Computers & Operations Research 39 (2012) 1290–1299

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

An analysis of cyclic scheduling problems in robot centered cells

Serdar Yildiz, Oya Ekin Karasan, M. Selim Akturk *

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

ARTICLE INFO

Available online 21 September 2010

Keywords:
Robot centered cell
CNC
Scheduling
Bicriteria optimization
Controllable processing times

ABSTRACT

The focus of this study is a robot centered cell consisting of m computer numerical control (CNC) machines producing identical parts. Two pure cycles are singled out and further investigated as prominent cycles in minimizing the cycle time. It has been shown that these two cycles jointly dominate the rest of the pure cycles for a wide range of processing time values. For the remaining region, the worst case performances of these pure cycles are established. The special case of 3-machines is studied extensively in order to provide further insight for the more general case. The situation where the processing times are controllable is analyzed. The proposed pure cycles also dominate the rest when the cycle time and total manufacturing cost objectives are considered simultaneously from a bicriteria optimization point of view. Moreover, they also dominate all of the pure cycles in in-line robotic cells. Finally, the efficient frontier of the 3-machine case with controllable processing times is depicted as an example.

1. Introduction

The foundation of new automation technologies and the technological advancements which improve the efficiency of automation equipments increased the importance of automation applications in manufacturing industry. Robots are one of the most common automation equipments used in industry and they are mostly used as material handling tools. In the current literature, a robotic cell is defined as a manufacturing cell composed of a number of machines and a material handling robot. There are different robotic cell layouts studied in the literature, namely, in-line robotic cells, robot centered cells, and mobile-robot cells. In in-line robotic cells, the machines are positioned in a linear formation and the robot moves in front of the machines on a linear track to transport parts. Most of the studies in robotic cell scheduling literature focus on in-line robotic cells or mobile-robot cells.

An extensive literature review of robotic cell scheduling is presented in the survey of Dawande et al. [3]. In addition, Crama et al. [2] present the cyclic scheduling problems in robotic flowshops. In robotic flowshops, each part is processed on all of the machines in the cell in the order respecting the layout. In general, the processing time on each machine is assumed to be fixed. However, the recent developments in process and operational flexibility challenge the necessity and accuracy of this assumption. Furthermore, the existing studies work on a single objective of maximizing throughput. In manufacturing industry, however, the focus is on minimizing cost as well as on maximizing throughput, simultaneously. In addition, most of the studies are limited to 1-unit cycles in 2- or 3-machine cells. Although this configuration is easier to analyze, it may not be realistic for some manufacturing settings. Sethi et al. [12] proved that 1-unit cycles give optimal solutions in 2-machine robotic cells producing identical parts. For a more detailed discussion on cyclic scheduling of identical parts in robotic cells, we refer the interested reader to Brauner [1]. In our study, we consider a scheduling problem of an m-machine flexible robotic cell with m-unit cycles producing identical parts. Our study differs from the literature, since we consider process and operational flexibility and m-unit cycles in m-machine cells.

There are few studies in the literature working on the scheduling problems in robot centered cells. As Han and Cook [9] mention, robot centered cells can improve the efficiency in the cell. The focus of this study is on the robot centered cells in which the robot is placed in the center of the cell and the machines are positioned in a circular formation around the robot. The robot rotates between the buffer and the machines in order to transfer the parts. The robot centered cells are used in many applications because of their space efficiency compared to in-line robotic cells as discussed in Gultekin et al. [6]. In addition, the installation cost of stationary base robots which are used in robot centered cells is less and the programming of these robots are easier compared to in-line robotic cells. The robot centered cell considered in this study is presented in Fig. 1. There is an I/O-station which is composed of an input device that contains the raw parts to be processed in the cell and an output device that stores the parts produced in the cell. Consistent with many studies in the literature, we assume the parts to be produced in the cell are identical requiring the same set of processes to be performed. Moreover, we assume that

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

0305-0548/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2010.09.005

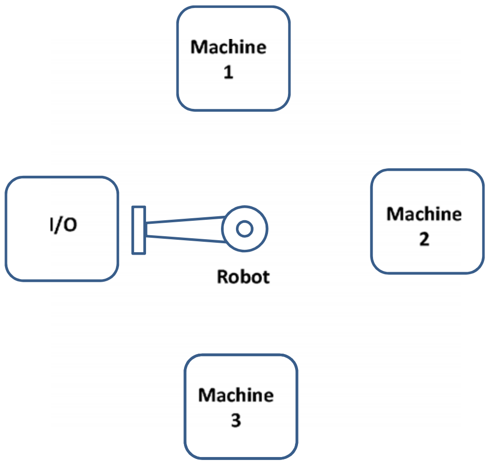

Fig. 1. 3-Machine robot centered cell.

there is no buffer between any machines. In a recent study, Drobouchevitch et al. [5] consider the problem of finding an optimal robot move sequence that maximizes the throughput in robot centered cells including an *I*/*O*-station. Dawande et al. [4] study the multiple part-type production in a robot centered cell. In both studies, each part must go through *m*-machines in the same sequence, a setting known as flowshop type robotic cell.

In a robotic cell for machining operations, the processing stations are predominantly CNC machines and these machines can communicate with the robot as well as with the cell controller on a real-time basis. The operational flexibility of CNC machines enables them to perform different operations on parts. As a result of operational flexibility, in a recent study, Gultekin et al. [8] defined a new class of cycles called pure cycles. In a pure cycle, the robot loads and unloads all of the *m* machines with a different part during one repetition of the cycle. So, each repetition of a pure cycle produces *m* parts. By using this definition, the robot and part movement in our study is described as follows: The robot transfers a part from the *I*/*O*-station to one of the machines. After all the operations on the part is finished, the robot transfers part from the machine to *I*/*O*-station again. Since Gultekin et al. [8] proved that pure cycles dominate all of the flowshop type cycles for the single objective problem of maximizing throughput, we focus on pure cycles in our study.

The scheduling literature on controllable processing times is presented in the survey paper of Shabtay and Steiner [13]. Within the bicriteria context of minimizing the cycle time and the total manufacturing cost simultaneously, the processing times are considered as controllable. Most of the studies in robot centered cells consider fixed processing times which are easier to analyze. However, process flexibility results in controllable processing times in which the processing times can be increased or decreased without violating a given upper bound in order to increase efficiency. To the knowledge of the authors, the only studies within the robotic cell scheduling literature which consider the bicriteria optimization problem of minimizing the total manufacturing cost and the cycle time simultaneously are Gultekin et al. [7] and Yildiz et al. [15], the former focusing on the flowshop setting and the latter focusing on the in-line setting. Furthermore, there are some studies such as Crama et al. [2], van de Klundert [10], and Lei and Wang [11] on robotic flowshops where the processing times are specified by a lower bound and an upper bound, i.e. processing time windows. Different than our study, the studies on processing time windows do not consider the manufacturing cost associated with the selected processing time.

The study is organized as follows: in Section 2, the assumptions and definitions used throughout this study are presented. In Section 3, we analyze the *m*-machine robot centered cell with fixed processing times in order to minimize the cycle time. In Section 4, different from the existing literature, we consider controllable processing times in *m*-machine robot centered cells and prove that the robot centered cells increase the efficiency of the cell when compared with in-line robotic cells. Furthermore, we determine the robot move sequence and the processing times that minimize the cycle time and the total manufacturing cost simultaneously. In Section 5, the concluding remarks and future research directions are presented.

2. Assumptions and definitions

In this section, we present the preliminary background information and set up the notation to be used throughout the remaining text. In this study, we consider identical parts to be processed on identical CNC machines and focus on a new class of cycles introduced to the literature in Gultekin et al. [8] as pure cycles. We assume that each machine is capable of performing all of the required operations of identical parts. The following definitions are borrowed from Gultekin et al. [8].

Definition 1. L_i is the robot activity during which the robot takes a part from the input buffer and loads machine $i=1,2,\ldots,m$. Similarly, U_i, $i=1,2,\ldots,m$, is the robot activity corresponding to movements while the robot unloads machine i and drops the part to the output buffer. Let $\mathcal{A}=(L_1,\ldots,L_m,\ U_1,\ldots,U_m)$ be the set of all activities.

A pure cycle is composed of m loading and m unloading activities and can be defined as follows:

Definition 2. Under a *pure cycle*, starting with an initial state, the robot performs each of the $2m$ activities $\{L_1,\ldots,L_m,\ U_1,\ldots,U_m\}$ exactly once and the final state of the system is identical with the initial state.

In particular, for a 2-machine robotic cell the robot activity sequence $L_1U_2L_2\ U_1$ constitutes a pure cycle. Since a repetition of any pure cycle produces m parts, pure cycles are classified as m-unit cycles.

In the considered cell, the input and the output devices are combined in an *I*/*O*-station. Within our setting, all of the required operations on a part are processed only on one machine. Thus, the only possible part movements are defined as follows: the robot takes a part from the input device at the *I*/*O*-station and loads it onto one of the machines. After all of the required operations on the part are finished, the robot unloads the part from the machine and drops the part to the output device at the *I*/*O*-station. Let C_i^m denote the ith pure cycle in an m-machine cell and $T_{C_i^m}$ denote its corresponding cycle time, i.e. the total time required to complete an m-unit pure cycle. We shall adopt the following notation throughout this study:

- δ : The time required for rotational movement between two consecutive machines. Since this is assumed to be additive, the traveling time between machine i and j is $\min\{|i-j|, m+1-|i-j|\}\delta$.
- ε : The load/unload times of machines by the robot which are assumed to be the same for all machines.
- P: Total processing time of any one of the identical parts on any one of the identical machines.

3. Problem definition and analysis

The number of different pure cycles in an m-machine cell is $(2m-1)!$. In this section, we single out two of the pure cycles as potentially the most prominent ones in minimizing the cycle time or in other words in maximizing the throughput rate. A similar approach is undertaken in Yildiz et al. [15] for in-line robotic cells.

Analyzing the structure of the cycle time of pure cycles, it is apparent that the cycle time of a pure cycle is composed of two components. The first component is the total time required for the robot activities and the second one is the total waiting time of the robot in front of the machines before unloading them. The time required for robot activities in turn is composed of load/unload and part transportation times. The time required for the robot load/unload times is calculated as follows: for each part, the part is taken from *I/O*-station (ε), then loaded onto machine i (ε), after all of the operations are finished the part is unloaded from machine i (ε) and finally the part is dropped into *I/O*-station (ε), which makes a total of 4ε time units for one part. For an m-machine cell, a pure cycle produces m parts, thus the total time required for loading/unloading is $4m\varepsilon$ and it is the same for all possible pure cycles for such a cell. However, the robot travel time and the total waiting time differ according to the robot move sequence. Let the total robot travel time for pure cycle C_i^m be $a_i\delta$ and the total waiting time at machine k be w_k. Now, the cycle time of pure cycle C_i^m can be presented as

$$T_{C_i^m} = 4m\varepsilon + a_i\delta + w_1 + w_2 + \cdots + w_m. \quad (1)$$

There could be two different approaches to minimize the cycle time in Eq. (1). The first approach is to minimize the robot travel time. If the processing times are small or negligible, this approach is more efficient in order to minimize the cycle time. The second approach is minimizing the total waiting times. In this study, we focus on the second approach, since it is more frequently observed in practice.

The waiting time of machine k can be represented as $w_k = \max\{0, P-v_k\}$ where v_k is defined as the amount of time between just after loading the machine k and the time robot returns back in front of machine k to unload it. Since P is a constant, in order to reduce this waiting time, we have to find the pure cycles resulting in higher v_k values. To do this, the loading activity of machine k should be immediately sequenced after unloading activity in the robot move sequence, and hence U_kL_k should be the activity sequence. In order to minimize the total waiting time, all of the individual waiting times on all machines have to be minimized. Thus, for each machine, the loading activity has to be immediately sequenced after the unloading activity. The resulting robot move sequence is

$$U_{\pi(1)}L_{\pi(1)}U_{\pi(2)}L_{\pi(2)}, \ldots, U_{\pi(m)}L_{\pi(m)}$$

where $\pi(k)$ denotes the distinct machine visited in the kth order within this cycle. There are $(m-1)!$ pure cycles in the structure defined above. Within this set, in order to minimize the cycle time, we shall focus on those for which the robot travel time is minimized. Note that in each pure cycle having the prescribed sequencing structure, the robot has to travel at least $2\delta\sum_{k=1}^{m}\min\{k, m+1-k\} = \lceil m(m+2)/2\rceil\delta$ time for the execution of m loading and unloading activities. Moreover, since every one of the machines and the *I/O*-station have to be visited in some sequence, the robot has to travel at least another $(m+1)\delta$ time. In other words, the lower bound for the robot travel time is

$$(\lceil m(m+2)/2\rceil + m + 1)\delta. \quad (2)$$

This line of thought brings us to the following two particular pure cycles which have robot travel times as low as the lower bound stated in (2).

Definition 3. C_2^m is the robot move cycle in an m-machine robotic cell with the following activity sequence: $L_1U_mL_mU_{m-1}L_{m-1}, \ldots, U_2L_2U_1$.

Definition 4. C_3^m is the robot move cycle in an m-machine robotic cell with the following activity sequence: $L_1U_2L_2U_3L_3U_4L_4, \ldots, U_{m-1}L_{m-1}U_mL_mU_1$.

The initial states of the cell are identical for both of C_2^m and C_3^m. All of the machines except machine 1 are loaded with a part and machine 1 is empty. The robot is in front of the *I/O*-station and it is idle. The first activity is identical for both of the cycles C_2^m and C_3^m and it is L_1. After L_1, the robot is in front of machine 1 for both cycles. At this point, the robot starts to move in opposite directions in the two cycles. However, the individual moves thereafter are mirror images of each other and result in exactly the same robot move times. The following lemma states this more formally:

Lemma 1. *For a given fixed processing time P, the cycle times of C_2^m and C_3^m are identical and represented as follows*: $T_{C_2^m} = T_{C_3^m} = 4m\varepsilon + (\lceil m(m+2)/2\rceil + m + 1)\delta + \max\{0, P-(4m-4)\varepsilon-(\lceil m(m+2)/2\rceil + m + 1 - 2\lceil m/2\rceil)\delta\}$.

Proof. Assume the starting time of the initial state is time 0 and let t_l be the completion time of activity $l \in \mathcal{A}$. At time t_{U_i}, the robot is at *I/O* station and at time t_{L_i}, the robot is at machine i. The cycle times of C_2^m and C_3^m are calculated as follows:

C_2^m	C_3^m
$t_{L_1} = 2\varepsilon + \delta$,	$t_{L_1} = 2\varepsilon + \delta$,
$t_{U_m} = t_{L_1} + 2\varepsilon + 3\delta + w_m$,	for $i = 2,3,\ldots,m-1$
$t_{L_m} = t_{U_m} + 2\varepsilon + \delta$,	$t_{U_i} = t_{L_{i-1}} + 2\varepsilon + \delta$
for $i = m-1,\ldots,3,2$	$+\min\{i, m+1-i\}\delta + w_i$,
	$t_{L_i} = t_{U_i} + 2\varepsilon + \min\{i, m+1-i\}\delta$,
$t_{U_i} = t_{L_{i+1}} + 2\varepsilon + \delta$	$t_{U_m} = t_{L_{m-1}} + 2\varepsilon + 2\delta + w_m$,
$+\min\{i, m+1-i\}\delta + w_i$,	
$t_{L_i} = t_{U_i} + 2\varepsilon + \min\{i, m+1-i\}\delta$,	$t_{L_m} = t_{U_m} + 2\varepsilon + \delta$,
$t_{U_1} = t_{L_2} + 2\varepsilon + 2\delta + w_1$.	$t_{U_1} = t_{L_m} + 2\varepsilon + 3\delta + w_1$.

After the last activity in both C_2^m and C_3^m, which is U_1, the robot is in front of the *I/O*-station as in the initial state of both cycles. t_{U_1} gives the cycle time in both of the proposed cycles and it is calculated as follows:

$$4m\varepsilon + (\lceil m(m+2)/2\rceil + m + 1)\delta + w_1 + w_2 + \cdots + w_m. \quad (3)$$

The waiting time for machine i is defined as $w_i = \max\{0, P-v_i\}$ and depends on v_i which is defined as the amount of time between just after loading machine i and the time the robot returns back in front of machine i to unload it. The time between two consecutive loadings of machine i gives the cycle time. In order to calculate v_i, first we calculate the complement of v_i for cycle time which is the time between just starting to unload the machine i and just after loading machine i. This time is calculated as follows:

The robot waits to unload machine $i(w_i)$, unloads machine $i(\varepsilon)$, travels to (I/O) station $(\min\{i, m+1-i\}\delta)$, drops part to (I/O) station (ε), takes a part (ε), travels to machine $i(\min\{i, m+1-i\}\delta + w_i))$, and loads machine $i(\varepsilon)$. In total this makes: $4\varepsilon + 2\min\{i, m+1-i\}\delta + w_i$.

Since the total of v_i and its complement gives the cycle time, we calculate the value of v_i by subtracting the complement from $T_{C_2^m}$. The v_i's are calculated for both of the cycles C_2^m and C_3^m, and found to be the same for these two cycles as:

$$v_i = T_{C_2^m} - (4\varepsilon + 2\min\{i, m+1-i\}\delta + w_i) = (4m-4)\varepsilon + (\lceil m(m+2)/2\rceil + m + 1 - 2\min\{i, m+1-i\})\delta + w_1 + w_2 + \cdots + w_m - w_i, \quad \forall i.$$

First, we prove that feasible solutions of v_i and w_i are the same for C_2^m and C_3^m for the corresponding machines. For C_2^m, in order to find the feasible solutions of v_i and w_i, the system of $2m$ equations which are presented as follows should be solved:

$$w_i = \max\{0, P - v_i\}, \quad \forall i,$$

$$v_i = (4m-4)\varepsilon + (\lceil m(m+2)/2 \rceil + m + 1 - 2\min\{i, m+1-i\})\delta + w_1 + w_2 + \cdots + w_m - w_i, \quad \forall i.$$

Similarly, the system has to be solved for C_3^m and the same feasible solution set arises for C_3^m.

For the same feasible v_i and w_i values the cycle times of C_2^m and C_3^m are the same and calculated by using Eq. (3) as follows:

$$T_{C_2^m} = 4m\varepsilon + (\lceil m(m+2)/2 \rceil + m + 1)\delta + w_1 + w_2 + \cdots + w_m = T_{C_3^m}. \tag{4}$$

In order to find the cycle time, we only have to find the total waiting time $\sum_i w_i$ in Eq. (4). In particular,

1. If $P \le v_i, \forall i$, then $w_1 + w_2 + \cdots + w_m = 0$.
2. Else if $\exists k \in [1, \ldots, m]$ such that $v_k < P$, then $w_k = P - v_k = P - (4m-4)\varepsilon - (\lceil m(m+2)/2 \rceil + m + 1 - 2\min\{k, m+1-k\})\delta - \sum_{i \ne k} w_i$. Hence, $w_1 + w_2 + \cdots + w_m = P - (4m-4)\varepsilon - (\lceil m(m+2)/2 \rceil + m + 1 - 2\min\{k, m+1-k\})\delta$.

Now we can conclude that: $T_{C_2^m} = T_{C_3^m} = 4m\varepsilon + (\lceil m(m+2)/2 \rceil + m + 1)\delta + \max\{0, P - (4m-4)\varepsilon - (\lceil m(m+2)/2 \rceil + m + 1 - 2\min\{k, m+1-k\})\delta; \forall k \in [1, \ldots, m]\}$.

Since $\min\{k, m+1-k\}$ takes its maximum value when $k = \lceil m/2 \rceil$, the equation becomes: $T_{C_2^m} = T_{C_3^m} = 4m\varepsilon + (\lceil m(m+2)/2 \rceil + m + 1)\delta + \max\{0, P - (4m-4)\varepsilon - (\lceil m(m+2)/2 \rceil + m + 1 - 2\lceil m/2 \rceil)\delta\}$. □

With the next theorem, we establish the cycle time lower bound of pure cycles for the robot centered cells.

Theorem 1. *For an m-machine robot centered cell, the cycle time of any pure cycle is no less than*

$$\underline{T_{I/O}} = \max\{4m\varepsilon + \lceil m(m+2)/2 \rceil \delta, 4\varepsilon + 2\lceil m/2 \rceil \delta + P\}. \tag{5}$$

Proof. A lower bound for pure cycles can be calculated by using two different definitions of the cycle time. The first lower bound is obtained from the exact robot activity duration and the second one is obtained from the given processing time vector. Since the robot has to perform a given set of robot activities, the total time required for these activities constitutes a lower bound. Thus, the first lower bound is obtained as follows: the set of robot activities can be analyzed in two groups and the first group is robot loading/unloading times. First, a part is taken from the *I/O*-station (ε), then loaded to one of the machines (ε), after the processing on the machine is finished, the part is unloaded (ε) and dropped to the *I/O*-station (ε). This makes a total of $4m\varepsilon$ for a repetition of cycle. The robot travel times constitute the second group of robot activities. The robot takes a part from *I/O*-station and travels to machine i to load it ($\min\{i, m+1-i\}\delta$), after the processing on the part is finished, the robot unloads the machine and travels to the *I/O*-station to drop the finished part ($\min\{i, m+1-i\}\delta$).

1. Suppose the number of machines is even, then the total robot travel time is calculated as:

$$\sum_{i=1}^{m} 2\min\{i, m+1-i\}\delta = 2\delta + 4\delta + 6\delta + \cdots + m\delta + m\delta + (m-2)\delta + (m-4)\delta + \cdots + 2\delta = \lceil m(m+2)/2 \rceil \delta.$$

2. Suppose the number of machines is odd, then the total robot travel time is calculated as:

$$\sum_{i=1}^{m} \min\{i, m+1-i\}\delta = 2\delta + 4\delta + 6\delta + \cdots + (m+1)\delta + (m-1)\delta + (m-3)\delta + \cdots + 2\delta = \lceil m(m+2)/2 \rceil \delta.$$

Consequently, the total of robot activities requires at least $4m\varepsilon + \lceil m(m+2)/2 \rceil \delta$ time units.

The second definition of a cycle time that leads to another lower bound is the minimum time between two consecutive loadings of any machine. The minimum time needed to unload machine i after loading it is P time units. After processing of the part is finished, the part is unloaded (ε), it is transferred to *I/O*-station ($\min\{i, m+1-i\}\delta$), and dropped (ε). After that, the robot takes a new part from *I/O*-station to make the consecutive loading of machine i (ε), brings the new part to machine i ($\min\{i, m+1-i\}\delta$), and finally loads it (ε). The total time between two consecutive loadings of machine i is at least $4\varepsilon + 2\min\{i, m+1-i\}\delta + P$. However, there are m machines and the total time for consecutive loadings are different for each of them. Thus, the cycle time has to be greater than or equal to the minimum time required between two consecutive loadings of any machine in the cell. So, the second lower bound of the cycle time is $4\varepsilon + 2\max\{\min\{i, m+1-i\}, i : 1, \ldots, m\}\delta + P$. □

The next theorem determines the processing time region where either C_2^m or C_3^m results in the minimum cycle time which is the cycle time lower bound for pure cycles in that region.

Theorem 2. *For an m-machine robot centered cell, either C_2^m or C_3^m dominates the rest of the pure cycles when*:

$$(4m-4)\varepsilon + (\lceil m(m+2)/2 \rceil + m + 1 - 2\lceil m/2 \rceil)\delta \le P.$$

Proof. Using the results of Lemma 1 and Theorem 1 for this region, we have

$$T_{C_2^m} = T_{C_3^m} = 4\varepsilon + 2\lceil m/2 \rceil \delta + P = \underline{T_{I/O}}. \qquad \square$$

The next lemma establishes the worst case performances of the two cycles for the remaining processing time region. The worst case performance is calculated by comparing the cycle time obtained from C_2^m and C_3^m to the cycle time lower bound. Let T^* represent the minimum cycle time attainable within the specified region.

Lemma 2. *When* $P < (4m-4)\varepsilon + (\lceil m(m+2)/2 \rceil + m + 1 - 2\lceil m/2 \rceil)\delta$ *we have*

$$T_{C_2^m} = T_{C_3^m} \le \left(1 + \frac{(m+1)\delta}{4m\varepsilon + \lceil m(m+2)/2 \rceil \delta}\right) \cdot T^*.$$

Proof. In the mentioned processing time region, the cycle time lower bound is calculated by using Theorem 1 as $4m\varepsilon + \lceil m(m+2)/2 \rceil \delta \le \underline{T_{I/O}}$. Then,

$$\frac{T_{C_2^m}}{T^*} = \frac{T_{C_3^m}}{T^*} \le \frac{T_{C_3^m}}{\underline{T_{I/O}}} \le \frac{4m\varepsilon + (\lceil m(m+2)/2 \rceil + m + 1)\delta}{4m\varepsilon + \lceil m(m+2)/2 \rceil \delta} = 1 + \frac{(m+1)\delta}{4m\varepsilon + \lceil m(m+2)/2 \rceil \delta}.$$

□

Since $(m+1)\delta / 4m\varepsilon + \lceil m(m+2)/2 \rceil \delta$ is a decreasing function of m, the difference between cycle time lower bound and the cycle time of either C_2^m or C_3^m decreases as the number of machines increases.

3.1. Detailed analysis of 3-machine case

In order to give some managerial insight, we analyze the 3-machine robot centered cell in more detail. There are 120 possible pure cycles in a 3-machine cell. The robot move sequences and cycle times of a selected sample of these pure cycles including the collection of best pure cycles (as will be formally shown later) are given in Table 1.

Fig. 2 plots the respective cycle times of a subset of these cycles against the processing time. The graph for the cycle time lower bound of $\underline{T}_{I/O}$ for 3-machine cells is also provided by dashed lines. The bold lines in the graph represent the minimum cycle times for the corresponding processing times. The graph clearly highlights the effectiveness of some of these pure cycles. In particular, pure cycles C_2^3, C_3^3, C_4^3, C_7^3, C_{15}^3, and C_{19}^3 stand out as nondominated ones for a range of processing time values. With cycles C_2^3 and C_3^3, the waiting times are minimized and hence these cycles are favorable for higher processing time values. In contrast, cycles C_4^3 and C_{19}^3 have the minimum total robot travel times and are favored for lower P values. In between these two extremes are the cycles C_7^3 and C_{15}^3 which try to balance the robot travel times and the waiting times.

The following sequence of lemmas will lead to Theorem 3 which will formalize our dominance results.

Lemma 3. *A pure cycle which has two consecutive load activities is never uniquely optimum.*

Table 1
A sample of pure cycles and their corresponding cycle times.

Cycle	Robot move sequence	Cycle time ($T_{C_i^3}$)
C_1^3	$L_1L_3U\ _2L_2U_1U_3$	$12\varepsilon+12\delta+\max\{0,P-8\delta-6\varepsilon\}$
C_2^3	$L_1U_3L_3\ U_2L_2U_1$	$12\varepsilon+12\delta+\max\{0,P-8\delta-8\varepsilon\}$
C_3^3	$L_1U_2L_2U_3\ L_3U_1$	$12\varepsilon+12\delta+\max\{0,P-8\delta-8\varepsilon\}$
C_4^3	$L_1U_1\ L_2U_2\ L_3U_3$	$12\varepsilon+8\delta+3P$
C_5^3	$L_1\ U_1L_2\ U_2U_3\ L_3$	$12\varepsilon+10\delta+2P$
C_6^3	$L_1U_1\ L_2L_3\ U_2U_3$	$12\varepsilon+12\delta+P+\max\{0,P-4\delta-2\varepsilon\}$
C_7^3	$L_1U_2\ L_2U_1\ L_3U_3$	$12\varepsilon+10\delta+P+\max\{0,P-4\varepsilon-6\delta\}$
C_8^3	$L_1L_2\ L_3U_1\ U_2U_3$	$12\varepsilon+16\delta+\max\{0,P-8\delta-4\varepsilon\}$
C_9^3	$L_1L_3\ U_3L_2\ U_2U_1$	$12\varepsilon+10\delta+2P$
C_{10}^3	$L_1L_3\ U_3U_1\ L_2U_2$	$12\varepsilon+10\delta+2P$
C_{11}^3	$L_1U_2\ L_3U_3\ L_2U_1$	$12\varepsilon+10\delta+P+\max\{0,P-4\varepsilon-4\delta\}$
C_{12}^3	$L_1U_1\ L_2U_3\ L_3U_2$	$12\varepsilon+10\delta+P+\max\{0,P-4\varepsilon-4\delta\}$
C_{13}^3	$L_1L_2\ U_2L_3\ U_3U_1$	$12\varepsilon+10\delta+2P$
C_{14}^3	$L_1L_2\ U_2U_1\ L_3U_3$	$12\varepsilon+10\delta+2P$
C_{15}^3	$L_1U_1\ L_3U_2\ L_2U_3$	$12\varepsilon+10\delta+P+\max\{0,P-4\varepsilon-6\delta\}$
C_{16}^3	$L_1U_1\ L_3L_2\ U_2U_3$	$12\varepsilon+10\delta+2P$
C_{17}^3	$L_1U_1\ U_3L_2\ U_2L_3$	$12\varepsilon+10\delta+2P$
C_{18}^3	$L_1U_1\ U_3L_3\ L_2U_2$	$12\varepsilon+10\delta+2P$
C_{19}^3	$L_1U_1\ L_3U_3\ L_2U_2$	$12\varepsilon+8\delta+3P$
C_{20}^3	$L_1U_3\ L_2U_2\ L_3U_1$	$12\varepsilon+12\delta+P+\max\{0,P-6\delta-4\varepsilon\}$
C_{21}^3	$L_1U_3\ L_3U_1\ L_2U_2$	$12\varepsilon+12\delta+P+\max\{0,P-6\delta-4\varepsilon\}$

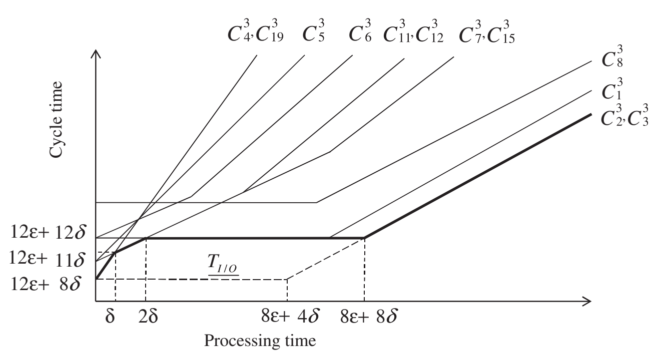

Fig. 2. 3-Machine cell analysis.

Proof. A list of all pure cycles of the stated form and their cycle times or lower bounds on their cycle times are tabulated in Table 4 in the Appendix. It can be seen that either C_2^3 (C_3^3) or C_7^3 (C_{15}^3) has a cycle time no worse than the bounds given in this table. □

Lemma 4. *A pure cycle of the form $U_iL_j\ U_kL_i\ U_jL_k$ where i,j, and k are distinct elements from set* {1, 2, 3} *is never uniquely optimum.*

Proof. It can be easily verified that the cycle time of a pure cycle in the stated form is at least $12\varepsilon+12\delta+\max\{0,P-(4\varepsilon+4\delta)\}$ which is dominated by the cycle time of C_2^3 (C_3^3). □

In light of the previous two lemmas, it is possible to eliminate all pure cycles but those of the following three forms, namely, $U_iL_i\ U_jL_j\ U_kL_k$ (i.e., C_2^3 and C_3^3), $L_iU_i\ L_jU_j\ L_kU_k$ (i.e., C_4^3 and C_{19}^3), and $U_iL_i\ U_jL_k\ U_kL_j$ (i.e.,C_7^3, C_{11}^3, C_{12}^3, C_{15}^3, C_{20}^3, and C_{21}^3) where i,j, and k are distinct elements from {1,2,3}. Moreover, cycles C_7^3 and C_{15}^3 have the same cycle time and dominate the four cycles C_{11}^3, C_{12}^3, C_{20}^3, and C_{21}^3 which share a similar form. Ultimately, in a 3-machine cell, there are six cycles, namely, C_2^3, C_3^3, C_4^3, C_7^3, C_{15}^3, and C_{19}^3 that are potentially optimal and the following theorem identifies the regions of optimality for these cycles.

Theorem 3. *For a* 3*-machine robot centered cell*:

1. *If $P\le\delta$, then C_4^3 (or C_{19}^3) has the minimum cycle time.*
2. *If $\delta\le P\le 2\delta$, then C_7^3 (or C_{15}^3) has the minimum cycle time.*
3. *If $P\ge 2\delta$, then C_2^3 (or C_3^3) has the minimum cycle time.*

Proof. The proof follows from a simple comparison of the three distinct cycles times, namely, $12\varepsilon+12\delta+\max\{0,P-8\delta-8\varepsilon\}$, $12\varepsilon+8\delta+3P$, and $12\varepsilon+10\delta+P+\max\{0,P-4\varepsilon-6\delta\}$. □

4. Bicriteria analysis of C_2^m and C_3^m

Up to now, we have focused solely on the cycle time objective and restricted our attention to the 3-machine case. We now analyze the m-machine case when the processing times are assumed to be controllable with the bicriteria viewpoint of minimizing the cycle time and the total manufacturing cost simultaneously. As shown in the previous section, the pure cycles C_2^m and C_3^m are quite effective in minimizing the cycle time. We propose that these two prominent cycles are also efficient pure cycles in terms of both objectives.

4.1. Problem definition

Let P_i denote the processing time on machine i, which is now to be considered as a decision variable. A feasible processing time value on any machine is bounded from above by an upper bound P^U which is the same for every machine, i.e. $0\le P_i\le P^U$. We let $\boldsymbol{P}=(P_1,P_2,\ldots,P_m)$ denote a processing time vector. We present the set of feasible processing time vectors as $\mathcal{P}_{feas}=\{(\mathcal{P}_1,\mathcal{P}_2,\ldots,\mathcal{P}_m)\in\mathcal{R}^m:0\le\mathcal{P}_i\le\mathcal{P}^U\forall\rangle\}$. We further need the following definitions:

$f(P_i)$:	The manufacturing cost incurred on machine i which is monotonically decreasing for $0\le P_i\le P^U$, $\forall i$.
$F_1(C_i^m,\boldsymbol{P})=\sum_{i=1}^m f(P_i)$:	Total manufacturing cost depending only on the processing times.
$F_2(C_i^m,\boldsymbol{P})$:	Cycle time corresponding to processing time vector $\boldsymbol{P}$ and the pure cycle C_i^m.

The manufacturing cost is the sum of machining and tooling costs for manufacturing operations. As the processing time decreases, the machining cost decreases, but the tool life decreases as well and ultimately the tooling cost increases. We have defined the upper bound P^U as the processing time value that minimizes the manufacturing cost function for each part without considering its impact on the cycle time objective. Since cycle time is a regular scheduling measure, increasing the processing time of any part beyond P^U will not improve the cycle time value. Consequently, any processing time value greater than P^U will lead to an inferior solution because both objectives will get worse. We thus assume that the manufacturing cost function is a monotonically decreasing and strictly convex function of the processing time. In the bicriteria optimization problem under consideration, the total manufacturing cost incurred throughout a cycle depends only on the processing times. On the other hand, the cycle time depends on both the robot move cycle and the selected processing times. A feasible solution to our problem is composed of a feasible robot move sequence and a feasible processing time vector. Since our study considers only the pure cycles, the set of feasible cycles in an m-machine cell is the set of pure cycles in that cell, i.e. $i \in [1,\ldots,(2m-1)!]$. The bicriteria optimization problem at hand is the following:

$$\begin{aligned}&\text{minimize} \quad F_1(C_i^m,\boldsymbol{P})\\&\text{minimize} \quad F_2(C_i^m,\boldsymbol{P})\\&\text{Subject to} \quad \boldsymbol{P}\in\mathcal{P}_{feas}.\end{aligned}$$

In our study, we used the posteriori optimization method since the considered two objectives are equally important. In this method, all of the nondominated solutions are found by minimizing the nondecreasing composite function $F(f,g)$ where f stands for the total manufacturing cost and g stands for the cycle time. We use the epsilon-constraint method denoted by $\varepsilon(f|g)$ to find the nondominated points that minimize f for a given upper bound of g as discussed in T'Kindt and Billaut [14]. So, for each pure cycle, the following ECP is solved to find the nondominated processing time vector for a given cycle time level K:

$$(\mathbf{ECP})\quad \begin{aligned}&\text{minimize} \quad F_1(C_i^m,\boldsymbol{P})\\&\text{Subject to} \quad F_2(C_i^m,\boldsymbol{P})\le K\\&\boldsymbol{P}\in\mathcal{P}_{feas}.\end{aligned}$$

The following definitions will be utilized when comparing cycles:

Definition 5. For a robot move sequence C_i^m and a given cycle time level K, the set of nondominated points is defined as $\mathbf{P}^*(C_i^m|K)=\{\boldsymbol{P}\in\mathcal{P}_{feas}:$ There is no other $\mathbf{P}'\in\mathcal{P}_{feas}$ such that $F_1(C_i^m,\mathbf{P}')<F_1(C_i^m,\boldsymbol{P})$ where $F_2(C_i^m,\boldsymbol{P})\le K$ and $F_2(C_i^m,\mathbf{P}')\le K\}$.

For a given cycle time level, in order to decide which pure cycle dominates another, we compare the incurred manufacturing cost values. More formally:

Definition 6. We say that a cycle C_i^m dominates another cycle C_j^m for a given cycle time level K, if there is no $\hat{\mathbf{P}}\in\mathbf{P}^*(C_j^m|K)$ such that $F_1(C_j^m,\hat{\mathbf{P}})<F_1(C_i^m,\tilde{\mathbf{P}})$ for all $\tilde{\mathbf{P}}\in\mathbf{P}^*(C_i^m|K)$, where $F_2(C_j^m,\hat{\mathbf{P}})\le K$ and $F_2(C_i^m,\tilde{\mathbf{P}})\le K$.

4.2. Solution procedure

In this section, we first determine the cycle time of the proposed pure cycles C_2^m and C_3^m when a processing time vector is given. Afterwards, we determine the nondominated points of C_2^m and C_3^m. Finally, the cycle time region where either C_2^m or C_3^m dominates the rest of the pure cycles is determined by comparing the total manufacturing cost obtained from the nondominated solutions of C_2^m and C_3^m with the lower bound of the total manufacturing cost. With the next lemma, we can determine the cycle time of either C_2^m or C_3^m when there is a given processing time vector.

Lemma 5. *The cycle time of C_2^m (and C_3^m) for a given processing time vector $\mathbf{P}=(P_1,\ldots,P_m)$ is:* $T_{C_2^m}=T_{C_3^m}=4m\varepsilon+(\lceil m(m+2)/2\rceil+m+1)\delta+\max\{0,\max\{P_i-(4m-4)\varepsilon-(\lceil m(m+2)/2\rceil+m+1-2\min\{i,m+1-i\})\delta, i:1,\ldots,m\}\}$.

Proof. The cycle times of C_2^m and C_3^m are calculated in Eq. (3) as $4m\varepsilon+(\lceil m(m+2)/2\rceil+m+1)\delta+w_1+w_2+\cdots+w_m$. The waiting time on machine i is defined as $w_i=\max\{0,P_i-v_i\}$. The values of v_i's are determined in the proof of Lemma 1 as $v_i=(4m-4)\varepsilon+(\lceil m(m+2)/2\rceil+m+1-2\min\{i,m+1-i\})\delta+w_1+w_2+\cdots+w_m-w_i$ for all machines.

There are two different cases for a total waiting time and the sufficient conditions for these cases are determined as follows:

1. If $P_i\le v_i$ for $\forall i\in[1,\ldots,m]$, then $w_i=0$, for $i=1,\ldots,m$.
2. Else if $\exists k\in[1,\ldots,m]$ such that $v_k<P_k$, then $w_k=P_k-v_k=P_k-(4m-4)\varepsilon-(\lceil m(m+2)/2\rceil+m+1-2\min\{k,m+1-k\})\delta-\sum_{i\ne k}w_i$.
   Hence,
   $w_1+w_2+\ldots+w_m=P_k-(4m-4)\varepsilon-(\lceil m(m+2)/2\rceil+m+1-2\min\{k,m+1-k\})\delta$.

So, $w_1+w_2+\cdots+w_m=\max\{0,\max\{P_k-(4m-4)\varepsilon-(\lceil m(m+2)/2\rceil+m+1-2\min\{k,m+1-k\})\delta\}$ and the cycle time is obtained by replacing the total of waiting time in Eq. (3) with this max function. $\square$

With the next theorem, the cycle time lower bound for pure cycles in robot centered cells for a given processing time vector is derived.

Theorem 4. *For an m-machine robot centered cell with controllable processing times, the cycle time of any pure cycle is no less than*: $\underline{T_L}=\max\{4m\varepsilon+\lceil m(m+2)/2\rceil\delta, 4\varepsilon+2\max\{\min\{i,m+1-i\}\delta+P_i, i:1,\ldots,m\}\}$.

Proof. From the definition of pure cycles, it is apparent that the cycle time of a pure cycle is bounded from below by two lower bounds. The first lower bound is obtained from the exact robot activity time that is composed of loading/unloading and part transportation times. In Theorem 1, this lower bound is calculated for fixed processing times. Since, loading/unloading and part transportation times do not depend on processing times, this lower bound remains the same for controllable processing times as $4m\varepsilon+\lceil m(m+2)/2\rceil\delta$.

The second lower bound is the minimum time required between two consecutive loadings of any machine. In Theorem 1, for machine i, this lower bound is calculated as $4\varepsilon+2\min\{i,m+1-i\}\delta+P$ for a fixed processing time P. Now we consider controllable processing times, thus the minimum time required between two consecutive loadings of machine i is calculated as $4\varepsilon+2\min\{i,m+1-i\}\delta+P_i$. However, there are m machines and the total time for consecutive loadings are different from each other. Since the cycle time is at least equal to the total time for consecutive loadings of any machine in the cell, the second lower bound is $4\varepsilon+\max_i\{2\min\{i,m+1-i\}\delta+P_i\}$. $\square$

With the next lemma, for a given cycle time level K, the individual upper bounds of processing times of pure cycles is determined. Let $\overline{\mathbf{P}}(K)=(\overline{P}_1(K),\ldots,\overline{P}_m(K))$ be the vector of individual upper bounds. Since increasing the processing times decreases the corresponding manufacturing costs, our aim is to find the maximum processing time for each machine within the feasible boundaries.

Lemma 6. *For a given cycle time level K, the vector of upper bounds of processing times in robot centered cells for pure cycles is*:

$\overline{\mathbf{P}}(K)=(\overline{P}_1(K),\ldots,\overline{P}_m(K))$, *where* $\overline{P}_i(K)=\min\{P^U,K-(4\varepsilon+2\min\{i,m+1-i\}\delta)\},\forall i$.

Proof. The two bounds constraining the processing times are the following:

1. The processing times must be less than or equal to P^U which leads to $\overline{P}_i(K)\le P^U$, $\forall i$.
2. In addition, the processing times on machines cannot exceed a specific value, otherwise the cycle time K will be exceeded. Using the results of Theorem 4, we must have:
   $\underline{T_L}=\max\{4m\varepsilon+\lceil m(m+2)/2\rceil\delta,4\varepsilon+\max\{2\min\{i,m+1-i\}\delta +P_i,i:1,\ldots,m\}\}\le K$.
   In particular, $\max\{2\min\{i,m+1-i\}\delta+P_i,i:1,\ldots,m\}\le K-4\varepsilon$, and therefore $P_i\le K-4\varepsilon-2\min\{i,m+1-i\}\delta$, $\forall i$. This implies that $\overline{P}_i(K)\le K-(4\varepsilon+2\min\{i,m+1-i\}\delta),\forall i$. □

Let us now deviate from the cycle time analysis towards the analysis of the effect of controllable processing times on minimizing the total manufacturing cost. Evidently, the total manufacturing cost might be decreased by using controllable processing times the reason simply being that we can increase the processing times without exceeding the cycle time limit. The cycle times of C_2^m and C_3^m are equal as shown in Lemma 5, thus these two cycles result in the same set of nondominated processing time vectors, i.e., $\mathbf{P}^*(C_2^m|K)=\mathbf{P}^*(C_3^m|K)$. In the next lemma, the processing time vectors that give the minimum total manufacturing cost obtained from either C_2^m or C_3^m for a given cycle time level K are determined.

Lemma 7. *Given any feasible cycle time level K, the nondominated processing time vector of* C_2^m (*or* C_3^m) *is defined as* $(P_1^*,P_2^*,\ldots,P_m^*)\in\mathbf{P}^*(C_2^m|K)=\mathbf{P}^*(C_3^m|K)$ *where* $P_i^*=\min\{P^U,K-(4\varepsilon+2\min\{i,m+1-i\}\delta)\}$, $\forall i$.

Proof. For a given cycle time level K, a feasible processing time vector is composed of processing times on machines that satisfy two upper bounds.

1. All processing times must be at most P^U.
2. In addition, the processing times, P_i's, are bounded so as not to exceed the cycle time level K. By fixing the cycle time to K in Lemma 5, we have: $K=4m\varepsilon+(\lceil m(m+2)/2\rceil+m+1)\delta+\max\{0,\max\{P_i-(4m-4)\varepsilon-(\lceil m(m+2)/2\rceil+m+1-2\min\{i,m+1-i\})\delta, i:1,\ldots,m\}\}$.
   This leads to $P_i\le K-(4\varepsilon+2\min\{i,m+1-i\}\delta)$.

The possible largest processing times without violating the bounds found in the first and the second arguments above compose the nondominated processing time vectors in Lemma 7. □

The numerical example below will be useful in order to see an application of Lemma 7.

Example 1. Consider a 5-machine robot centered cell. Let $\delta=0.1$, $\varepsilon=0.1$, $P^U=4.5$ and $K=5.0$. For this cycle time level, the nondominated processing time vector $(P_1^*,P_2^*,P_3^*,P_4^*,P_5^*)\in\mathbf{P}^*(C_2^5|5.0)=\mathbf{P}^*(C_3^5|5.0)$ is calculated using Lemma 7 as follows:

$$\begin{bmatrix}P_1^*\\P_2^*\\P_3^*\\P_4^*\\P_5^*\end{bmatrix}=\begin{bmatrix}\min\{P^U,K-(4\varepsilon+2\delta)\}\\\min\{P^U,K-(4\varepsilon+4\delta)\}\\\min\{P^U,K-(4\varepsilon+6\delta)\}\\\min\{P^U,K-(4\varepsilon+4\delta)\}\\\min\{P^U,K-(4\varepsilon+2\delta)\}\end{bmatrix}=\begin{bmatrix}\min\{4.5,4.4\}\\\min\{4.5,4.2\}\\\min\{4.5,4.0\}\\\min\{4.5,4.2\}\\\min\{4.5,4.4\}\end{bmatrix}=\begin{bmatrix}4.4\\4.2\\4.0\\4.2\\4.4\end{bmatrix}.$$

It is interesting to notice that although the parts are identical, the optimum processing times may be different for each machine.

The next theorem presents the cycle time region where either C_2^m or C_3^m dominates the rest of the pure cycles according to our bicriteria optimization problem. Any feasible cycle time K of C_2^m and C_3^m as determined by using Lemma 5 must satisfy $4m\varepsilon+(\lceil m(m+2)/2\rceil+m+1)\delta\le K$. This is exactly the minimum required time for loading and unloading and travel times for the robot even when the waiting times, w_i, or the processing times, P_i, are equal to zero. Therefore, we consider this region in the following theorem.

Theorem 5. *Whenever* C_2^m *or* C_3^m *is feasible, they dominate all other pure cycles in robot centered cells.*

Proof. Since $\mathbf{P}^*(C_2^m|K)=\mathbf{P}^*(C_3^m|K)=(P_1,P_2,\ldots,P_m)=\overline{\mathbf{P}}(K)$ where $P_i=\min\{P^U,K-(4\varepsilon+2\min\{i,m+1-i\}\delta)\},\forall i$, there is no other processing time vector with any component greater than that of the nondominated processing time vector obtained from C_2^m (or C_3^m). □

The following example depicts this strong result.

Example 2. Consider a 5-machine robot centered cell with the same parameters as in Example 1. In that example, the nondominated processing time vector of C_2^5 and C_3^5 is calculated as $\mathbf{P}^*(C_2^5|5.0)=\mathbf{P}^*(C_3^5|5.0)=(4.4,4.2,4.0,4.2,4.4)$. The upper bound of processing time vector for cycle time level $K=5.0$ is calculated from Lemma 6 as follows:

$$\overline{\mathbf{P}}(K)=\begin{bmatrix}\overline{P}_1(K)\\\overline{P}_2(K)\\\overline{P}_3(K)\\\overline{P}_4(K)\\\overline{P}_5(K)\end{bmatrix}=\begin{bmatrix}\min\{P^U,K-(4\varepsilon+2\delta)\}\\\min\{P^U,K-(4\varepsilon+4\delta)\}\\\min\{P^U,K-(4\varepsilon+6\delta)\}\\\min\{P^U,K-(4\varepsilon+4\delta)\}\\\min\{P^U,K-(4\varepsilon+2\delta)\}\end{bmatrix}=\begin{bmatrix}\min\{4.5,4.4\}\\\min\{4.5,4.2\}\\\min\{4.5,4.0\}\\\min\{4.5,4.2\}\\\min\{4.5,4.4\}\end{bmatrix}=\begin{bmatrix}4.4\\4.2\\4.0\\4.2\\4.4\end{bmatrix}.$$

Since the nondominated processing time vectors of C_2^5 and C_3^5 are equal to the upper bound of processing time vectors $\mathbf{P}^*(C_2^5|5.0)=\mathbf{P}^*(C_3^5|5.0)=\overline{\mathbf{P}}(K)$, there is no other pure cycle that can result in less total manufacturing cost than either C_2^5 or C_3^5.

Recently, Gultekin et al. [8] analyzed pure cycles with fixed processing times and Yildiz et al. [15] analyzed pure cycles with controllable processing times in m-machine in-line robotic cells. In this study, we consider the pure cycles in robot centered cells and propose new robot move sequences. With the next theorem, we compare the results of our study to Gultekin et al. [8] and prove that the pure cycles in robot centered cells dominate the pure cycles in in-line robotic cells.

Theorem 6. C_2^m (*or* C_3^m) *of robot centered cells dominates all pure cycles of in-line robotic cells.*

Proof. The cycle time lower bound for pure cycles in in-line robotic cells is derived by Gultekin et al. [8] as $4m\varepsilon+2m(m+1)\delta$. For this region, the processing time vector resulting in the lower bound of total manufacturing cost for in-line robotic cells can be found as $\overline{\mathbf{P}}_{inline}(K)=(P_1(K),\ldots,P_m(K))$, where $P_i(K)=\min\{P^U, K-(4\varepsilon+(2m+2)\delta)\}$, $\forall i$.

Since $4m\varepsilon+(\lceil m(m+2)/2\rceil+m+1)\delta<4m\varepsilon+2m(m+1)\delta$, from Lemma 5, we know that the proposed C_2^m and C_3^m cycles are feasible in this region. In addition, by using Lemma 7, we find the optimum processing time vector obtained from either C_2^m or C_3^m for robot centered cells as follows: $(P_1^*,P_2^*,\ldots,P_m^*)\in(\mathbf{P}^*(C_2^m|K)=\mathbf{P}^*(C_3^m|K))$ where $P_i^*=\min\{P^U,K-(4\varepsilon+2\min\{i,m+1-i\}\delta)\},\forall i$.

When we compare these two processing time vectors, we have

$$\overline{\mathbf{P}}_{inline}(K)=\begin{bmatrix} P_1(K)\\ P_2(K)\\ \vdots\\ P_i(K)\\ \vdots\\ P_{m-1}(K)\\ P_m(K)\end{bmatrix}=\begin{bmatrix}\min\{P^U,K-(4\varepsilon+(2m+2)\delta)\}\\ \min\{P^U,K-(4\varepsilon+(2m+2)\delta)\}\\ \vdots\\ \min\{P^U,K-(4\varepsilon+(2m+2)\delta)\}\\ \vdots\\ \min\{P^U,K-(4\varepsilon+(2m+2)\delta)\}\\ \min\{P^U,K-(4\varepsilon+(2m+2)\delta)\}\end{bmatrix}$$

$$\leq\begin{bmatrix}\min\{P^U,K-(4\varepsilon+2\delta)\}\\ \min\{P^U,K-(4\varepsilon+2\min\{2,m-1\}\delta)\}\\ \vdots\\ \min\{P^U,K-(4\varepsilon+2\min\{i,m+1-i\}\delta)\}\\ \vdots\\ \min\{P^U,K-(4\varepsilon+2\min\{m-1,2\}\delta)\}\\ \min\{P^U,K-(4\varepsilon+2\delta)\}\end{bmatrix}=\begin{bmatrix}P_1^*\\ P_2^*\\ \vdots\\ P_i^*\\ \vdots\\ P_{m-1}^*\\ P_m^*\end{bmatrix}$$

$$=\mathbf{P}^*(C_2^m|K)=\mathbf{P}^*(C_3^m|K).$$

Finally, it can be seen that the optimum processing time vector obtained from C_2^m and C_3^m in robot centered cell is greater than or equal to the processing time upper bound of in-line robotic cell for pure cycles, i.e., $\mathbf{P}^*(C_2^m|K)=\mathbf{P}^*(C_3^m|K)\geq\overline{\mathbf{P}}_{inline}(K)$. Thus, the cost obtained from $\mathbf{P}^*(C_2^m|K)$ and $\mathbf{P}^*(C_3^m|K)$ is less than the cost obtained from $\overline{\mathbf{P}}_{inline}(K)$. In other words, $F_1(C_2^m,\mathbf{P}^*(C_2^m|K))=F_1(C_3^m,\mathbf{P}^*(C_3^m|K))\leq F_1(C_i^m,\overline{\mathbf{P}}_{inline}(K))$. □

4.3. 3-Machine case with controllable processing times

In this section, we study the bicriteria optimization problem in the special case of 3-machine cells. The previous section has established the dominance of cycles C_2^3 (or C_3^3) whenever they are feasible. Thus, we only need to do our analysis for the region when the cycle time value K is strictly less than $12\varepsilon+12\delta$. In the sequel, we will consider all feasible pure cycles in this restricted cycle time region and for each provide its set of nondominated processing time vectors as defined in Definition 5.

The cycle time calculations and the derivation of nondominated points are depicted only for cycle C_7^3 which involves the most complicated analysis.

Lemma 8. *The cycle time of C_7^3 for a given processing time vector $\mathbf{P}=(P_1,P_2,P_3)$ is: $T_{C_7^3}=12\varepsilon+10\delta+\max\{P_3,P_1+P_3-4\varepsilon-6\delta, P_2-8\varepsilon-6\delta\}$.*

Proof. The robot move sequence of C_7^3 is $L_1U_2L_2U_1L_3U_3$. The cycle time is the sum of three quantities, namely, total robot move time, total robot load/unload, pick-up/drop time, and total waiting time. Initially the robot is in front of *I/O* buffer, takes a part (ε), moves to machine 1 (δ), loads machine 1 (ε), moves to machine 2 (δ), waits until the job is finished (w_2), unloads machine 2 (ε), moves to *I/O* buffer (2δ), drops the part (ε), takes a part (ε), moves to machine 2 (2δ), loads machine 2 (ε), moves to machine 1 (δ), waits until the job is finished (w_1), unloads machine 1 (ε), moves to *I/O* buffer (δ), drops the part (ε), takes a part (ε), moves to machine 3 (δ), loads machine 3 (ε), waits until the job is finished (P_3), unloads machine 3 (ε), moves to *I/O* buffer (δ), and drops the part (ε). The union of all these evaluates to: $T_{C_7^3}=12\varepsilon+10\delta+w_1+w_2+P_3$ with $w_1=\max\{0,P_1-v_1\}$ and $w_2=\max\{0,P_2-v_2\}$ and where v_i for $i=1,2$ is the amount of time between just after loading the machine i and the time the robot returns back to machine i to unload it.

We determine v_1 as follows: after loading machine 1, the robot moves to machine 2 (δ), waits until the job is finished (w_2), unloads the part (ε), moves to *I/O* buffer (2δ), drops the part (ε), takes a part (ε), moves to machine 2 (2δ), loads machine 2 (ε), and finally moves to machine 1 to unload it (δ). Thus, $v_1=4\varepsilon+6\delta+w_2$. Similarly, $v_2=8\varepsilon+6\delta+w_1+P_3$. In turn, $T_{C_7^3}=12\varepsilon+10\delta+\max\{0,P_1-4\varepsilon-6\delta-w_2\}+\max\{0,P_2-8\varepsilon-6\delta-w_1-P_3\}+P_3$.

There are four possible cases that may arise:

1. If $P_1\leq v_1$ and $P_2\leq v_2$ then $w_1=0$, $w_2=0$. Thus, $T_{C_7^3}=12\varepsilon+10\delta+P_3$.
2. If $P_1>v_1$ and $P_2\leq v_2$ then $w_1=P_1-4\varepsilon-6\delta$ and $w_2=0$. Thus, $T_{C_7^3}=12\varepsilon+10\delta+P_1+P_3-4\varepsilon-6\delta$.
3. If $P_1\leq v_1$ and $P_2>v_2$ then $w_1=0$ and $w_2=P_2-8\varepsilon-6\delta-P_3$. Thus, $T_{C_7^3}=12\varepsilon+10\delta+P_2-8\varepsilon-6\delta$.
4. If $P_1>v_1$ and $P_2>v_2$ then $w_1=P_1-4\varepsilon-6\delta-w_2$ and $w_2=P_2-8\varepsilon-6\delta-P_3-w_1$. Thus, $w_1+w_2=P_1-4\varepsilon-6\delta=P_2-8\varepsilon-6\delta-P_3$.

More compactly, $T_{C_7^3}=12\varepsilon+10\delta+\max\{P_3,P_1+P_3-4\varepsilon-6\delta,P_2-8\varepsilon-6\delta\}$. □

In this section, we shall assume for simplicity that the cycle time value K is small enough so that no processing time value hits its allowed upper bound of P^U. If this is not the case, P^U should appear as a bounding value in all the processing time derivations. The following lemma provides the nondominated processing time vector of C_7^3 under this nonrestrictive assumption.

Lemma 9. *For a given cycle time level K such that $12\varepsilon+10\delta\leq K\leq 16\varepsilon+16\delta$, the nondominated processing time vector of C_7^3 is $(P_1^*,P_2^*,P_3^*)=(4\varepsilon+6\delta,K-4\varepsilon-4\delta,K-12\varepsilon-10\delta)$.*

Proof. There are two upper bounds that bound the processing times:

1. All processing times must satisfy the upper bound, P^U, limitation. We assume for simplicity that this bound is not tight.
2. In addition, the processing times, P_i's, are jointly bounded so as not to exceed the cycle time level K. By fixing the cycle time to K in the previous lemma, we have

$$K=12\varepsilon+10\delta+\max\{P_3,P_1+P_3-4\varepsilon-6\delta,P_2-8\varepsilon-6\delta\}.$$

This leads to the following system of inequalities:

$$P_3\leq K-12\varepsilon-10\delta,$$
$$P_1\leq K-8\varepsilon-4\delta-P_3,$$
$$P_2\leq K-4\varepsilon-4\delta.$$

It can easily be verified that $(P_1^*,P_2^*,P_3^*)=(4\varepsilon+6\delta,K-4\varepsilon-4\delta,K-12\varepsilon-10\delta)$ is the unique vector satisfying the above system of inequalities tightly. Moreover, in the specified cycle time region of $12\varepsilon+10\delta\leq K\leq 16\varepsilon+16\delta$, $P_3^*\leq P_1^*\leq P_2^*$. Since both P_2^* and P_3^* are at their possible largest values, the only way to improve the cost is by increasing P_1^* value. However, the nonincreasing nature of the underlying cost function implies that it is not possible to decrease cost by increasing P_1^* value and correspondingly decreasing P_3^* value. □

Tables 2 and 3 enlist the results of the analysis done for C_7^3 above for all the 14 feasible cycles in the region of study. As can be observed in Table 3, sometimes, the nondominated point is not unique and only upper bounds can be attained for the processing times.

Since the manufacturing cost is machine independent, for each cycle C_i^3 we may assume without loss of generality that the nondominated processing time vector $\mathbf{P}^*(C_i^3|K)=(P_1^*,P_2^*,P_3^*)$ is permuted such that $P_1^*\leq P_2^*\leq P_3^*$. It can easily be verified that with this ordering of nondominated processing times:

$P^*(C_4^3|K) = P^*(C_{19}^3|K)$,

$P^*(C_5^3|K) = P^*(C_9^3|K) = P^*(C_{13}^3|K) = P^*(C_{18}^3|K)$,

$P^*(C_7^3|K) = P^*(C_{15}^3|K)$,

$P^*(C_{10}^3|K) = P^*(C_{17}^3|K)$,

$P^*(C_{11}^3|K) = P^*(C_{12}^3|K)$,

$P^*(C_{14}^3|K) = P^*(C_{16}^3|K)$.

With these equivalence relationships we may simplify our comparison of 14 cycles into just the comparison of the leftmost six cycles appearing above. Now, we are ready to present the results of the bicriteria optimization problem in the special case of 3-machine cells.

Let K_1 be the cycle time for which the total manufacturing costs of $\mathbf{P}^*(C_4^3|K_1)$ and $\mathbf{P}^*(C_7^3|K_1)$ coincide. More formally,

$$3f((K_1-12\varepsilon-8\delta)/3) = f(4\varepsilon+6\delta)+f(K_1-4\varepsilon-4\delta)+f(K_1-12\varepsilon-10\delta).$$

For C_7^3 to be feasible, $K_1 > 12\varepsilon+10\delta$ must hold. Moreover, if $K \geq 12\varepsilon+11\delta$, then $\mathbf{P}^*(C_4^3|K_1) \leq \mathbf{P}^*(C_7^3|K_1)$. Hence, $12\varepsilon+10\delta < K_1 < 12\varepsilon+11\delta$ must hold and the actual point value of K_1 will be determined by the manufacturing cost function.

Theorem 7. *For 3-machine robot centered cells,*

1. *If $K < K_1$, then either C_4^3 or C_{19}^3 dominates the rest of the pure cycles.*
2. *If $K_1 \leq K < 12\varepsilon+12\delta$, then either C_7^3 or C_{15}^3 dominates the rest of the pure cycles.*
3. *If $K \geq 12\varepsilon+12\delta$, then either C_2^3 or C_3^3 dominates the rest of the pure cycles.*

Proof.

- Case 3. In the region when the cycle time satisfies $K \geq 12\varepsilon+12\delta$, C_2^3 (or C_3^3) is feasible, and Theorem 5 establishes Case 3.

Table 2
The feasible pure cycles and their corresponding cycle times when $K < 12\varepsilon+12\delta$.

Cycle	Cycle time
C_4^3	$12\varepsilon+8\delta+P_1+P_2+P_3$
C_5^3	$12\varepsilon+10\delta+P_1+P_2+\max\{0,P_3-8\varepsilon-8\delta-P_1-P_2\}$
C_7^3	$12\varepsilon+10\delta+\max\{0,P_1-4\varepsilon-6\delta-w_2\}+\max\{0,P_2-8\varepsilon-6\delta-w_1-P_3\}+P_3$
C_9^3	$12\varepsilon+10\delta+\max\{0,P_1-8\varepsilon-8\delta-P_3-P_2\}+P_2+P_3$
C_{10}^3	$12\varepsilon+10\delta+\max\{0,P_1-4\varepsilon-4\delta-P_3\}+P_2+P_3$
C_{11}^3	$12\varepsilon+10\delta+\max\{0,P_1-8\varepsilon-8\delta-P_3-w_2\}+\max\{0,P_2-4\varepsilon-4\delta-w_1\}+P_3$
C_{12}^3	$12\varepsilon+10\delta+P_1+\max\{0,P_2-4\varepsilon-4\delta-w_3\}+\max\{0,P_3-8\varepsilon-8\delta-w_2-P_1\}$
C_{13}^3	$12\varepsilon+10\delta+\max\{0,P_1-8\varepsilon-8\delta-P_3-P_2\}+P_2+P_3$
C_{14}^3	$12\varepsilon+10\delta+\max\{0,P_1-4\varepsilon-6\delta-P_2\}+P_2+P_3$
C_{15}^3	$12\varepsilon+10\delta+P_1+\max\{0,P_2-8\varepsilon-6\delta-P_1-w_3\}+\max\{0,P_3-4\varepsilon-6\delta-w_2\}$
C_{16}^3	$12\varepsilon+10\delta+P_1+P_2+\max\{0,P_3-4\varepsilon-6\delta-P_2\}$
C_{17}^3	$12\varepsilon+10\delta+P_1+P_2+\max\{0,P_3-4\varepsilon-4\delta-P_1\}$
C_{18}^3	$12\varepsilon+10\delta+P_1+P_2+\max\{0,P_3-8\varepsilon-8\delta-P_1-P_2\}$
C_{19}^3	$12\varepsilon+8\delta+P_1+P_2+P_3$

Table 3
The nondominated processing times (or bounds) of feasible pure cycles when $K < 12\varepsilon+12\delta$.

Cycle	$\mathbf{P}^*(C_i^3\|K)$		
	Machine 1	Machine 2	Machine 3
C_4^3	$(K-12\varepsilon-8\delta)/3$	$(K-12\varepsilon-8\delta)/3$	$(K-12\varepsilon-8\delta)/3$
C_5^3	$(K-12\varepsilon-10\delta)/2$	$(K-12\varepsilon-10\delta)/2$	$K-4\varepsilon-2\delta$
C_7^3	$4\varepsilon+6\delta$	$K-4\varepsilon-4\delta$	$K-12\varepsilon-10\delta$
C_9^3	$K-4\varepsilon-2\delta$	$(K-12\varepsilon-10\delta)/2$	$(K-12\varepsilon-10\delta)/2$
C_{10}^3	$\leq K-8\varepsilon-6\delta$	$\leq K-12\varepsilon-10\delta$	$\leq K-12\varepsilon-10\delta$
C_{11}^3	$K-4\varepsilon-2\delta$	$4\varepsilon+4\delta$	$K-12\varepsilon-10\delta$
C_{12}^3	$K-12\varepsilon-10\delta$	$4\varepsilon+4\delta$	$K-4\varepsilon-2\delta$
C_{13}^3	$K-4\varepsilon-2\delta$	$(K-12\varepsilon-10\delta)/2$	$(K-12\varepsilon-10\delta)/2$
C_{14}^3	$\leq K-8\varepsilon-4\delta$	$\leq K-12\varepsilon-10\delta$	$\leq K-12\varepsilon-10\delta$
C_{15}^3	$K-12\varepsilon-10\delta$	$K-4\varepsilon-4\delta$	$4\varepsilon+6\delta$
C_{16}^3	$\leq K-12\varepsilon-10\delta$	$\leq K-12\varepsilon-10\delta$	$\leq K-8\varepsilon-4\delta$
C_{17}^3	$\leq K-12\varepsilon-10\delta$	$\leq K-12\varepsilon-10\delta$	$\leq K-8\varepsilon-6\delta$
C_{18}^3	$(K-12\varepsilon-10\delta)/2$	$(K-12\varepsilon-10\delta)/2$	$K-4\varepsilon-2\delta$
C_{19}^3	$(K-12\varepsilon-8\delta)/3$	$(K-12\varepsilon-8\delta)/3$	$(K-12\varepsilon-8\delta)/3$

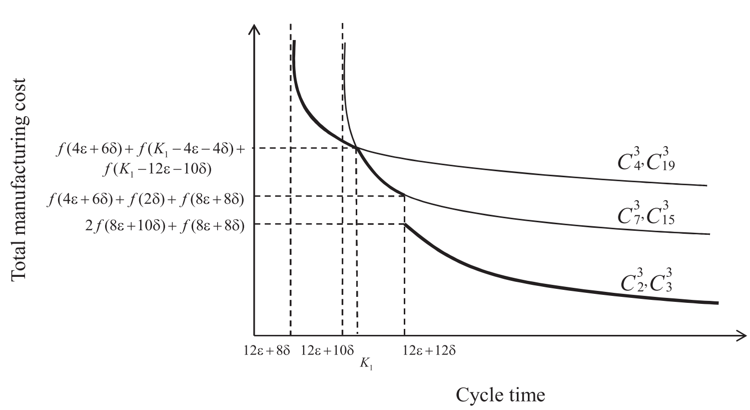

Fig. 3. Efficient frontier of 3-machine cell with controllable processing times.

- Case 2. $\mathbf{P}^*(C^3_{10}|K) \le \mathbf{P}^*(C^3_7|K)$ and $\mathbf{P}^*(C^3_{14}|K) \le \mathbf{P}^*(C^3_7|K)$ and therefore C^3_7 dominates both C^3_{10} and C^3_{14} in the sense of Definition 6. Note that

$$\mathbf{P}^*(C^3_7|K) = \mathbf{P}^*(C^3_5|K) + \left(\frac{K-12\varepsilon-10\delta}{2}, \frac{20\varepsilon+22\delta-K}{2}, -2\delta\right).$$

In other words, $\mathbf{P}^*(C^3_7|K)$ is attained from $\mathbf{P}^*(C^3_5|K)$ by incrementing the second component and decrementing the third component. In the specified region of Case 2, since the increment is more in absolute value than the decrement, and since the manufacturing cost is nondecreasing by assumption, C^3_7 dominates C^3_5 in this region. Similarly,

$$\mathbf{P}^*(C^3_7|K) = \mathbf{P}^*(C^3_{11}|K) + (0, 2\delta, -2\delta)$$

and again by the nondecreasing nature of the manufacturing cost function, we conclude that C^3_7 dominates C^3_{11} for all cycle time values.

If $K_1 \le K < 12\varepsilon+12\delta$, then $\mathbf{P}^*(C^3_7|K) \ge \mathbf{P}^*(C^3_4|K)$ and the dominance of C^3_7 over C^3_4 follows from Definition 6.

- Case 1. If $K < K_1$ then C^3_4 has a lower manufacturing cost value than C^3_7 and since C^3_7 dominates all the other pure cycles, C^3_4 will be the best cycle in this region. □

Finally, to put all the findings of this section into perspective, we provide Fig. 3 which depicts the efficient frontier of the 3-machine cell with bold lines.

5. Conclusion

In this study, we consider an m-machine robot centered cell producing identical parts on identical CNC machines. The existing robotic cell scheduling literature mainly focuses on in-line or mobile robotic cells. In many practical applications, robot centered cells are used simply because they require less space than in-line robotic cell layouts. Furthermore, stationary base robots (as in robot-centered cells) are cheaper to install and easier to program and, consequently, more robust than mobile robots. Initially, we focus on minimizing the cycle time with uniform and fixed processing times on each machine. We present the cycle time lower bound of pure cycles for robot centered cells. We propose two pure cycles and establish that they dominate the rest of the pure cycles for a large range of processing time values. For the remaining region, we provide the worst case performance of the proposed cycles. Later, the processing times are considered as controllable—a situation which is a closer reflection of the real life. The cycle time lower bound is determined for controllable processing times. The proposed two pure cycles are shown to dominate the rest of the pure cycles and the pure cycles in in-line robotic cells, whenever they are feasible. Finally, for the 3-machine case, the bicriteria optimization problem of minimizing both the cycle time and the total manufacturing cost, simultaneously, is solved. Interestingly, pure cycles are used extensively in metal cutting industry, not because they are provably optimal, but because they are very practical and easy to understand and implement. More specifically, in a pure cycle, each part is loaded and unloaded only once, which means less gaging, one probable reason why this cycle is preferred in practice.

Future lines of research directions might be to extend the current study to include multiple part types or dual gripper robots.

Table 4
Cycle times (or lower bounds) of all possible pure cycles of the form stated in Lemma 3.

Robot move sequence	Cycle time
$L_iL_j\ U_iL_k\ U_kU_j$	$12\varepsilon+12\delta+P+\max\{0,P-2\varepsilon-4\delta\}$
$L_iL_j\ U_iL_k\ U_jU_k$	$\ge 12\varepsilon+12\delta+\max\{0,P-2\varepsilon-4\delta\}$
$L_iL_j\ U_iU_k\ L_kU_j$	$\ge 12\varepsilon+12\delta+\max\{0,P-2\varepsilon-4\delta\}$
$L_iL_j\ U_iU_k\ U_jL_k$	$\ge 12\varepsilon+14\delta+\max\{0,P-2\varepsilon-4\delta\}$
$L_iL_j\ U_iU_j\ L_kU_k$	$\ge 12\varepsilon+12\delta+P+\max\{0,P-2\varepsilon-4\delta\}$
$L_iL_j\ U_iU_j\ U_kL_k$	$\ge 12\varepsilon+14\delta+\max\{0,P-2\varepsilon-4\delta\}$
$L_iL_j\ U_jL_kU_kU_i$	$\ge 12\varepsilon+10\delta+2P$
$L_iL_j\ U_jL_k\ U_iU_k$	$12\varepsilon+12\delta+P+\max\{0,P-2\varepsilon-4\delta\}$
$L_iL_j\ U_jU_k\ L_kU_i$	$12\varepsilon+12\delta+P$
$L_iL_j\ U_jU_k\ U_iL_k$	$\ge 12\varepsilon+12\delta+P$
$L_iL_j\ U_jU_i\ L_kU_k$	$\ge 12\varepsilon+10\delta+2P$
$L_iL_j\ U_jU_i\ U_kL_k$	$\ge 12\varepsilon+12\delta+P$
$L_iL_j\ U_kL_k\ U_iU_j$	$\ge 12\varepsilon+12\delta+\max\{0,P-6\varepsilon-8\delta\}$
$L_iL_j\ U_kL_k\ U_jU_i$	$\ge 12\varepsilon+12\delta+\max\{0,P-4\varepsilon-6\delta\}$
$L_iL_j\ U_kU_i\ L_kU_j$	$\ge 12\varepsilon+12\delta+\max\{0,P-4\varepsilon-6\delta\}$
$L_iL_j\ U_kU_i\ U_jL_k$	$\ge 12\varepsilon+14\delta+\max\{0,P-4\varepsilon-6\delta\}$
$L_iL_jU_kU_j\ L_kU_i$	$\ge 12\varepsilon+12\delta+\max\{0,P-2\varepsilon-4\delta\}$
$L_iL_j\ U_kU_j\ U_iL_k$	$\ge 12\varepsilon+14\delta+\max\{0,P-4\varepsilon-6\delta\}$

Appendix

See the Table 4.

References

[1] Brauner N. Identical part production in cyclic robotic cells: concepts, overview and open questions. Discrete Applied Mathematics 2008;156: 2480–92.

[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] Dawande M, Pinedo M, Sriskandarajah C. Multiple part-type production in robotic cells: equivalence of two real-world models. Manufacturing & Service Operations Management 2009;11:210–28.

[5] Drobouchevitch IG, Sethi SP, Sriskandarajah C. Scheduling dual gripper robotic cell: one-unit cycles. European Journal of Operational Research 2006;171:598–631.

[6] Gultekin H, Akturk MS, Karasan OE. Scheduling in robotic cells: process flexibility and cell layout. International Journal of Production Research 2008;46:2105–21.

[7] Gultekin H, Akturk MS, Karasan OE. Bicriteria robotic cell scheduling. Journal of Scheduling 2008;11:457–73.

[8] Gultekin H, Karasan OE, Akturk MS. Pure cycles in flexible robotic cells. Computers & Operations Research 2009;36:329–43.

[9] Han BT, Cook JS. An efficient heuristic for robot acquisition and cell formation. Annals of Operations Research 1998;77:229–52.

[10] van de Klundert JJ. Scheduling problems in automated manufacturing. Dissertation 96-35, Faculty of Economics and Business Administration, Maastricht University, The Netherlands; 1996.

[11] Lei L, Wang TJ. The minimum common cycle algorithm for cyclic scheduling of two hoists with time window constraints. Management Science 1991;37:1629–39.

[12] 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.

[13] Shabtay D, Steiner G. A survey of scheduling with controllable processing times. Discrete Applied Mathematics 2007;155:1643–66.

[14] T'Kindt V, Billaut J-C. Multicriteria scheduling: theory, models and algorithms. 2nd ed.. Berlin: Springer; 2006.

[15] Yildiz S, Akturk MS, Karasan OE. Bicriteria robotic cell scheduling with controllable processing times. International Journal of Production Research, in press, doi:10.1080/00207540903491799.