



Bicriteria robotic cell scheduling with controllable processing times

Serdar Yildiz , M. Selim Akturk & Oya Ekin Karasan

To cite this article: Serdar Yildiz , M. Selim Akturk & Oya Ekin Karasan (2011) Bicriteria robotic cell scheduling with controllable processing times, International Journal of Production Research, 49:2, 569-583, DOI: [10.1080/00207540903491799](https://doi.org/10.1080/00207540903491799)

To link to this article: <http://dx.doi.org/10.1080/00207540903491799>



Published online: 17 Feb 2010.



Submit your article to this journal [↗](#)



Article views: 158



View related articles [↗](#)



Citing articles: 11 View citing articles [↗](#)

Bicriteria robotic cell scheduling with controllable processing times

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

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

(Received 29 January 2009; final version received 12 November 2009)

The current study deals with a bicriteria scheduling problem arising in an m -machine robotic cell consisting of CNC machines producing identical parts. Such machines by nature possess the process flexibility of altering processing times by modifying the machining conditions at differing manufacturing costs. Furthermore, they possess the operational flexibility of being capable of processing all the operations of these identical parts. This latter flexibility in turn introduced a new class of robot move cycles, called pure cycles, to the literature. Within the restricted class of pure cycles, our task is to find the processing times on machines so as to minimise the cycle time and the manufacturing cost simultaneously. We characterise the set of all non-dominated solutions for two specific pure cycles that have emerged as prominent ones in the literature. We prove that either of these pure cycles is non-dominated for the majority of attainable cycle time values. For the remaining regions, we provide the worst case performance of one of these two cycles.

Keywords: robotic cell; CNC; scheduling; bicriteria optimisation; controllable processing times

1. Introduction

Robots are extensively used in many diverse industries ranging from semiconductor manufacturing to electroplating (Dawande *et al.* 2005). The current study has an underlying focus restricted to the metal cutting applications in which the machines are usually CNC machines. Robots are primarily used as material handling instruments. A robotic cell is defined as a manufacturing cell composed of a number of machines and a material handling robot. Figure 1 depicts the m -machine robotic cell considered in this study. We assume that there are no buffers at or between the machines; thus, at any time epoch, a part is either on one of the machines, at the input, or at the output buffer, or on the robot being transported. Note that relaxing this assumption and placing input and output buffers next to each machine can lead to different and more efficient cell structures.

Within the scope of this study lies a set of robot move sequences introduced as *pure cycles* by Gultekin *et al.* (2009) to the literature. Such cycles simply arose as consequences of the inherent operational flexibility of the underlying machines being capable of handling all of the operations of a part. Pure cycles are defined in Gultekin *et al.* (2009) as the robot move sequences in which the robot loads and unloads all of the m machines with a different part during one repetition of the cycle and the initial and the final states are the same so that the cycle can be repeated. Therefore, for each repetition a pure cycle

*Corresponding author. Email: akturk@bilkent.edu.tr

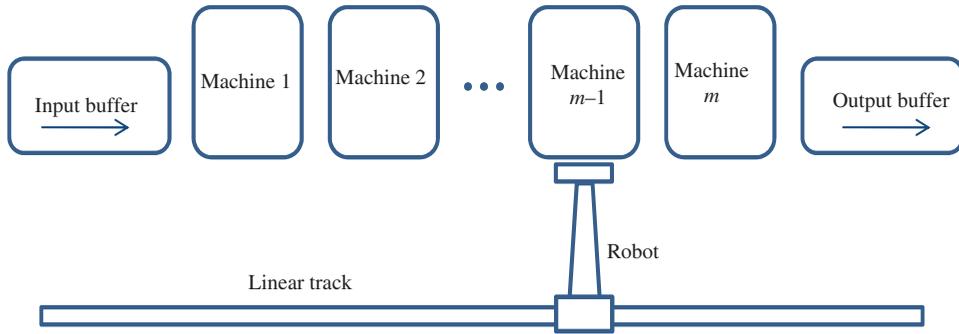


Figure 1. m -Machine inline robotic cell.

produces m parts. In such a cell, the cycle time of a pure cycle is defined as the long run average time required to produce m parts. Each part is completely performed by only one machine and no part is transferred from one machine to another one. Since pure cycles are practical and elementary, they are widely used in industry. Under the assumption of processing times being fixed and identical for all the machines, Gultekin *et al.* (2009) proved that the set of pure cycles dominates all flowshop type robot move cycles with respect to cycle time and showed that two specific pure cycles outperform the remaining pure robot move cycles in a wide range of potential cycle time values. They also derived the worst case performances of these two specific cycles.

Though the single objective of minimising the cycle time is a fundamental one in the existing literature, as far as the authors know, there is only one study, namely that of Gultekin *et al.* (2008), that considers the more realistic bicriteria optimisation problem of minimising the cycle time and the manufacturing cost in robotic cells.

In a flexible manufacturing cell, the processing times can be altered or controlled (albeit at higher cost) by changing machining conditions such as cutting speed and feed rate. Controllable processing times provide additional flexibility in finding solutions to the scheduling problem with improved overall performance of the robotic cell. Most of the studies on scheduling with controllable processing times assume that the processing time is a linear function of the amount of resource allocated to the processing of the job. A summary of such results is presented in the recent survey of Shabtay and Steiner (2007). Since the analysis of linear cost functions is tractable, most of the current literature on controllable processing time problems focus on such functions (e.g. Vickson 1980, Cheng *et al.* 1998). However, using linear cost functions does not reflect the law of diminishing returns. There are some papers that relax the linearity assumption by using either a specific or a general type of convex decreasing resource consumption function (e.g. Lee and Lei 2001, Shakhlevich and Strusevich 2006, Gurel and Akturk 2007, Yedidsion *et al.* 2007). Our study also relaxes the common linear cost assumption and only assumes that the cost function is a monotonically decreasing function.

Dawande *et al.* (2005) present an extensive literature on robotic cell scheduling problems. Crama *et al.* (2000) survey cyclic scheduling problems in robotic flowshops, whereas Galante and Passannanti (2006) study the use of dual gripper robots in a robotic flowshop. In a robotic flowshop, each part must go through all of the m machines in the same sequence. In these systems, it is generally assumed that processing times and allocations of operations to the machines are fixed. We believe this is a critical assumption that limits the flexibility of the expensive CNC machines unnecessarily. Research on

robotic cells focuses on minimising the cycle time; in other words, maximising the throughput. Since 1-unit cycles are easy to implement and easy to analyse theoretically, studies on robotic cells focus on these cycles. Sethi *et al.* (1992) proved that 1-unit cycles give optimal solutions in two machine robotic cells producing identical parts. Nevertheless, 1-unit cycles are not always the optimal cycles for maximising the throughput for a higher number of machines. In this study, we consider a scheduling problem of an m -machine flexible robotic cell with m -unit cycles producing identical parts. For a more detailed discussion on identical parts in cyclic robotic cells, we refer the interested reader to Brauner (2008).

The organisation of this paper is as follows. In the next section, the notation and basic definitions to be used throughout the paper are presented. In Section 3, m -machine cells are analysed and the bicriteria optimisation problem of simultaneously minimising the cycle time and the manufacturing cost is tackled. Finally, Section 4 includes some conclusions and future research directions.

2. Notation and definitions

In this section, we adopt the standard terminology from the robotic cell literature and present the distinguishing features pertinent to the current study.

The current study focuses on robot move sequences defined by Gultekin *et al.* (2009) as pure cycles which simply arose as consequences of the inherent flexibility of the cells considered. More specifically, each machine is capable of performing all of the operations making up any one of the identical parts. Gultekin *et al.* (2009) use the following definitions to characterise pure cycles.

Definition 2.1: L_i is the robot activity in which the robot takes a part from the input buffer and loads machine i , $i = 1, 2, \dots, m$. Similarly, U_i , $i = 1, 2, \dots, m$, is the robot activity in which the robot unloads machine i and drops the part to the output buffer. Let $\mathcal{A} = \{L_1, \dots, L_m, U_1, \dots, U_m\}$ be the set of all activities.

Definition 2.2: Under a **pure cycle**, starting with an initial state of the cell, the robot performs each of the $2m$ activities (L_i, U_i , $i = 1, \dots, m$) exactly once and returns to the initial state of the cell.

In other words, any permutation of the m load and the m unload activities results in a pure cycle. For example, in a 2-machine robotic cell, the robot activity set is $\mathcal{A} = \{L_1, L_2, U_1, U_2\}$ and the robot move sequence $L_1U_1L_2U_2$ is a pure cycle. Since there are m machines in the robotic cell under consideration, each pure cycle produces m parts and is consequently an m -unit cycle in the classification of Dawande *et al.* (2005). It is important to note that there is not necessarily a single efficient pure cycle. In this study, we shall let C_i^m define the i th pure cycle in an m -machine robotic cell and $T_{C_i^m}$ denote its corresponding cycle time. All the operations of any one of the identical parts are to be processed solely on one of the identical machines. We let the decision variable P_i denote the processing time of any one of the identical parts on machine i such that the processing times are the same on each machine, but they can vary between machines. We assume that a feasible processing time on any machine is bounded from above by an upper bound denoted as P^U . We denote by a processing time vector $\mathbf{P} = (P_1, P_2, \dots, P_m)$ the processing times on individual machines. In a feasible processing time vector, all of the processing times have to obey the non-negativity and upper bound restrictions.

In particular, the set of feasible processing time vectors is defined as $P_{\text{feas}} = \{(P_1, P_2, \dots, P_m) \in R^m: 0 \leq P_i \leq P^U, \forall i\}$.

We adopt the following nomenclature:

- ε are the load/unload times of machines by the robot which are the same for all machines;
- δ is the time taken by the robot to travel between two consecutive machines, which is additive such that the travelling time from machine i to machine j is equal to $|i - j|\delta$;
- $f(P_i)$ is the manufacturing cost incurred from processing time on machine i ; this cost is assumed to be monotonically decreasing for increasing processing times.

There are two objectives:

- (1) $F_1(\mathbf{P}) = \sum_{i=1}^m f(P_i)$ is the total manufacturing cost depending only on the processing times;
- (2) $F_2(C_i^m, \mathbf{P})$ is the cycle time corresponding to processing time vector \mathbf{P} and the pure cycle C_i^m , i.e. the total time required to complete the m -unit pure cycle C_i^m .

Although there are $(2m)!$ possible pure cycles, some of them correspond to the same move sequences. For instance, in a 2-machine cell, $L_1U_1L_2U_2$ and $L_2U_2L_1U_1$ are different permutations representing the same cycle. As observed in Gultekin *et al.* (2009), there are $(2m - 1)!$ distinct pure cycles in an m -machine cell. Let Π be the set of all $(2m - 1)!$ pure cycles, i.e.

$$\Pi = \bigcup_{i=1}^{(2m-1)!} C_i^m.$$

Finding a pure cycle among this group which outperforms the rest in terms of, say, the objective of minimisation of the cycle time, does not appear to be an easy task. The constituents of the cycle time are the time involved during robot activities, such as load, unload and part transfer operations, and the time spent during the waiting periods of the robot in front of machines for unloading operations. As such, finding a pure cycle with the minimum cycle time entails choosing a load and unload sequence of the machines by the robot in a way that the two aspects of the cycle time balance each other. Though without an accompanying proof, it is a strong belief of the authors that such a task is computationally quite cumbersome. However, the computational complexity status of this problem is an interesting open question. With this in mind, in this paper we focus on the following two particular cycles, which have emerged as favourable ones in Gultekin *et al.* (2009).

Definition 2.3: C_1^m is the robot move cycle in an m -machine robotic cell with the activity sequence $L_1L_mU_{m-1}L_{m-1}U_{m-2}L_{m-2} \dots U_2L_2U_1U_m$.

Definition 2.4: C_2^m is the robot move cycle in an m -machine robotic cell with the activity sequence $L_1U_mL_mU_{m-1}L_{m-1} \dots U_2L_2U_1$.

In the initial state of the cycle C_1^m , machines 1 and m are idle and the rest of the machines are each loaded with a part. In the initial state of the cycle C_2^m , only machine 1 is idle and the rest of the machines are loaded.

The total manufacturing cost is the sum of tooling and machining costs. As the processing times increase, the machining cost increases and the tooling cost decreases. Conversely, reducing the processing times decreases the machining cost, but increases the tooling cost. In this paper, we have defined the processing time upper bound P^U as the

processing time value that minimises 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. As a result, we assume that the manufacturing cost is a monotonically decreasing function of the processing times. The total manufacturing cost of a repetition of a cycle is defined as the sum of the manufacturing costs incurred by the processing times of all machines and hence depends only on the processing times but not on the robot move cycle. In contrast, the cycle time is the time required to complete the activities in the cycle and finally return back to the initial state, which depends on both the robot move cycle and processing times.

Against this background, our resulting problem deals with the following bicriteria optimisation of minimising the cycle time and the total manufacturing cost simultaneously:

$$\begin{aligned} & \text{Minimise} && \text{Total manufacturing cost} \\ & \text{Minimise} && \text{Cycle time} \\ & \text{Subject to} && \mathbf{P} \in \mathcal{P}_{\text{feas}}. \end{aligned}$$

There are different strategies, such as those classified in Hoogeveen (2005), to tackle optimisation problems of a multicriteria nature. Typically, one seeks to find non-dominated solutions, where a solution is defined as a non-dominated one if no other solution is as good as it is for all the objectives and is better for at least one objective. Since it is hard to determine which performance measure is more important, it is useful to present an extensive list of non-dominated solutions and give the decision maker the opportunity to select the most appropriate solution. Ideally, any feasible solution of the above bicriteria optimisation problem corresponds to a feasible robot move sequence and a feasible processing time vector. In our approach, we shall solve the above bicriteria optimisation problem independently for each robot move sequence under consideration. In order to generate the non-dominated points, we shall resort to the epsilon-constraint approach as presented in Hoogeveen (2005). The epsilon constraint formulation of the problem for pure cycle C_i^m is denoted as $\epsilon(F_1(\mathbf{P})|F_2(C_i^m, \mathbf{P}))$ where one finds the processing time vector minimising the total manufacturing cost $F_1(\mathbf{P})$ for a given level of cycle time $F_2(C_i^m, \mathbf{P})$. Thus, given any cycle C_i^m and cycle time level K , the following epsilon-constraint problem (ECP) is solved to find the corresponding non-dominated processing time vector:

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

More formally, we seek to find the set of all the non-dominated processing time vectors for an m -unit robot move cycle C_i^m and for a given cycle time level K as defined below.

Definition 2.5: For a robot move sequence C_i^m and a given cycle time level K , the set of non-dominated points is defined as $\mathbf{P}^*(C_i^m|K) = \{\mathbf{P} \in \mathcal{P}_{\text{feas}} : \text{there is no other } \mathbf{P}' \in \mathcal{P}_{\text{feas}} \text{ such that } F_1(\mathbf{P}') < F_1(\mathbf{P}) \text{ where } F_2(C_i^m, \mathbf{P}) \leq K \text{ and } F_2(C_i^m, \mathbf{P}') \leq K\}$.

In the next section, this set of efficient processing time vectors, i.e. the efficient frontier, is presented for each of the considered pure cycles. Moreover, it has been shown that one of these cycles will be non-dominated for the majority of attainable cycle time values.

3. Solution procedure

With the following theorems and lemmas we shall proceed to find the cycle times and the efficient frontiers corresponding to the aforementioned two pure cycles. The performances of these two prominent cycles are then compared to the other pure cycles in Theorems 3.8 and 3.9, and sufficient conditions under which one of these two cycles is efficient are characterised.

Given a processing time vector $\mathbf{P} = (P_1, \dots, P_m)$, we let $P_{\max}^1 = \max_{i=2, \dots, m-1} P_i$.

Theorem 3.1: *The cycle time of C_1^m for a given processing time vector \mathbf{P} is $T_{C_1^m} = 4m\epsilon + 2m(m+1)\delta + \max\{0, P_1 - [(4m-6)\epsilon + 2(m^2-2)\delta], P_m - [(4m-6)\epsilon + 2(m^2-2)\delta], P_{\max}^1 - [(4m-4)\epsilon + 2(m^2-1)\delta]\}$.*

Proof: The proof is presented in Appendix A. □

When there is a given processing time vector, Theorem 3.1 determines the corresponding cycle time obtained from cycle C_1^m . Conversely, with the result of this theorem, for a specified cycle time level, the largest individual processing times on machines (ultimately the least manufacturing cost alternative) that obey this cycle time level can be found as well.

Given \mathbf{P} , let $P_{\max}^2 = \max_{i=1, \dots, m} P_i$, then we have the following theorem.

Theorem 3.2: *The cycle time of C_2^m for a given processing time vector \mathbf{P} is $T_{C_2^m} = 4m\epsilon + 2[(m+1)^2 - 2]\delta + \max\{0, P_{\max}^2 - [(4m-4)\epsilon + 2(m-1)(m+2)\delta]\}$.*

Proof: For the proof, we refer the reader to Appendix B. □

The next theorem provides a lower bound for the cycle time of any pure cycle.

Theorem 3.3: *For an m -machine robotic cell with given processing time vector \mathbf{P} , the cycle time of any pure cycle is no less than*

$$T_{\text{contr}} = \max\{4m\epsilon + 2m(m+1)\delta, 4\epsilon + (2m+2)\delta + P_{\max}^2\}. \quad (1)$$

Proof: The cycle time of any pure cycle has to be greater than or equal to two lower bounds. The first lower bound is obtained from the robot activity time and the second one is obtained from the given processing time vector. First, a part is taken from the input buffer (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 output buffer (ϵ). This makes a total of $4m\epsilon$ for any pure cycle. For any part, the robot takes the part from the input buffer to the output buffer $(m+1)\delta$. Then, the robot travels from the output buffer to the input buffer to take a new part or to complete the cycle $(m+1)\delta$. This makes a total of $2m(m+1)\delta$ time units for any cycle. Consequently, the first lower bound, which is the total time required to complete the set of robot activities, evaluates to $4m\epsilon + 2m(m+1)\delta$.

The second lower bound is derived from the minimum time between two consecutive loadings of any machine. The minimum time needed to unload machine i after loading it is P_i time units. After processing on the part is finished, it is unloaded (ϵ), the part is

transferred to output buffer $(m + 1 - i)\delta$, and the part is dropped (ϵ). After that, the robot travels to the input buffer to take a new part to make the consecutive loading of machine i ($(m + 1)\delta$), takes a new part, part (ϵ), brings the new part to machine i ($i\delta$) and finally loads the machine (ϵ). Hence, the minimum time required between two consecutive loadings of machine i is $4\epsilon + (2m + 2)\delta + P_i$. However, there are m machines and the processing times on these machines may be different from each other, due to controllability. Thus, the cycle time has to be at least equal to the minimum time required between two consecutive loadings of the machine having the worst such time. So, the second lower bound of the cycle time becomes $4\epsilon + (2m + 2)\delta + P_{\max}^2$. \square

A non-dominated processing time vector provides the minimum cost for a given cycle time level. The total manufacturing cost is a function of only the processing times on machines. Recall that the manufacturing cost of a machine decreases as the processing time increases on that machine and that a feasible processing time must satisfy $0 \leq P_i \leq P^U$.

For a given cycle time level, say K , satisfying the lower bound of Theorem 3.3, we can find the upper bounds of processing times for pure cycles that do not violate this cycle time level. Since this processing time vector is an upper bound for the processing time vectors obtained from pure cycles for a cycle time level K , it also results in the lower bound of total manufacturing cost that a pure cycle can result in. Let $\bar{P}(K) = (\bar{P}_1(K), \dots, \bar{P}_m(K))$ denote the upper bound of processing time vectors. Now, $\bar{P}(K)$ for a given cycle time level K is found as follows.

Lemma 3.4: For a given cycle time level K , the upper bound of processing time vectors for pure cycles is:

$$\bar{P}(K) = (\bar{P}_1(K), \dots, \bar{P}_m(K)),$$

where $\bar{P}_i(K) = \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\}, \forall i$.

Proof: The two bounds constraining processing time vectors are found in the following cases.

- (1) The P^U value cannot be exceeded by any feasible processing time on a machine. This leads to $\bar{P}_i(K) \leq P^U, \forall i$.
- (2) The processing times on the machines are additionally bounded from above since otherwise the cycle time level K would be exceeded. Theorem 3.3 determines the lower bound for the cycle time when a processing time vector is given. In other words, we must have

$$\underline{T}_{\text{contr}} = \max\{4m\epsilon + 2m(m + 1)\delta, 4\epsilon + (2m + 2)\delta + \max\{P_i, i: 1, \dots, m\}\} \leq K.$$

- (3) Therefore, $\max\{P_i, i: 1, \dots, m\} \leq K - [4\epsilon + (2m + 2)\delta]$.
- (4) This implies that $\bar{P}_i(K) \leq K - [4\epsilon + (2m + 2)\delta], \forall i$.

Consequently, $\bar{P}(K)$ is the upper bound of processing time vectors satisfying the two bounds described above for a given cycle time level. \square

In order to compare the total manufacturing costs obtained from cycles C_1^m and C_2^m to the remaining pure cycles, we construct with Lemmas 3.5 and 3.7 the efficient frontiers (the set of non-dominated processing time vectors) corresponding to these two cycles, respectively.

In the following lemma, we determine $P^*(C_1^m|K)$, the set of non-dominated processing time vectors for cycle C_1^m . Theorem 3.1 implies that the minimum cycle time value for C_1^m is $[4m\epsilon + 2m(m + 1)\delta]$, thus the ECP problem is solved for cycle time level K satisfying this bound to construct the efficient frontier.

Lemma 3.5: *For the cycle C_1^m given any feasible cycle time level K , i.e. $K \geq 4m\epsilon + 2m(m + 1)\delta$, there exists a unique non-dominated processing time vector $(P_1^*, P_2^*, \dots, P_m^*) \in P^*(C_1^m|K)$ defined as*

$$P^*(C_1^m|K) = \begin{bmatrix} P_1^* \\ P_2^* \\ \vdots \\ P_{m-1}^* \\ P_m^* \end{bmatrix} = \begin{bmatrix} \min\{P^U, K - [6\epsilon + (2m + 4)\delta]\} \\ \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\} \\ \vdots \\ \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\} \\ \min\{P^U, K - [6\epsilon + (2m + 4)\delta]\} \end{bmatrix}.$$

Proof: Now we determine the non-dominated vector by describing the upper bounds constraining the processing time vector $P^*(C_1^m|K)$. There are two obvious upper bounds the processing times must obey:

- (1) any feasible processing time is at most equal to P^U , i.e. $P_i^* \leq P^U, \forall i$;
- (2) additionally, the processing times on individual machines must ensure the cycle time of Theorem 3.1 does not exceed the value K . A simple inspection of this formula reveals the following largest possible individual processing times on each machine without exceeding the given cycle time:

$$\begin{bmatrix} P_1^* \\ P_2^* \\ \vdots \\ P_{m-1}^* \\ P_m^* \end{bmatrix} \leq \begin{bmatrix} K - [6\epsilon + (2m + 4)\delta] \\ K - [4\epsilon + (2m + 2)\delta] \\ \vdots \\ K - [4\epsilon + (2m + 2)\delta] \\ K - [6\epsilon + (2m + 4)\delta] \end{bmatrix}.$$

□

The following example is beneficial in conveying the contribution of controllable processing times. The total manufacturing cost of cycle C_1^m with controllable processing times, as discussed in this study, is compared to the total manufacturing cost of C_1^m in Gultekin *et al.* (2009), where the processing times on machines are assumed to be fixed and the same for all machines.

Example 3.6: Consider a 4-machine robotic cell. We will show that the cycle C_1^4 with controllable processing times results in less cost than C_1^4 with fixed processing times for the same cycle time level. Let $\epsilon = 0.2, \delta = 0.1, P^U = 6.5$, and assume that the required cycle time level is $K = 8.0$.

By using Lemma 3.5, for cycle time level K , the non-dominated processing time vector, $(P_1^*, P_2^*, P_3^*, P_4^*) \in P^*(C_1^4)$, giving the minimum total manufacturing cost found as follows:

$$\begin{bmatrix} P_1^* \\ P_2^* \\ P_3^* \\ P_4^* \end{bmatrix} = \begin{bmatrix} \min\{P^U, K - [6\epsilon + (2m + 4)\delta]\} \\ \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\} \\ \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\} \\ \min\{P^U, K - [6\epsilon + (2m + 4)\delta]\} \end{bmatrix} = \begin{bmatrix} \min\{6.5, 5.6\} \\ \min\{6.5, 6.2\} \\ \min\{6.5, 6.2\} \\ \min\{6.5, 5.6\} \end{bmatrix} = \begin{bmatrix} 5.6 \\ 6.2 \\ 6.2 \\ 5.6 \end{bmatrix}.$$

The cycle time corresponding to a fixed processing time P on each machine, as presented in Gultekin *et al.* (2009) is

$$T_{C_1^m} = 4m\epsilon + 2m(m + 1)\delta + \max\{0, P - [(4m - 6)\epsilon + 2(m^2 - 2)\delta]\}.$$

After a simple calculation, we obtain $P \leq T_{C_1^m} - 6\epsilon - (2m + 4)\delta$.

For the given set of data in this example, $P \leq 5.6$ for all machines, therefore $P_{\text{fixed}}^*(C_1^4|8.0) = (5.6, 5.6, 5.6, 5.6)$. Since

$$P_{\text{fixed}}^*(C_1^4|8.0) = \begin{bmatrix} P \\ P \\ P \\ P \end{bmatrix} = \begin{bmatrix} 5.6 \\ 5.6 \\ 5.6 \\ 5.6 \end{bmatrix} \leq \begin{bmatrix} 5.6 \\ 6.2 \\ 6.2 \\ 5.6 \end{bmatrix} = \begin{bmatrix} P_1^* \\ P_2^* \\ P_3^* \\ P_4^* \end{bmatrix} = P^*(C_1^4|8.0),$$

by comparing the processing times of $P_{\text{fixed}}^*(C_1^4|8.0)$ and $P^*(C_1^4|8.0)$, we see that $P = P_1^* = P_4^*$ and $P < P_2^* = P_3^*$. Thus, the non-dominated processing time vector of C_1^4 with controllable processing times results in lower total manufacturing cost when compared against the fixed processing time alternative.

In the following lemma, we determine $P^*(C_2^m|K)$, the set of non-dominated processing time vectors for cycle C_2^m that simultaneously minimise the cycle time and the total manufacturing cost. It can be seen from Theorem 3.2 that the minimum cycle time value for C_2^m is $\{4m\epsilon + 2[(m + 1)^2 - 2]\delta\}$, thus the ECP problem is solved for cycle time level K respecting this boundary to construct the efficient frontier.

Lemma 3.7: *For the cycle C_2^m given any feasible cycle time level K , i.e. $K \geq 4m\epsilon + 2[(m + 1)^2 - 2]\delta$, there exists a unique non-dominated processing time vector $(P_1^*, P_2^*, \dots, P_m^*) \in P^*(C_2^m|K)$ defined as*

$$P^*(C_2^m|K) = \begin{bmatrix} P_1^* \\ \vdots \\ P_m^* \end{bmatrix} = \begin{bmatrix} \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\} \\ \vdots \\ \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\} \end{bmatrix}.$$

Proof: The proof follows closely that of Lemma 3.5 and hence is omitted here. □

With Theorems 3.8 and 3.9 we prove that, in terms of manufacturing cost, one of the two prominent pure cycles is efficient over the majority of the feasible cycle time region of pure cycles obtained from Theorem 3.3, i.e. the region where $4m\epsilon + 2m(m + 1)\delta \leq K$. We analyse this cycle time region in two parts. The first region is where C_2^m is feasible and the second region is where C_2^m is not feasible but C_1^m is feasible. We can see from Theorems 3.1 and 3.2 that C_2^m is feasible whenever the cycle time satisfies $4m\epsilon + 2[(m + 1)^2 - 2]\delta \leq K$ and infeasible when $4m\epsilon + 2m(m + 1)\delta \leq K < 4m\epsilon + 2[(m + 1)^2 - 2]\delta$ where C_1^m is feasible.

Theorem 3.8: *If C_2^m is feasible, then it is also efficient.*

Proof: The non-dominated processing time vector obtained from C_2^m for the cycle time level K is determined in Lemma 3.7 as follows:

$$P^*(C_2^m|K) = (P_1^*, P_2^*, \dots, P_m^*),$$

where

$$P_i^* = \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\}, \quad \forall i.$$

The upper bound of processing time vectors is determined in Lemma 3.4 as follows:

$$\bar{P}(K) = (\bar{P}_1(K), \dots, \bar{P}_m(K)),$$

where

$$\bar{P}_i(K) = \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\}, \quad \forall i.$$

Hence, the processing time vector obtained from C_2^m is equal to the upper bound of processing time vectors for any cycle time level K . Consequently, the total manufacturing cost of $P^*(C_2^m|K)$ is equal to the lower bound of the total manufacturing cost obtained from $\bar{P}(K)$. \square

In the next theorem, we prove that C_1^m is efficient in the remaining region under the specified condition.

Theorem 3.9: *If $4m\epsilon + 2m(m + 1)\delta \leq K < 4m\epsilon + 2[(m + 1)^2 - 2]\delta$, and $P^U \leq K - [6\epsilon + (2m + 4)\delta]$, then C_1^m is efficient.*

Proof: The non-dominated processing time vector for cycle C_1^m is found by Lemma 3.5. Under the extra condition that $P^U \leq K - [6\epsilon + (2m + 4)\delta]$, it is easy to verify that

$$(P_1^*, P_2^*, \dots, P_m^*) = (P^U, P^U, \dots, P^U) = \bar{P}(K)$$

using Lemmas 3.4 and 3.5.

Since all the processing times on the machines take their maximum value, there is no other pure cycle that can result in smaller total manufacturing cost. \square

We now proceed to determine the worst case performance of C_1^m with respect to the total manufacturing cost. The region considered in Lemma 3.10 is the only region where neither C_1^m nor C_2^m is guaranteed to be efficient. Since C_1^m is feasible and C_2^m is not feasible in this region, only C_1^m is considered. The term $F_1(P^*(C_1^m|K))$ denotes the total manufacturing cost incurred by the non-dominated processing time vector of C_1^m for the given cycle time level K . Similarly, the term $F_1^{LB}(\bar{P}(K))$ denotes the lower bound of the total manufacturing cost of pure cycles for the given cycle time level K . We present the performance analysis below in which $f(\chi)$ gives the manufacturing cost on a machine when the processing time is equal to χ .

Lemma 3.10: *If $4m\epsilon + 2m(m + 1)\delta \leq K < 4m\epsilon + 2[(m + 1)^2 - 2]\delta$ and $K - [6\epsilon + (2m + 4)\delta] < P^U$, the worst case performance of C_1^m is bounded as*

$$F_1(P^*(C_1^m|K)) \leq F_1^{LB}(\bar{P}(K)) \times \varrho,$$

where

$$\varrho = \left\{ 1 - \frac{2}{m} + \frac{2f[(4m - 6)\epsilon + 2(m^2 - 2)\delta]}{mf[(4m - 4)\epsilon + 2(m^2 - 1)\delta]} \right\}.$$

Proof: Take the non-dominated processing time vector $(P_1^*, P_2^*, \dots, P_m^*) = P^*(C_1^m|K)$. Using Lemma 3.5 and the fact that $K - [6\epsilon + (2m + 4)\delta] < P^U$, it is easy to conclude that

$$P^*(C_1^m|K) = \begin{bmatrix} P_1^* \\ P_2^* \\ \vdots \\ P_{m-1}^* \\ P_m^* \end{bmatrix} = \begin{bmatrix} K - [6\epsilon + (2m + 4)\delta] \\ \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\} \\ \vdots \\ \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\} \\ K - [6\epsilon + (2m + 4)\delta] \end{bmatrix}.$$

We compare the total manufacturing cost of $\mathbf{P}^*(C_1^m|K)$ to the lower bound of total manufacturing cost which corresponds to the upper bound of processing time vectors for cycle time level K . $\bar{\mathbf{P}}(K)$ can be found by using Lemma 3.4 as follows:

$$\bar{\mathbf{P}}(K) = (\bar{P}_1(K), \dots, \bar{P}_m(K)),$$

where

$$\bar{P}_i(K) = \min\{P^U, K - [4\epsilon + (2m + 2)\delta]\}, \quad \forall i.$$

The total manufacturing cost obtained from $\mathbf{P}^*(C_1^m|K)$ is calculated as follows:

$$F_1(\mathbf{P}^*(C_1^m|K)) = (m - 2)f(\min\{P^U, K - [4\epsilon + (2m + 2)\delta]\}) + 2f\{K - [6\epsilon + (2m + 4)\delta]\}.$$

The lower bound of total manufacturing cost for cycle time level K is found as

$$F_1^{LB}(\bar{\mathbf{P}}(K)) = \sum_{i=1}^m f(P_i) = mf(\min\{P^U, K - [4\epsilon + (2m + 2)\delta]\}).$$

Now, we can calculate the worst case performance by dividing the total manufacturing cost obtained from cycle C_1^m to the lower bound of total manufacturing cost.

$$\begin{aligned} \frac{F_1(\mathbf{P}^*(C_1^m|K))}{F_1^{LB}(\bar{\mathbf{P}}(K))} &= \frac{(m - 2)f(\min\{P^U, K - [4\epsilon + (2m + 2)\delta]\}) + 2f\{K - [6\epsilon + (2m + 4)\delta]\}}{mf(\min\{P^U, K - [4\epsilon + (2m + 2)\delta]\})} \\ &= 1 - \frac{2}{m} + \frac{2f\{K - [6\epsilon + (2m + 4)\delta]\}}{mf(\min\{P^U, K - [4\epsilon + (2m + 2)\delta]\})}. \end{aligned}$$

Since $f\{K - [4\epsilon + (2m + 2)\delta]\} \leq f(\min\{P^U, K - [4\epsilon + (2m + 2)\delta]\})$,

$$\frac{f\{K - [6\epsilon + (2m + 4)\delta]\}}{f(\min\{P^U, K - [4\epsilon + (2m + 2)\delta]\})} \leq \frac{f\{(K - [6\epsilon + (2m + 4)\delta])\}}{f\{K - [4\epsilon + (2m + 2)\delta]\}}.$$

We have assumed that $4m\epsilon + 2m(m + 1)\delta \leq K$, so we can easily show that

$$\frac{f\{K - [6\epsilon + (2m + 4)\delta]\}}{f\{K - [4\epsilon + (2m + 2)\delta]\}} \leq \frac{f[(4m - 6)\epsilon + 2(m^2 - 2)\delta]}{f[(4m - 4)\epsilon + 2(m^2 - 1)\delta]},$$

thus

$$\frac{F_1(\mathbf{P}^*(C_1^m|K))}{F_1^{LB}(\bar{\mathbf{P}}(K))} \leq 1 - \frac{2}{m} + \frac{2f[(4m - 6)\epsilon + 2(m^2 - 2)\delta]}{mf[(4m - 4)\epsilon + 2(m^2 - 1)\delta]}.$$

□

As can be seen from the statement in Lemma 3.10, the number of machines directly affects the difference in total manufacturing cost between the total manufacturing cost of C_1^m and the lower bound. The following result, which states that the worst case performance of C_1^m gets better as the number of machines increases, immediately follows from Lemma 3.10.

Corollary 3.11: *As m tends to ∞ , the total manufacturing cost of C_1^m approaches the lower bound of the total manufacturing cost.*

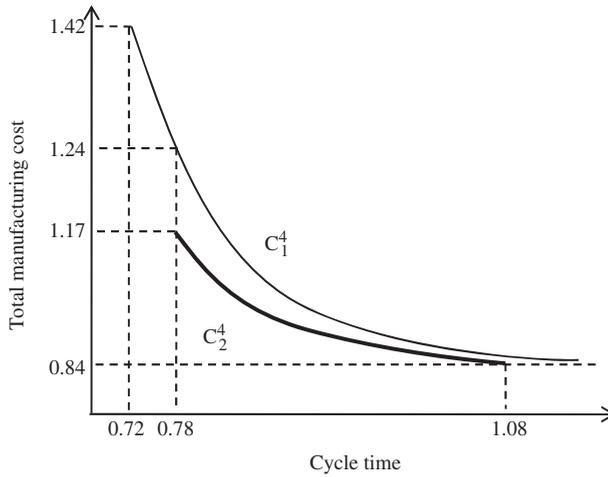


Figure 2. Total manufacturing cost with respect to cycle time.

Proof: Clearly, $\varphi \rightarrow 1$ as $m \rightarrow \infty$ and the result follows. □

The next example presents the efficient frontiers of cycles C_1^m and C_2^m for given parameters and illustrates Theorem 3.8 and Lemma 3.10.

The cost function used in the next example is modified from the cost functions given in Kayan and Akturk (2005). There is a set of identical parts requiring a single pass turning operation performed on identical CNC turning machines. The total manufacturing cost is the sum of the machining and tooling costs on each machine. The machining cost for machine i is defined as $O \times P_i$, where O is the operating cost which is identical for all machines. The tooling cost on machine i is defined as $T \times U \times P_i^\alpha$, where $T > 0$ and $\alpha < 0$ are constants for identical tools and $U > 0$ is a specific constant for identical operations using identical tools. Consequently, the manufacturing cost for machine i is $f(P_i) = O \times P_i + T \times U \times P_i^\alpha$.

Example 3.12: In this example, we consider a flexible robotic cell with four CNC turning machines. For this turning operation, let the parameters be given as $T = 0.1$, $O = 0.1$, $U = 1$, $\alpha = -1.6423$, $\epsilon = 0.02$, $\delta = 0.01$ and $P^U = 0.90$. The cost function corresponding to processing times on the CNC machines is described as $f(P_i) = O \times P_i + T \times U \times P_i^\alpha$.

The two curves in Figure 2 represent the efficient frontiers of C_1^4 and C_2^4 , which are constructed by using Lemmas 3.5 and 3.7, respectively. In this figure, $4m\epsilon + 2[(m + 1)^2 - 2]\delta = 0.78$ is the point found from Theorem 3.2 where C_2^4 becomes feasible. The curves clearly illustrate the findings in Theorem 3.8 that C_2^4 dominates the rest of the pure cycles when the cycle time is at least 0.78. In addition, $4m\epsilon + 2m(m + 1)\delta = 0.72$ is the point where C_1^4 becomes feasible. The only region where C_2^4 is not feasible is the cycle time region $0.72 \leq K < 0.78$. In this region, C_1^4 is feasible. Since $K - [6\epsilon + (2m + 4)\delta] < P^U$ in this region, we cannot say that C_1^4 dominates the rest of the pure cycles by using Theorem 3.9. However, by using Lemma 3.10, we can calculate the worst case performance of C_1^4 in this region as $\varphi = 1.08$. In other words, the total manufacturing cost obtained from cycle C_1^4 is at most 8% higher than the lower bound of the total manufacturing cost in this region.

The next section concludes this study and presents some future research directions.

4. Conclusion

In this study, we considered an m -machine flexible robotic cell composed of CNC machines producing identical parts. The machines are assumed to be capable of performing all of the operations for identical parts. Since the machines are highly flexible, the processing times are assumed to be controllable, in which the machining conditions can be changed to decrease or increase the processing times. This study is restricted to a new class of cycles, called pure cycles, which results from the flexibility of the underlying machines. Given the robot move sequence, we consider finding non-dominated processing time solutions to the bicriteria problem of minimising the cycle time and the total manufacturing cost simultaneously.

We analysed two specific pure cycles and determined the non-dominated solutions for these cycles in Lemmas 3.5 and 3.7. With Theorems 3.8 and 3.9, we proved that one of these two prominent cycles is efficient in most of the regions. For the remaining region, we determined the worst case performance. The results show that these two prominent cycles are not only simple and practical to implement, but also efficient.

Given a fixed processing time vector, whether the problem of finding the pure cycle with the best cycle time value is NP-hard or not remains an open question. As future research directions, the results of this study can be extended to cells producing multiple parts or cells possessing a dual gripper robot. In order to obtain more efficient solutions for both cycle time and processing time criteria, different robotic cell structures such as those involving input and output buffers beside individual machines can be considered.

References

- Brauner, N., 2008. Identical part production in cyclic robotic cells: concepts, overview and open questions. *Discrete Applied Mathematics*, 156 (13), 2480–2492.
- Cheng, T.C.E., Janiak, A., and Kovalyov, M.Y., 1998. Bicriterion single machine scheduling with resource dependent processing times. *SIAM Journal on Optimization*, 8 (2), 617–630.
- Crama, Y., *et al.*, 2000. Cyclic scheduling in robotic flowshops. *Annals of Operations Research*, 96 (1–4), 97–124.
- Dawande, M., *et al.*, 2005. Sequencing and scheduling in robotic cells: recent developments. *Journal of Scheduling*, 8 (5), 387–426.
- Galante, G. and Passannanti, G., 2006. Minimizing the cycle time in serial manufacturing systems with multiple dual-gripper robots. *International Journal of Production Research*, 44 (4), 639–652.
- Gultekin, H., Akturk, M.S., and Karasan, O.E., 2008. Bicriteria robotic cell scheduling. *Journal of Scheduling*, 11 (6), 457–473.
- Gultekin, H., Karasan, O.E., and Akturk, M.S., 2009. Pure cycles in flexible robotic cells. *Computers & Operations Research*, 36 (2), 329–343.
- Gurel, S. and Akturk, M.S., 2007. Considering manufacturing cost and scheduling performance on a CNC turning machine. *European Journal of Operational Research*, 177 (1), 325–343.
- Hoogeveen, H., 2005. Multicriteria scheduling. *European Journal of Operational Research*, 167 (3), 592–623.
- Kayan, R.K. and Akturk, M.S., 2005. A new bounding mechanism for the CNC machine scheduling problems with controllable processing times. *European Journal of Operational Research*, 167 (3), 624–643.
- Lee, C.-Y. and Lei, L., 2001. Multiple-project scheduling with controllable project duration and hard resource constraint: some solvable cases. *Annals of Operations Research*, 102 (1–4), 287–307.

- Sethi, S.P., *et al.*, 1992. Sequencing of parts and robot moves in a robotic cell. *International Journal of Flexible Manufacturing Systems*, 4 (3–4), 331–358.
- Shakhlevich, N.V. and Strusevich, V.A., 2006. Single machine scheduling with controllable release and processing parameters. *Discrete Applied Mathematics*, 154 (15), 2178–2199.
- Shabtay, D. and Steiner, G., 2007. A survey of scheduling with controllable processing times. *Discrete Applied Mathematics*, 155, 1643–1666.
- Vickson, R.G., 1980. Choosing the job sequence and processing times to minimize processing plus flow cost on a single machine. *Operations Research*, 28 (5), 1155–1167.
- Yedidsion, L., Shabtay, D., and Kaspi, M., 2007. A bicriteria approach to minimise maximal lateness and resource consumption for scheduling a single machine. *Journal of Scheduling*, 10 (6), 341–352.

Appendix A. Proof of Theorem 3.1

Before proceeding with the proof of Theorem 3.1, we first provide the necessary definitions and results from Gultekin *et al.* (2009). The time between loading machine i and the arrival time of the robot in front of the same machine to unload it is denoted as v_i . If the processing time on that machine exceeds this time epoch, then a waiting time will incur at that machine. In other words, the waiting time on machine i is defined as $w_i = \max\{0, P_i - v_i\}$. The cycle time of C_1^m is the total time required for all of the robot activities and the waiting times in front of the machines. A similar reasoning to the one in Gultekin *et al.* (2009) reveals the cycle time as follows:

$$T_{C_1^m} = 4m\epsilon + (2m^2 + 2m)\delta + w_1 + w_2 + \dots + w_m. \quad (\text{A1})$$

The robot travel time between consecutive machines (δ), the load/unload time of machines (ϵ), and the number of machines (m) are constant. Thus, we only have to find the total waiting time in front of the machines to calculate the cycle time. Different from the analysis in Gultekin *et al.* (2009), the processing times in each machine could be different. However, the processing time information is inherent in the waiting time values. Hence, the following system of equations, as derived in Gultekin *et al.* (2009) should be solved:

$$v_1 = T_{C_1^m} - [6\epsilon + (2m + 4)\delta + w_1 + w_m] = (4m - 6)\epsilon + (2m^2 - 4)\delta + w_2 + \dots + w_{m-1} \quad (\text{A2})$$

$$v_m = T_{C_1^m} - [6\epsilon + (2m + 4)\delta + w_m] = (4m - 6)\epsilon + (2m^2 - 4)\delta + w_1 + w_2 + \dots + w_{m-1}. \quad (\text{A3})$$

Finally,

$$v_i = T_{C_1^m} - [4\epsilon + (2m + 2)\delta + w_i] = (4m - 4)\epsilon + (2m^2 - 2)\delta + w_1 + \dots + w_m - w_i \quad (\text{A4})$$

for every $i \in \{2, \dots, m - 1\}$.

We first prove the following very strong property that waiting times can only occur on machines having the greatest processing time value among the set of machines $\{2, \dots, m - 1\}$. This property will considerably simplify our case analysis in the proof of Theorem 3.1.

Lemma A.1: For every machine $i \in \{2, \dots, m - 1\}$ such that $P_i < P_{\max}^1$, the waiting time value w_i is zero.

Proof: Assume to the contrary that $\exists i \in \{2, \dots, m - 1\}$ such that $P_i < P_{\max}^1$ but $w_i > 0$. Since $w_i > 0$, $P_i = v_i + w_i$. Moreover, using the set of equations (A4), $T_{C_1^m} = P_i + [4\epsilon + (2m + 2)\delta]$. Now, let $k \in \arg \max\{P_i : i \in \{2, \dots, m - 1\}\}$. There are two cases to consider.

Case (i) $w_k = 0$

In other words, $P_k \leq v_k$ and from equation set (A4), $v_k = T_{C_1^m} - [4\epsilon + (2m + 2)\delta]$. As derived above, $T_{C_1^m} - [4\epsilon + (2m + 2)\delta] = P_i$. However, all these conclusions force $P_{\max}^1 = P_k \leq P_i$ which is contradictory to our starting hypothesis.

Case (ii): $w_k > 0$

In other words, $w_k + v_k = P_k$ and from equation set (A4), $w_k + v_k = P_k = T_{C_1^m} - [4\epsilon + (2m + 2)\delta]$. However, this forces $P_{\max}^1 = P_k = P_i$ which is again contradictory to our starting hypothesis that $P_i < P_{\max}^1$. \square

With the following case analysis, we shall find the total waiting time as a function of the processing times.

- **If** $P_m > v_m$ or, equivalently, $w_m = P_m - v_m > 0$, then using Equation (A3), $v_m = P_m - w_m = (4m - 6)\epsilon + (2m^2 - 4)\delta + \sum_{i \neq m} w_i$, and therefore, $\sum_i w_i = P_m - [(4m - 6)\epsilon + (2m^2 - 4)\delta]$
- **Else If** $P_1 > v_1$, then using Equation (A2), $v_1 = P_1 - w_1 = (4m - 6)\epsilon + (2m^2 - 4)\delta + w_2 + \dots + w_{m-1}$ and since $w_m = 0$, $\sum_i w_i = P_1 - [(4m - 6)\epsilon + (2m^2 - 4)\delta]$
- **Else If** $P_i > v_i$ for some $i \in \{2, \dots, m - 1\}$, then we know from Lemma A.1 that $w_k > 0$ for any $k \in \arg \max\{P_i : i \in \{2, \dots, m - 1\}\}$. Since $w_1 = w_m = 0$ in this case, a reasoning similar to the previous cases results in $\sum_i w_i = P_{\max}^1 - [(4m - 4)\epsilon + (2m^2 - 2)\delta]$
- **Else** no waiting time occurs on any of the machines and hence $\sum_i w_i = 0$.

In the worst case, our total waiting time will correspond to the maximum possible of the above resulting four cases, and consequently, $w_1 + w_2 + \dots + w_m = \max\{0, P_1 - [(4m - 6)\epsilon + (2m^2 - 4)\delta], P_m - [(4m - 6)\epsilon + (2m^2 - 4)\delta], P_{\max}^1 - [(4m - 4)\epsilon + (2m^2 - 2)\delta]\}$ and the cycle time of C_1^m is obtained by replacing the total waiting time in Equation (A1) with this max function.

Appendix B. Proof of Theorem 3.2

In Gultekin *et al.* (2009), the cycle time of the second pure cycle C_2^m is derived as

$$T_{C_2^m} = 4m\epsilon + (2m^2 + 4m - 2)\delta + w_1 + w_2 + \dots + w_m. \tag{B1}$$

For this cycle, the v_i values are identical for all the machines and are

$$v_i = T_{C_2^m} - [4\epsilon + (2m + 2)\delta + w_i] = (4m - 4)\epsilon + 2(m - 1)(m + 2)\delta + w_1 + \dots + w_m - w_i. \tag{B2}$$

We first prove that if waiting time occurs on a machine then this machine has to be one of those with the greatest processing time.

Lemma B.1: *If $P_i < P_{\max}^2$, then $w_i = 0$ for any $i \in \{1, \dots, m\}$.*

Proof: Assume to the contrary that there exists $i \in \{1, \dots, m\}$ such that $P_i < P_{\max}^2$ but $w_i > 0$. Following a similar line of reasoning as done in the proof of Lemma A.1, we have $w_i + v_i = P_i = T_{C_2^m} - [4\epsilon + (2m + 2)\delta]$ using Equation (B2). Now, let $k \in \{1, \dots, m\}$ be a machine for which $P_k = P_{\max}^2$. There are two cases to consider.

Case (i) $w_k = 0$

In other words, $P_k \leq v_k$ and from equation set (B2), $v_k = T_{C_2^m} - [4\epsilon + (2m + 2)\delta]$. As derived above, $T_{C_2^m} - [4\epsilon + (2m + 2)\delta] = P_i$. However, all these conclusions force $P_{\max}^2 = P_k \leq P_i$ which is contradictory to our starting hypothesis.

Case (ii) $w_k > 0$

In other words, $w_k + v_k = P_k$ and from equation set (B2), $w_k + v_k = P_k = T_{C_2^m} - [4\epsilon + (2m + 2)\delta]$. Again, this forces $P_{\max}^2 = P_k = P_i$ which contradicts our starting hypothesis. \square

A shorter version of the case analysis done for the proof of Theorem 3.1 leads to the conclusion that, in the worst case, the total waiting time for this cycle, i.e. $\sum_i w_i$, is $\max\{0, P_{\max}^2 - (4m - 4)\epsilon - 2(m - 1)(m + 2)\delta\}$ and thus the theorem follows.