take another part or to complete the cycle which makes $8\delta$. On the other hand, the second argument of the lower bound is the minimum time between two consecutive loadings of any machine. After loading any machine, minimum time required before the robot can unload it is $P$. Then, the robot unloads the machine, travels to output buffer and drops the part, travels to input buffer and brings another part and loads the machine. Since one repetition of this cycle produces three parts, the total time is divided by 3. □

*Corollary 7.* $\underline{T_{pure}} \leq \underline{T_{flowshop}}$

**PROOF.** Found by a simple comparison of $\underline{T_{pure}}$ with $\underline{T_{flowshop}}$. □

This corollary states that the lower bound for the pure cycles is also a lower bound for the flowshop type robot move cycles. The following theorem makes use of this to determine the worst case performances of the pure cycles $C1$ and $C2$ with respect to all pure cycles and all flowshop type robot move cycles.

*Theorem 8.* Let $T^*$ be the cycle time of the best pure or flowshop type robot move cycle. Then the following holds:

1. If $P \leq 4\epsilon + 10\delta$, $T_{C1} = T^*$
2. Else if $4\epsilon + 10\delta < P \leq 4\epsilon + 14\delta$, $T_{C1} \leq (1 + 1/6) \cdot T^*$
3. Else if $4\epsilon + 14\delta < P \leq 8\epsilon + 20\delta$, $T_{C2} < (1 + 1/2) \cdot T^*$
4. Else if $P \geq 8\epsilon + 20\delta$, $T_{C2} = T^*$

**PROOF.**

1. If $P \leq 4\epsilon + 10\delta$,
   $$T_{C1} = 4\epsilon + 8\delta = \underline{T_{pure}}$$
2. If $4\epsilon + 10\delta < P \leq 4\epsilon + 14\delta$,
   $$\frac{T_{C1}}{T^*} \leq \frac{1/3(8\epsilon + 14\delta + P)}{4\epsilon + 8\delta}$$

Since $P \leq 4\epsilon + 14\delta$,

$$\frac{T_{C1}}{T^*} \leq \frac{12\epsilon + 28\delta + P}{12\epsilon + 24\delta} \leq 1 + 1/6$$

3. If $4\epsilon + 14\delta < P \leq 8\epsilon + 20\delta$,

$$\frac{T_{C2}}{T^*} \leq \frac{1/3(12\epsilon + 28\delta)}{1/3(4\epsilon + 8\delta + P)}$$

Since $P \geq 4\epsilon + 14\delta$,

$$\frac{T_{C1}}{T^*} \leq 1 + 1/2 - \frac{5\delta}{8\epsilon + 22\delta} < 1 + 1/2$$

4. If $P \geq 8\epsilon + 20\delta$,

$$T_{C2} = 1/3(4\epsilon + 8\delta + P) = \underline{T_{pure}}$$

□

The following theorem compares these pure cycles with the flowshop type robot move cycles.

*Theorem 9.* All flowshop type robot move cycles are dominated by either $C1$ or $C2$ in all regions except $(\delta > 12\epsilon) \wedge (16\epsilon + 13\delta < P < 16\delta)$.

**PROOF.** Corollary 7 and cases 1 and 2 of Theorem 8 together proves that all flowshop robot move cycles are dominated by either $C1$ or $C2$ for $P \leq 4\epsilon + 10\delta$ and $P \geq 8\epsilon + 20\delta$. The remaining regions will be analyzed in the following cases:

1. If $4\epsilon + 10\delta < P \leq 4\epsilon + 14\delta$, $T_{C1} \leq T_{C2}$, $T_{C1} = 1/3(8\epsilon + 14\delta + P)$ and $\underline{T_{flowshop}} = 8\epsilon + 9\delta$. Comparing these two, we can conclude that $\underline{T_{flowshop}} > T_{C1}$ if $P > 16\epsilon + 13\delta$. However, if $\delta \leq 12\epsilon$, $16\epsilon + 13\delta \geq 4\epsilon + 14\delta$. Hence, we must consider $\delta > 12\epsilon$.
2. If $(P > 4\epsilon + 14\delta) \wedge (P < 8\epsilon + 20\delta) \wedge (P \leq 12\epsilon + 15\delta)$, $T_{C2} \leq T_{C1}$, $T_{C2} = 4\epsilon + (28/3)\delta$ and $\underline{T_{flowshop}} = 8\epsilon + 9\delta$. Comparing these, $\underline{T_{flowshop}} > T_{C2}$ if $\delta > 12\epsilon$. With this setting of $\delta$, $12\epsilon + 15\delta < 8\epsilon + 20\delta$.
3. If $(P > 4\epsilon + 14\delta) \wedge (P < 8\epsilon + 20\delta) \wedge (P > 12\epsilon + 15\delta)$, $T_{C2} \leq T_{C1}$, $T_{C2} = 4\epsilon + (28/3)\delta$ and $\underline{T_{flowshop}} = 4\epsilon + 4\delta + P/3$. Comparing these, $\underline{T_{flowshop}} > T_{C2}$ if $P > 16\delta$. However, if $\delta \leq 12\epsilon$, $16\delta \leq 12\epsilon + 15\delta$. Hence, we must consider $\delta > 12\epsilon$.

This completes the proof. □

## 4. CONCLUSION AND FUTURE RESEARCH

In this study, a three-machine robotic cell used for metal cutting operations is considered. The machines used in such manufacturing cells are CNC machines which are highly flexible. As a consequence, each part is assumed to be composed of a number of operations and each machine is assumed to be capable of performing all of the required operations of each part. We investigated the productivity gain attained by the additional flexibility introduced by the CNC machines. A new class of robot move cycles, called pure cycles, which are resulted from operational and process flexibilities are defined. We selected three simple and most widely used robot move cycles from this huge class and compared them with each other to find the regions where each dominates the others. Lower bounds for pure cycles and classical flowshop type robot move cycles are derived and compared with the proposed cycles. The results show that these proposed cycles are not only simple and practical but performs very efficiently. Extending the analysis to the $m$-machine case can be considered as a future research direction.


## ACKNOWLEDGEMENTS

This work supported in part by the Scientific and Technical Research Council of Turkey, TUBITAK, under grant #2211.


## REFERENCES


Akturk, M.S., H. Gultekin and O.E. Karasan (2005). Robotic cell scheduling with operational flexibility. *Discrete Applied Mathematics* **145**(3), 334–348.

Brauner, N. and G. Finke (2001). On cycles and permutations in robotic cells. *Mathematical and Computer Modeling* **34**, 565–591.

Crama, Y. and J. Van de Klundert (1997). Cyclic scheduling of identical parts in a robotic cell. *Operations Research* **45**(6), 952–965.

Crama, Y. and J. Van de Klundert (1999). Cyclic scheduling in 3-machine robotic flow shops. *Journal of Scheduling* **4**, 35–54.

Crama, Y., V. Kats, J. Van de Klundert and E. Levner (2000). Cyclic scheduling in robotic flowshops. *Annals of Operations Research* **96**, 97–124.

Dawande, M., H.N. Geismar, S. Sethi and C. Sriskandarajah (2005). Sequencing and scheduling in robotic cells: Recent developments. *Journal of Scheduling* **8**, 387–426.

Gultekin, H., M.S. Akturk and O.E. Karasan (2005*a*). Robotic cell scheduling with tooling constraints. *European Journal of Operational Research.* to appear.

Gultekin, H., M.S. Akturk and O.E. Karasan (2005*b*). Scheduling in a three-machine robotic flexible manufacturing cell. *Computers & Operations Research.* to appear.

Sethi, S.P., C. Sriskandarajah, G. Sorger, J. Blazewicz and W. Kubiak (1992). Sequencing of parts and robot moves in a robotic cell. *International Journal of Flexible Manufacturing Systems* **4**, 331–358.