Available online at www.sciencedirect.com

SCIENCE DIRECT®

European Journal of Operational Research 174 (2006) 777–796

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

Production, Manufacturing and Logistics

Cyclic scheduling of a 2-machine robotic cell with tooling constraints

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

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

Received 29 September 2003; accepted 8 March 2005
Available online 16 June 2005

Abstract

In this study, we deal with the robotic cell scheduling problem with two machines and identical parts. In an ideal FMS, CNC machines are capable of performing all the required operations as long as the required tools are stored in their tool magazines. However, this assumption may be unrealistic at times since the tool magazines have limited capacity and in many practical instances the required number of tools exceeds this capacity. In this respect, our study assumes that some operations can only be processed on the first machine while some others can only be processed on the second machine due to tooling constraints. Remaining operations can be processed on either machine. The problem is to find the allocation of the remaining operations to the machines and the optimal robot move cycle that jointly minimize the cycle time. We prove that the optimal solution is either a 1-unit or a 2-unit robot move cycle and we present the regions of optimality. Finally, a sensitivity analysis on the results is conducted.

Keywords: Production scheduling; Automated systems; Robotic cell

1. Introduction

Efforts to improve productivity while maintaining the flexibility of conventional systems led to the development of Flexible Manufacturing Systems (FMS). The term FMS covers a range of systems from Flexible Manufacturing Cells (FMC) to Flexible Transfer Lines. An FMC is an integrated computer controlled system of a set of computer numerical control (CNC) machines and an automated material handling device. An FMC in which the material handling is done by a robot is called a robotic cell. A robotic cell with two machines consists of two machines located serially and served by a robot. In this study, consistent with the

* Corresponding author. Tel.: +90 312 266 4126; fax: +90 312 266 4054.
E-mail address: akturk@bilkent.edu.tr (M.S. Akturk).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2005.03.021

robotic cell scheduling literature, we will assume that each part being processed goes through the input buffer, the first machine, the second machine and finally the output buffer in that order. After loading a part to one of the machines, the robot either waits for the part to complete its processing or moves to unload another machine as soon as the processing of the part finishes (there is no waiting time to take a part from the input buffer or to drop a part to output buffer). A state of the system is defined by whether the robot and the machines are loaded or empty and by the location of the robot.

In the existing literature, the allocation of the operations to each machine is assumed to be constant and for given processing times the optimum robot move cycle minimizing the cycle time is to be determined. In some manufacturing operations such as chemical electroplating this assumption is meaningful and these operations mostly require no-wait constraints (see for example Dawande et al., 2003). However, in the case of machining operations for which the CNC machines are used, the allocation of the operations to the machines is also a decision problem. This is because CNC machines are capable of performing a wide range of operations with means of using different cutting tools. Therefore, assuming that processing times are fixed on each CNC machine may not accurately represent the capabilities of the CNC machines and limits the number of alternatives unnecessarily for these systems.

In this study, we consider a robotic cell of two identical CNC machines which repeatedly produces one type of product. We assume that there is an infinite number of parts to be processed, where a part is defined by a fixed set of operations to be performed in any order and each operation requires a unique type of cutting tool to be performed. That is, one tool can perform different operations but an operation can only be performed by a unique type of tool. We also assume that some tools are loaded only on the first machine and some others are loaded only on the second machine. A third set of tools are duplicated and loaded on both machines. As a result of this, an operation can either be processed only on the first machine, only on the second machine or on either machine. This assumption is valid since the tool magazines of these machines have limited capacity. Additionally, though duplicating the required tools increases flexibility, duplicating all of them may not be economically justifiable. Then the problem is not only sequencing the robot's activities but also partitioning the set of flexible operations into two machines. The objective is to minimize the cycle time which is defined as the long run average time required by the robotic cell to complete one part. More formally, we assume that we have infinite number of parts and if S_n denotes the completion time of the nth part then the long run average cycle time is $\limsup_{n\to\infty} S_n/n$ (Crama et al., 2000). Cyclic production in a robotic cell refers to the production of finished parts by repeating a fixed sequence of robot moves. As discussed in Dawande et al. (2003), the main motivation for studying cyclic production comes from practice: cyclic schedules are easy to implement and control and are the primary way of specifying the operation of a robotic cell in industry. Furthermore, Dawande et al. (2002a) show that for the problem of scheduling operations in bufferless robotic cells that produce identical parts (similar to our problem), it is sufficient to consider cyclic schedules in order to maximize throughput. They proved that there is at least one cyclic schedule in the set of all schedules that optimizes the throughput of the cell.

In the literature, there exist various problems concerning the robotic cells. An extensive survey of the robotic cell scheduling problems can be found in Crama et al. (2000) and Dawande et al. (2003). Robotic cells with no storage buffers at or between machines are widely used in practice (Miller and Walker, 1990), and as stated by Kise et al. (1991), small-scale flexible manufacturing cells with simple material handling devices are quite common in real applications. Since there can be several machines requesting services of the robot at once, there will be conflicts in which one machine will have to wait while another is served and if this waiting time is not handled properly, the overall throughput of the cell will decrease (King et al., 1993).

For 2-machines, no-buffer, identical parts robotic cell scheduling problem, Sethi et al. (1992) proved that the optimal solution is a 1-unit cycle where an n-unit cycle can be defined as, starting with an initial state of the system, loading and unloading all of the machines exactly n times and returning to the initial state of the system (note that in an n-unit cycle exactly n parts are produced). They also conjectured that *optimal* 1-*unit cycles are superior to every* n-*unit cycle*, *for* $n \geqslant 2$. 1-unit cycles are important since they are simple, easy to implement and control. Crama and van de Klundert (1997) considered the identical parts problem with m machines and showed that, considering only 1-unit cycles, the problem can be solved in (strongly) polynomial time. Hall et al. (1997) considered three machine cells producing single part-types and proved that the repetition of 1-unit cycles dominates more complicated policies that produce two units. The validity of the conjecture of Sethi et al. (1992) for 3-machine robotic flowshops is established by Crama and van de Klundert (1999). Brauner and Finke (1997) disproved this conjecture for cells of size four or large.

Additionally, there are many studies considering the multiple parts robotic cell scheduling problem in which the objective is to find the optimal robot move sequence as well as the optimal part input sequence that jointly minimize the cycle time. The complexity of these problems can be found in Sriskandarajah et al. (1998) and Hall et al. (1998).

With the settings of this study, we prove that the optimal solution is not necessarily a 1-unit cycle as in the case of Sethi et al. (1992) and that a 2-unit cycle can also give the optimal cycle time for some parameter values. We also show that the problem considered by Sethi et al. (1992) is a special case of the problem under consideration in this study. We report some sensitivity analysis results on the parameters such as: the loading and unloading time of the robot and the transportation time of the robot.

The remainder of the paper is organized as follows. In the next section we introduce our problem and present the notations, definitions and the assumptions pertaining to this study. The solution of the operation allocation problem with tooling constraints is presented in Section 3 and the last section is devoted to the concluding remarks.

2. Problem definition

In this section, we will state our problem more formally and provide the required definitions and notations. We will also highlight the difference of the considered problem from the earlier studies with the aid of an example.

The workstations in a robotic cell for machining operations are CNC machines and in these machines the processing of a part requires performing a series of operations using different sets of cutting tools. The cutting tools are stored in the tool magazines of these machines. In an ideal FMS, each machine is capable of performing all operations of all parts scheduled for production as long as it has the required tools in its tool magazine. However, Gray et al. (1993) observe that a CNC has a limited tool magazine capacity and the total set of tools required to process all jobs usually exceeds this capacity. Furthermore, duplicating all the required tools and loading them to each tool magazine may not be economically justifiable due to high tool investment costs. Therefore, we assume that there is a single copy of some tools. A subset of these single copies are loaded on the first machine and the remaining ones are loaded on the second machine. On the other hand, some tools are duplicated and loaded on both machines. As a consequence, each part to be processed has three sets of operations. O_1 is the set of operations that can only be processed on the first machine, O_2 is the set of operations that can only be processed on the second machine, and O is the set of operations that can be processed on either machine. The allocation of the operations that are in set O to the machines can be modified for each individual part. We assume that we have an infinite number of identical parts to be processed on these two machines and each part is defined by a fixed set of operations to be performed in any order. So the problem is to find the optimal robot move cycle with the corresponding allocation of operations that are in set O to the machines in order to minimize the cycle time. In fact, allocation of the operations to the machines is equivalent to partitioning set O into two subsets so that, the operations that are in the first subset will be processed on the first machine while the operations in the other subset will be processed on the second machine.

The following definitions well accepted in the literature are borrowed from Crama and van de Klundert (1997).

Definition 1. An *n-unit robot move cycle* is a sequence of robot moves during which each machine is loaded and unloaded exactly n times, and finally the cell returns to its initial state.

Definition 2. Robot activity A_i corresponds to the following sequence of movements: unload a part from machine i, transport it to machine $i+1$, and load machine $i+1$.

According to Definition 2, in a 2-machine robotic cell where the machines are numbered as 1 and 2, the input buffer is numbered as 0 and the output buffer is numbered as 3, we have exactly three robot activities: A_0, A_1 and A_2.

Since in an optimal cycle we require that the robot move path is as short as possible, any two consecutive activities uniquely determine the robot moves between those activities. Therefore, any robot move cycle can be uniquely described by a permutation of the above activities. For two machines, we have two 1-unit robot move cycles: S_1: $A_0A_1A_2$ and S_2: $A_2A_1A_0$. Let us recall that a state of the system is defined by whether the robot and the machines are loaded or empty and by the location of the robot. These two 1-unit cycles have one common state in which the first machine is empty and the second machine has just been loaded by the robot. Thus, a transition from one of these cycles to the other can only be made at this common state. If the robot waits in front of the second machine to finish the processing of the part, then the robot follows the activities of the S_1 cycle. Else if the robot travels to input buffer to take a part and load the first machine, then the robot follows the activities of the S_2 cycle. During these transitions no extra movements are made, thus no loss occurs in terms of robot transportation time.

Hall et al. (1997) define two other robot move sequences to represent higher unit robot move cycles. These are the transition movements of the robot from performing cycle S_1 to S_2 (represented as S_{12}) and S_2 to S_1 (represented as S_{21}). If a part is produced according to the S_1 (S_2) cycle in the first machine and according to the S_2 (S_1) cycle in the second machine then the transition movement S_{12} (S_{21}) is made. To properly clarify these new robot move sequences, we shall use the terminology of Dawande et al. (2002b). *Full waiting* is defined as the robot which has just loaded a machine waiting in front of a machine through the whole processing time of a part. On the other hand, *partial waiting* is the waiting time from the arrival of the robot at the machine till the processing of the part completes at this machine. Now, we are ready to list the S_{12} robot movements which can be described as $A_0A_1A_0$: (i) load the first machine and perform a full wait, (ii) load the second machine and immediately return to input buffer, (iii) load the first machine, and (iv) while processing continues on the first machine, return to the second machine and perform a partial wait.

In a similar fashion, in S_{21} movement which can be described as $A_2A_1A_2$, the robot does not wait in front of the first machine for the part but instead travels to the second machine, unloads it and drops the part to the output buffer and returns back to the first machine to unload the part. That is, a partial waiting on the first machine and a full waiting on the second machine are encountered.

Further analysis of the sequences yields the following: (i) an execution of S_1 starts and ends with empty machines, (ii) an execution of S_2 starts and ends with loaded machines, (iii) an execution of S_{12} starts with empty machines and ends with full machines, and (iv) an execution of S_{21} starts with loaded machines and ends with empty machines. Based on these observations, the transition digraph in Fig. 1 depicts the feasible transitions among sequences.

Hall et al. (1997) showed that any robot move cycle can be represented by the four robot move sequences: S_1, S_2, S_{12} and S_{21}. For example, $S_{12}S_{21}$ is a 2-unit robot move cycle, actually it is the only 2-unit robot move cycle and we can describe this cycle by the robot activity sequence: $A_0A_1A_0A_2A_1A_2$. $S_{12}S_{21}$ and $S_{21}S_{12}$ represent the same robot move cycle, the only difference being the starting state of the cycle (the animated views of these robot move cycles can be found at the web site http://www.ie.bilkent.edu.tr/~robot).

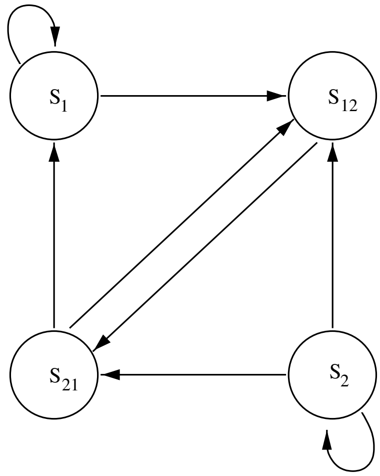

Fig. 1. Transition digraph.

We will use the following parameters and decision variables:

t_i	processing time of operation i for $i \in O_1 \cup O_2 \cup O$. Note that the processing times of operation i on both machines are equal for any i in O
P^{M1}	total processing time of the operations that are in set O_1, $P^{M1} = \sum_{i \in O_1} t_i$
P^{M2}	total processing time of the operations that are in set O_2, $P^{M2} = \sum_{i \in O_2} t_i$
P	total processing time of the operations that are to be allocated to either one of the machines, $P = \sum_{i \in O} t_i$
ϵ	the load (or unload) time of workstations by the robot
δ	time taken by the robot to travel between two consecutive stations. We assume that the robot travel time from machine i to j is $\lvert j - i \rvert\delta$. So the triangular equality is satisfied. For example, the transportation time from input buffer to the second machine is $\lvert 2 - 0\rvert\delta = 2\delta$ and the transportation time from output buffer to input buffer is $\lvert 0 - 3\rvert\delta = 3\delta$
T	long run average cycle time to produce one part, a decision variable of our problem

Note that if set O is empty, that is when $P = 0$, there is no allocation problem. As a consequence, the problem becomes identical with the problem considered by Sethi et al. (1992). Another important observation is that the order in which the operations are processed on the machines does not have an effect on the cycle time.

In order to calculate the cycle time of any given robot move cycle, the processing times of parts on each machine must be known in advance. However, in order to find the processing times of the parts on the machines, we have to know the allocation of the operations in set O to the machines. Furthermore, the allocation of the operations to the machines need not be fixed for each part to be processed. That is, for one part, O can be partitioned as Z_1 and $(O \backslash Z_1)$ and for the next part to be processed it can be partitioned as Z_2 and $(O \backslash Z_2)$ where $Z_1 \neq Z_2$. So the processing time of the first part on the first machine is $\sum_{i \in O_1 \cup Z_1} t_i$ while the processing time of the second part on the same machine is $\sum_{i \in O_1 \cup Z_2} t_i$. The processing times of the first and second parts on the second machine are $\sum_{i \in O_2 \cup (O \backslash Z_1)} t_i$ and $\sum_{i \in O_2 \cup (O \backslash Z_2)} t_i$, respectively. Thus the processing times of the machines may change from one repetition of the robot move cycle to the other. In fact, using different allocations for each part to be processed on these two machines may yield a shorter cycle time than the case when we have a fixed allocation type. In what follows, we give an example of such a situation for the robot move cycle S_2.

Example 1. Let us consider the robot move cycle S_2. Assume that a part has five operations to be processed with the corresponding operation times: $t_1 = 15$, $t_2 = 30$, $t_3 = 45$, $t_4 = 10$, and $t_5 = 30$. Also let $\epsilon = 5$ and $\delta = 10$. Furthermore, assume that because of the tooling constraints, operation 3 must be processed on the first machine and operation 5 must be processed on the second machine. Thus, $P^{M1} = t_3 = 45$, $P^{M2} = t_5 = 30$ and $P = t_1 + t_2 + t_4 = 55$. Sethi et al. (1992) proved that the cycle time of S_2 for fixed processing time, i.e., same allocation of the operations for all parts, is:

$$T = 6\epsilon + 8\delta + \max\{0, a - (2\epsilon + 4\delta), b - (2\epsilon + 4\delta)\},$$

where a and b are the processing times of the part on the first and second machine respectively, i.e., $a = P^{M1} + a'$ and $b = P^{M2} + P - a'$ where a' takes some value in $\{0, 10, 15, 25, 30, 40, 45, 55\}$. Since $\max\{P^{M1} + a', P^{M2} + P - a'\} \geqslant 2\epsilon + 4\delta = 50$ the max term in the description of T will take a nonnegative value and hence will be minimized when $|a - b|$ is minimized. With the given operation times this happens if operation 1 is allocated to the first machine and operations 2 and 4 are allocated to the second machine. So $a = P^{M1} + t_1 = 60$, $b = P^{M2} + t_2 + t_4 = 70$ and $|a - b| = 10$. Then the cycle time becomes 130 (another alternative is to allocate operations 1 and 4 to the first machine and operation 2 to the second machine). On the other hand, let us now assume that for the first part to be processed we allocated the operations as before so that we have $a_1 = P^{M1} + t_1 = 60$ and $b_1 = P^{M2} + t_2 + t_4 = 70$, where a_1 and b_1 represent the processing time of the first part on the first and the second machines, respectively. However, assume that for the second part we allocated such that $a_2 = P^{M1} + t_1 + t_4 = 70$ and $b_2 = P^{M2} + t_2 = 60$ where a_2 and b_2 are the respective processing times of the second part on the two machines. Assume further that these two allocation types are used alternately in processing the parts. Then, in one repetition of the cycle, a part with the first allocation type will be processed on the first machine and a part with the second allocation type will be processed on the second machine. This is because, during one repetition of the cycle S_2, two different parts become available in the system; one new part enters the system and one previously entered part exits the system. In the second repetition of S_2, a part with the second allocation type will be processed on the first machine and a part with the first allocation type will be processed on the second machine because we use these two allocation types alternatingly. Then, the long run average cycle time to produce one part can be found as follows: Since we consider cyclic robot moves, let the cycle start with the state in which the second machine is loaded by a part with the first allocation type and the robot is in front of the input buffer to take a part with the second allocation type. In order to find the long run average time to produce one part, we have to produce two parts with different allocation types and take the average. This means repeating the S_2 cycle two times. Let w_{ij} be the partial waiting time in front of machine i, $i = 1, 2$, for the part with allocation type j, $j = 1, 2$. The total loading/unloading and travel times are the same for both repetitions of the S_2 cycle, namely,$6\epsilon + 8\delta$. In the first repetition of the cycle there is partial waiting in front of machine 2 for the part with the first allocation type, w_{21}, and another partial wait in front of the first machine for the part with the second allocation type, w_{12}. On the other hand, in the second repetition we have w_{11} and w_{22}. One can easily conclude that $w_{11} = \max\{0, a_1 - (2\epsilon + 4\delta + w_{22})\}$, $w_{12} = \max\{0, a_2 - (2\epsilon + 4\delta + w_{21})\}$, $w_{21} = \max\{0, b_1 - (2\epsilon + 4\delta)\}$ and $w_{22} = \max\{0, b_2 - (2\epsilon + 4\delta)\}$. From these,

$$w_{12} + w_{21} = \max\{0, a_2 - (2\epsilon + 4\delta), b_1 - (2\epsilon + 4\delta)\},$$
$$w_{11} + w_{22} = \max\{0, a_1 - (2\epsilon + 4\delta), b_2 - (2\epsilon + 4\delta)\}.$$

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

$$T = 6\epsilon + 8\delta + \frac{\max\{0, a_1 - (2\epsilon + 4\delta), b_2 - (2\epsilon + 4\delta)\}}{2} + \frac{\max\{0, a_2 - (2\epsilon + 4\delta), b_1 - (2\epsilon + 4\delta)\}}{2}$$

and with the given data this boils down to $T = 125$.

This example shows that fixing the allocation type of every part limits the better of available alternatives. Note that finding the allocation of the operations to the machines even if the robot move cycle is fixed is not an easy task. This will be further explored in Section 3.

We will refer to the following definitions throughout this paper especially when finding the cycle times of the robot move cycles with differing part allocation types.

Definition 3. In a k-*allocation* type production, the allocation of every kth part in the infinite sequence is identical and k is minimal with this property. That is, k is the period of the allocation.

Note that a 2-unit robot move cycle with one-allocation type means that all parts to be processed have the same allocation of operations to the machines. A 2-unit cycle with two-allocation types means that two identical parts with different allocation of operations to the machines are produced alternatingly. The cycle time for this case is found by calculating the total time to finish one repetition of the cycle and dividing by two. On the other hand, a 2-unit cycle with three-allocation types means that the allocation of operations for parts $1, 4, \ldots, 3z+1$ are the same. This also holds for parts $2, 5, \ldots, 3z+2$ and parts $3, 6, \ldots, 3z$. In order to find the cycle time of an n-unit cycle with k-allocation types, we have to find the least common multiple of n and k; let this number be M. To find the average time to produce one part, we have to repeat the cycle M/n times and divide the total time by M. In particular, for a 2-unit cycle with three-allocation types, we have to repeat the cycle 3 times and divide the total time by $M = 6$.

Note also that in a k-allocation type production for a specific k, the allocation of the operations is not unique but rather there exist finitely many different allocations. Let σ_k be a specific allocation of operations for a k-allocation type production and σ_k^* be the optimal allocation of operations for a k-allocation type production. We further need the following definition for the processing times of the parts on both machines.

Definition 4. Consider any robot move cycle using a k-allocation type production. Let σ_k

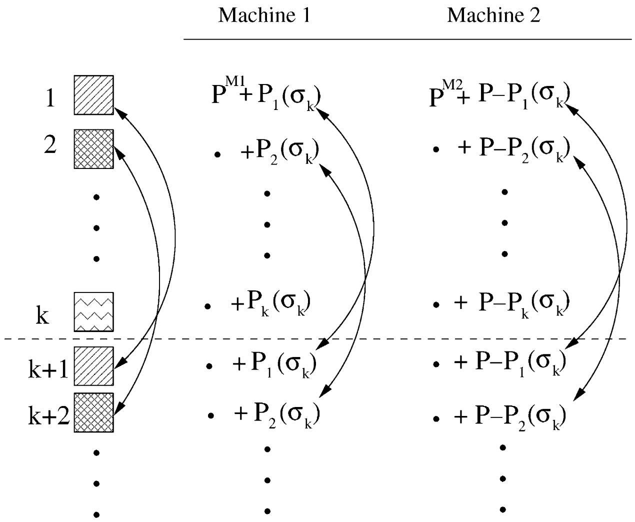

Fig. 2. Processing times on both machines for k-allocation types.

for each of the three robot move cycles and Section 3.3 is devoted to the sensitivity analysis on problem parameters.

3.1. Optimal allocation of operations

In this section we will first observe that there is no allocation problem for S_1. Theorems 1 and 3 will determine the optimal k to be used with the cycles S_2 and $S_{12}S_{21}$ respectively. In Theorem 2 we will prove that determining the optimal allocation of operations to the machines is NP-complete for robot move cycle S_2. We shall assume $P^{M1} \geqslant P^{M2}$ throughout all proofs. The other case can be treated in a similar fashion.

The cycle times of the three robot move cycles discussed previously are presented in Appendix A. The cycle time of S_1, given in Eq. (A.1) does not depend on the processing times of the parts on the machines individually but only depends on the total processing time, in our case $P + P^{M1} + P^{M2}$. Thus, whatever the allocation of the operations be, the cycle time is the same and hence there is no allocation problem for S_1.

The following theorem provides the optimal k to be used with S_2:

Theorem 1. *Consider a cyclic production performing the* 1-*unit cycle* S_2. *We have*:

1. *If* $P^{M1} \geqslant P + P^{M2}$, *then using one-allocation is optimal*,
2. *Otherwise*, *using either one-allocation or two-allocation is optimal.*

Proof

1. The total time to complete production of all k parts with k-allocation type for cycle S_2 under specific allocation of operations σ_k, is given in Appendix A, Eq. (A.4). Since $P^{M1} \geqslant P + P^{M2}$, then $P^{M1} + P_j(\sigma_k) \geqslant P^{M2} + P - P_{(j-1)}(\sigma_k),\ \forall j \in [1, \ldots, k]$ where for notational purposes we take $P_0(\sigma_k) \equiv P_k(\sigma_k)$. Therefore, under optimal allocation σ_k^*, we have $P_j(\sigma_k^*) = 0,\ \forall j$. However, this is nothing but one-allocation type and the optimal cycle time is:

$$T_{S_2(\mathrm{opt})} = 6\epsilon + 8\delta + \max\{0, P^{M1} - (2\epsilon + 4\delta)\}.$$

2. Consider again (A.4). Since $P^{M1} < P + P^{M2}$, under optimal allocation σ_k^*, each max term will individually be minimized when the last two nonzero components are as close as possible, i.e., when $P^{M1} + P_j(\sigma_k^*) \approx P + P^{M2} - P_{(j-1)}(\sigma_k^*),\ \forall j$. Consider two such consecutive relations, say for jth and $(j+1)$th max terms:

$$P^{M1} + P_j(\sigma_k^*) \approx P + P^{M2} - P_{(j-1)}(\sigma_k^*), \tag{1}$$

$$P^{M1} + P_{(j+1)}(\sigma_k^*) \approx P + P^{M2} - P_j(\sigma_k^*). \tag{2}$$

From relations (1) and (2), if the equalities cannot be satisfied, then the difference between both sides of each equality must be minimized. That is, both $|P^{M1} + P_j(\sigma_k^*) - (P + P^{M2} - P_{(j-1)}(\sigma_k^*))|$ and $|P^{M1} + P_{(j+1)}(\sigma_k^*) - (P + P^{M2} - P_j(\sigma_k^*))|$ must be minimized. Note however that if $P_j(\sigma_k^*)$ and $P_{(j-1)}(\sigma_k^*)$ minimize the first difference, then $P_j(\sigma_k^*)$ and $P_{(j+1)}(\sigma_k^*) = P_{(j-1)}(\sigma_k^*)$ simultaneously minimize the second difference. This yields either a one-allocation ($P_{(j-1)}(\sigma_k^*) = P_j(\sigma_k^*) = P_{(j+1)}(\sigma_k^*)$ for all j) or a two-allocation ($P_{(j-1)}(\sigma_k^*) = P_{(j+1)}(\sigma_k^*)$ and $P_j(\sigma_k^*) \neq P_{j-1}(\sigma_k^*)$ for all j). $\square$

Though Theorem 1 guides us in selecting the optimal allocation type in a cyclic production S_2, finding the optimal allocation of operations to the two machines is not an easy job even when there is a fixed allocation type and even when $P^{M1} = P^{M2} = 0$. More formally, we shall now show that the following decision problem is NP-complete.

S_2 operation allocation for one-allocation type decision problem (*problem S2TAP*):

Instance: A finite set of operations O with respective integer processing times $\{t_1, \ldots, t_p\}$, loading/unloading time ϵ, transportation time δ and a real number K.
Question: Can we find an allocation of operations to the two machines, σ_1, so that the long run average cycle time $T_{S_2(\sigma_1)} \leqslant K$?

Theorem 2. *Problem S2TAP is NP-complete.*

Proof. S2TAP is in NP since whenever we are given a specific allocation of operations, we can readily find the corresponding long run average cycle time and hence decide if it is less than or equal to K. It is apparent from expression (A.2) for $T_{S_2(\sigma_1)}$ (see Appendix A) that the well known NP-complete problem 2-partition (see Garey and Johnson, 1976) readily reduces to S2TAP. $\square$

The following theorem determines the optimal k to be used with $S_{12}S_{21}$.

Theorem 3. *For 2-unit robot move cycle $S_{12}S_{21}$, using two allocation types is optimal.*

Proof. In order to prove this theorem, we will first compare the cycle time of two-allocation type for the cycle $S_{12}S_{21}$ with the cycle time of one-allocation type for the cycle $S_{12}S_{21}$. The long run average cycle times of one-allocation and two-allocation types for the cycle $S_{12}S_{21}$ are given in Eqs. (A.5) and (A.6), respectively.

We first argue that the optimal allocation for two-allocation type for the cycle $S_{12}S_{21}$ is found by letting $P_1(\sigma_2^*) = (P - P_2(\sigma_2^*)) = 0$. Let us first rewrite Eq. (A.6) by entering $\frac{1}{2}(P_1(\sigma_2) + (P - P_2(\sigma_2)))$ into the max term, i.e., adding it to all the three arguments of max. This leads us to the following form:

$$\begin{aligned} T_{S_{12}S_{21}(\sigma_2)} = {} & 1/2(12\epsilon + 14\delta + P^{M1} + P^{M2}) + 1/2(\max\{P_1(\sigma_2) + P - P_2(\sigma_2), P^{M1} + P_1(\sigma_2) \\ & + P - (2\epsilon + 4\delta), P^{M2} + 2P - P_2(\sigma_2) - (2\epsilon + 4\delta)\}). \end{aligned}$$

Now, $P_2(\sigma_2)$ only appears with a negative coefficient and $P_1(\sigma_2)$ only appears with a positive coefficient. To minimize, we must increase $P_2(\sigma_2)$ and decrease $P_1(\sigma_2)$ accordingly. So, we take $P_1(\sigma_2^*) = 0$ and $P_2(\sigma_2^*) = P$. Since we also have $P^{M1} \geqslant P^{M2}$ the cycle time corresponding to this allocation type becomes:

$$T_{S_{12}S_{21}(\sigma_2^*)} = \frac{12\epsilon + 14\delta + P^{M1} + P^{M2} + \max\{0, (P + P^{M1}) - (2\epsilon + 4\delta)\}}{2}. \tag{3}$$

Let us move P in Eq. (A.5) inside the max term. Thus we have:

$$\begin{aligned} T_{S_{12}S_{21}(\sigma_1^*)} = {} & 1/2(12\epsilon + 14\delta + P^{M1} + P^{M2}) + 1/2(\max\{P, P + P^{M1} + P_1(\sigma_1^*) \\ & - (2\epsilon + 4\delta), P + P^{M2} + (P - P_1(\sigma_1^*)) - (2\epsilon + 4\delta)\}). \end{aligned}$$

Comparing this with Eq. (3), it is easily seen that $T_{S_{12}S_{21}(\sigma_2)} \leqslant T_{S_{12}S_{21}(\sigma_1)}$.

Now let us consider the cycle time for k-allocation ($k > 2$) type for the cycle $S_{12}S_{21}$. Observe that, in robot move cycle $S_{12}S_{21}$, initially, the machines are empty and the robot is waiting in front of the input buffer. Since this is a 2-unit cycle, exactly two parts are loaded and unloaded to each machine and at the end, identical to the initial state, the machines are again empty and the robot is waiting in front of the input buffer. Then, in a k-allocation type for the cycle $S_{12}S_{21}$, this 2-unit cycle is repeated exactly $k/2$ times to produce all k parts with different allocation types. The objective is to find the optimal allocation for these parts to the machines. However, the optimal allocation at each repetition of the cycle must be the same as the optimal allocation of the operations for a unique cycle $S_{12}S_{21}$. This is because, a k-allocation type is a concatenation of $k/2$ two-allocation types for the cycle $S_{12}S_{21}$. Thus, we conclude that the optimal cycle time for a k-allocation type for the cycle $S_{12}S_{21}$ is the same as the optimal cycle time for a two-allocation type for the cycle $S_{12}S_{21}$. □

Now, we are ready to provide one of the major results of our paper which will restrict our search for the optimal cycle to three robot move cycles. We first recall a result due to Hall et al. (1997).

Lemma 1 (Hall et al., 1997). *In any feasible robot move cycle, the number of S_{12} sequences is equal to the number of S_{21} sequences.*

Theorem 4. *At least one of the cycles S_1, S_2 or $S_{12}S_{21}$ has a cycle time that is less than or equal to the cycle time of any given n-unit robot move cycle.*

For the clarity of the presentation, this proof is deferred to Appendix B.

In summary, we have three potentially optimal robot move cycles. For the robot move cycle S_2, we have an allocation problem. However, we know from Theorem 1 that if $P^{M1} \geqslant P + P^{M2}$, then the allocation problem disappears and we have $T_{S_2(\text{opt})} = 6\epsilon + 8\delta + \max\{0, P^{M1} - (2\epsilon + 4\delta)\}$. Else, we can find lower and upper bounds for this cycle time. We get a lower bound when it is possible to have $P^{M1} + P_j(\sigma_2) = P^{M2} + (P - P_j(\sigma_2))$, $j \in [1, 2]$. Since any feasible solution provides an upper bound for the optimal cycle time, we will present a feasible solution that can be attained with any given problem instance and use this feasible solution as an upper bound. For a given problem instance, let us allocate all of the operations that are in set O to the first machine for the first part and to the second machine for the second part. Note that such an allocation is feasible with any given problem parameter values. Thus, $P_1(\sigma_2) = (P - P_2(\sigma_2)) = P$ in Eq. (A.3). Let $\underline{T_{S_2}}$ represent the lower bound of cycle S_2 and $\overline{T_{S_2}}$ represent the upper bound. In other words:

$$\underline{T_{S_2}} = 6\epsilon + 8\delta + \max\{0, (P + P^{M1} + P^{M2})/2 - (2\epsilon + 4\delta)\},$$

$$\begin{aligned} \overline{T_{S_2}} = {} & 6\epsilon + 8\delta + \frac{\max\{0, P + P^{M1} - (2\epsilon + 4\delta), P + P^{M2} - (2\epsilon + 4\delta)\}}{2} \\ & + \frac{\max\{0, P^{M1} - (2\epsilon + 4\delta), P^{M2} - (2\epsilon + 4\delta)\}}{2}. \end{aligned}$$

Since we assumed that $P^{M1} \geqslant P^{M2}$ the upper bound becomes:

$$\overline{T_{S_2}} = 6\epsilon + 8\delta + \frac{1}{2}\max\{0, P + P^{M1} - (2\epsilon + 4\delta)\} + \frac{1}{2}\max\{0, P^{M1} - (2\epsilon + 4\delta)\}.$$

Now we are ready to present the optimal S_2 cycle times along with lower and upper bounds based on a set of breakpoints partitioning the search space.

(1) If $P^{M1} \geqslant P + P^{M2}$, then
 (i) If $P^{M1} \geqslant 2\epsilon + 4\delta$, then

$$T_{S_2(\text{opt})} = 4\epsilon + 4\delta + P^{M1}. \tag{4a}$$

 (ii) Otherwise,

$$T_{S_2(\text{opt})} = 6\epsilon + 8\delta. \tag{4b}$$

(2) Else (i.e. $P^{M1} < P + P^{M2}$) we have lower and upper bounds. The lower bounds are as follows:
 (i) If $(P + P^{M1} + P^{M2})/2 \geqslant 2\epsilon + 4\delta$, then

$$\underline{T_{S_2}} = 4\epsilon + 4\delta + (P + P^{M1} + P^{M2})/2. \tag{4c}$$

 (ii) Otherwise,

$$\underline{T_{S_2}} = 6\epsilon + 8\delta. \tag{4d}$$

The upper bounds are as follows:

 (i) If $P^{M1} \geqslant 2\epsilon + 4\delta$, then

$$\overline{T_{S_2}} = 4\epsilon + 4\delta + P^{M1} + P/2. \tag{4e}$$

 (ii) Else if $P + P^{M1} < 2\epsilon + 4\delta$, then

$$\overline{T_{S_2}} = 6\epsilon + 8\delta. \tag{4f}$$

 (iii) Else,

$$\overline{T_{S_2}} = 5\epsilon + 6\delta + (P + P^{M1})/2. \tag{4g}$$

On the other hand, the cycle time of two-allocation type for the cycle $S_{12}S_{21}$ is given in Eq. (A.6). Since $P^{M1} \geqslant P^{M2}$ and employing Theorem 3, we get the following cycle time:

$$T_{S_{12}S_{21}(\text{opt})} = 6\epsilon + 7\delta + \frac{(P^{M1} + P^{M2})}{2} + \frac{1}{2}\max\{0, (P^{M1} + P) - (2\epsilon + 4\delta)\}.$$

Therefore, we have the following breakpoints for this cycle time:

(1) If $P + P^{M1} \geqslant 2\epsilon + 4\delta$ then,

$$T_{S_{12}S_{21}(\text{opt})} = 5\epsilon + 5\delta + P^{M1} + \frac{(P + P^{M2})}{2}. \tag{5a}$$

(2) Else,

$$T_{S_{12}S_{21}(\text{opt})} = 6\epsilon + 7\delta + \frac{(P^{M1} + P^{M2})}{2}. \tag{5b}$$

3.2. Regions of optimality

In the sequel, we will prove a group of lemmas which will jointly lead to Theorem 5 presenting the regions of optimality for these three robot move cycles.

Lemma 2. *If $P^{M1} + P^{M2} \geqslant 2\delta$, then S_2 gives the minimum cycle time.*

Proof. Assume $P^{M1} + P^{M2} \geqslant 2\delta$. Let us first compare T_{S_1} and $T_{S_{12}S_{21}(\text{opt})}$. T_{S_1} is as given in Eq. (A.1).

1. If $P + P^{M1} \geqslant 2\epsilon + 4\delta$, $T_{S_{12}S_{21}(\text{opt})}$ is given in Eq. (5a). A simple comparison yields $T_{S_{12}S_{21}(\text{opt})} < T_{S_1}$.
2. Otherwise, $T_{S_{12}S_{21}(\text{opt})}$ is given by Eq. (5b). Then we have the following:

$$T_{S_1} = 6\epsilon + 6\delta + P + P^{M1} + P^{M2} \geqslant 6\epsilon + 6\delta + P + \frac{(P^{M1} + P^{M2})}{2} + \frac{2\delta}{2} \geqslant T_{S_{12}S_{21}(\text{opt})} \Rightarrow T_{S_{12}S_{21}(\text{opt})} \leqslant T_{S_1}.$$

We will now compare $T_{S_2(\text{opt})}$ with $T_{S_{12}S_{21}(\text{opt})}$.

(1) If $P^{M1} \geqslant P + P^{M2}$, then
 (1.1) If $P^{M1} \geqslant 2\epsilon + 4\delta$, this implies that $P + P^{M1} \geqslant 2\epsilon + 4\delta$. Then, $T_{S_2(\text{opt})}$ and $T_{S_{12}S_{21}(\text{opt})}$ are given by Eqs. (4a) and (5a) respectively. Hence, we conclude that in this region $T_{S_2(\text{opt})} < T_{S_{12}S_{21}(\text{opt})}$.
 (1.2) If $P^{M1} < 2\epsilon + 4\delta$ and $P + P^{M1} \geqslant 2\epsilon + 4\delta$, then $T_{S_{12}S_{21}(\text{opt})}$ is the same as above. $T_{S_2(\text{opt})}$ in this region is given in Eq. (4b). Since $P + P^{M1} \geqslant 2\epsilon + 4\delta$ and $P^{M1} + P^{M2} \geqslant 2\delta$ we have:

$$T_{S_{12}S_{21}(\text{opt})} = 5\epsilon + 5\delta + \frac{P + P^{M1} + P^{M1} + P^{M2}}{2} \geqslant 5\epsilon + 5\delta + \frac{2\epsilon + 4\delta + 2\delta}{2} = 6\epsilon + 8\delta = T_{S_2(\text{opt})}$$
$$\Rightarrow T_{S_2(\text{opt})} \leqslant T_{S_{12}S_{21}(\text{opt})}.$$

 (1.3) If $P + P^{M1} < 2\epsilon + 4\delta$, $T_{S_2(\text{opt})}$ and $T_{S_{12}S_{21}(\text{opt})}$ are presented in Eqs. (4b) and (5b), respectively. Since $P^{M1} + P^{M2} \geqslant 2\delta$ we have:

$$T_{S_{12}S_{21}(\text{opt})} = 6\epsilon + 7\delta + \frac{(P^{M1} + P^{M2})}{2} \geqslant 6\epsilon + 8\delta = T_{S_2(\text{opt})} \Rightarrow T_{S_2(\text{opt})} \leqslant T_{S_{12}S_{21}(\text{opt})}.$$

(2) Else if $P^{M1} < P + P^{M2}$, then we have upper and lower bounds for the cycle time of S_2. If we can show that $\overline{T_{S_2}} \leqslant T_{S_{12}S_{21}(\text{opt})}$, then we can conclude that $T_{S_2(\text{opt})} \leqslant T_{S_{12}S_{21}(\text{opt})}$. We have the following cases:
 (2.1) If $P^{M1} \geqslant 2\epsilon + 4\delta$, this implies that $P + P^{M1} \geqslant 2\epsilon + 4\delta$. $\overline{T_{S_2}}$ and $T_{S_{12}S_{21}(\text{opt})}$ are given in Eqs. (4e) and (5a), respectively. We conclude easily that $T_{S_2(\text{opt})} \leqslant \overline{T_{S_2}} < T_{S_{12}S_{21}(\text{opt})}$.
 (2.2) If $P^{M1} < 2\epsilon + 4\delta$ and $P + P^{M1} \geqslant 2\epsilon + 4\delta$, $\overline{T_{S_2}}$ and $T_{S_{12}S_{21}(\text{opt})}$ are given in Eqs. (4g) and (5a), respectively. Since $P^{M1} + P^{M2} \geqslant 2\delta$, we have:

$$T_{S_{12}S_{21}(\text{opt})} = 5\epsilon + 5\delta + \frac{P + P^{M1} + P^{M1} + P^{M2}}{2} \geqslant 5\epsilon + 5\delta + \frac{P + P^{M1} + 2\delta}{2} = 5\epsilon + 6\delta + \frac{(P + P^{M1})}{2}$$
$$\Rightarrow T_{S_2(\text{opt})} \leqslant \overline{T_{S_2}} \leqslant T_{S_{12}S_{21}(\text{opt})}.$$

 (2.3) If $P + P^{M1} < 2\epsilon + 4\delta$, this implies that $P^{M1} < 2\epsilon + 4\delta$. For this case, $\overline{T_{S_2}}$ and $T_{S_{12}S_{21}(\text{opt})}$ are given in Eqs. (4f) and (5b), respectively. Since $P^{M1} + P^{M2} \geqslant 2\delta$ we conclude easily that for this case also $T_{S_2(\text{opt})} \leqslant \overline{T_{S_2}} \leqslant T_{S_{12}S_{21}(\text{opt})}$.

Since in all of the cases $\overline{T_{S_2}} \leqslant T_{S_{12}S_{21}(\text{opt})}$, we conclude that S_2 has the minimum cycle time. □

Lemma 3. *If* $2P + P^{M1} + P^{M2} \leqslant 2\delta$, *then* S_1 *gives the minimum cycle time.*

Proof. $2P + P^{M1} + P^{M2} \leqslant 2\delta \Rightarrow P + P^{M1} < 2\epsilon + 4\delta$. $T_{S_{12}S_{21}(\text{opt})}$ is given in Eq. (5b). When we compare this with T_{S_1} given in Eq. (A.1), since $2P + P^{M1} + P^{M2} \leqslant 2\delta$, we have:

$$T_{S_1} = 6\epsilon + 6\delta + P + P^{M1} + P^{M2} = 6\epsilon + 6\delta + \frac{2P + P^{M1} + P^{M2} + P^{M1} + P^{M2}}{2} \leqslant 6\epsilon + 7\delta + \frac{(P^{M1} + P^{M2})}{2}$$
$$\Rightarrow T_{S_1} \leqslant T_{S_{12}S_{21}(\text{opt})}.$$

Now we will compare T_{S_1} with $T_{S_2(\text{opt})}$. We have the following cases:

(1) Assume $P^{M1} \geqslant P + P^{M2}$. Since $2P + P^{M1} + P^{M2} \leqslant 2\delta$, then $P^{M1} < 2\epsilon + 4\delta$. Using $T_{S_2(\text{opt})}$ as given in Eq. (4b) we have:

$$T_{S_1} = 6\epsilon + 6\delta + P + P^{M1} + P^{M2} \leqslant 6\epsilon + 6\delta + 2P + P^{M1} + P^{M2} \leqslant 6\epsilon + 8\delta$$ δ, then
$$\Rightarrow T_{S_1} \leqslant T_{S_2}$$

Lemma 5. *If $P^{M1} + P^{M2} < 2\delta$, $2P + P^{M1} + P^{M2} > 2\delta$ and $P + 2P^{M1} + P^{M2} > 2\epsilon + 6\delta$, then either S_2 or $S_{12}S_{21}$ gives the minimum cycle time.*

Proof. In this region since we assumed that $P^{M1} + P^{M2} < 2\delta$, we have:

$$2\epsilon + 6\delta \leqslant P + 2P^{M1} + P^{M2} < P + P^{M1} + 2\delta \Rightarrow P + P^{M1} > 2\epsilon + 4\delta.$$

$T_{S_{12}S_{21}(\text{opt})}$ is given in Eq. (5a). When we compare this cycle time with T_{S_1} given in Eq. (A.1), we conclude that $T_{S_{12}S_{21}(\text{opt})} < T_{S_1}$. □

As a result of this lemma we showed that S_1 is dominated in this region. On the other hand the following example will show that we cannot establish any dominance relation between cycles S_2 and $S_{12}S_{21}$.

Example 2. Let us suppose that we have four operations with a total processing time of 100, and $\epsilon = 10$ and $\delta = 10$. Because of the tooling constraints, one of the operations with a processing time of 10 must be processed on the first machine, and another one with a processing time of 5 must be processed on the second machine. Therefore, $P^{M1} = 10$ and $P^{M2} = 5$. The remaining two operations with a total operation time of 85 will be allocated to the machines. When we observe the parameters, we see that Lemma 5 is applicable to this case. Since $P + P^{M1} \geqslant 2\epsilon + 4\delta$, the cycle time for $S_{12}S_{21}$ is given in Eq. (5a) and with given parameters becomes 155.

For S_2, the allocation of the operations becomes important. For the first case, assume that we have a total of two operations to be allocated for which, one operation has a processing time of 50 and the other 35 making a total of 85. Calculating the cycle time of S_2 given in Eq. (A.3), we get 140, which is less than the cycle time of $S_{12}S_{21}$. For the second case, assume that we have again two operations to be allocated for which, one operation has a processing time of 75 and the other 10 making a total of 85. Now, the cycle time for S_2 is equal to 160. Since this is greater than 155, for this case $S_{12}S_{21}$ is optimal. Therefore we conclude that depending on the allocation of the operations of S_2, either S_2 or $S_{12}S_{21}$ can be optimal.

Combining the findings in Lemmas 2–5, we now provide the main result of this paper.

Theorem 5

(1) *If $P^{M1} + P^{M2} \geqslant 2\delta$, then S_2 gives the minimum cycle time,*
(2) *Else,*
    (2.1) *If $2P + P^{M1} + P^{M2} \leqslant 2\delta$, then S_1 gives the minimum cycle time,*
    (2.2) *Else,*
        (2.2.1) *If $2P^{M1} + P^{M2} + P \leqslant 2\epsilon + 6\delta$, then $S_{12}S_{21}$ gives the minimum cycle time,*
        (2.2.2) *Else, depending on the allocation of the operations for S_2, either S_2 or $S_{12}S_{21}$ gives the minimum cycle time.*

Remember that for this theorem and the proof we assumed that $P^{M1} \geqslant P^{M2}$. For the reverse case, the results can easily be adapted in analogy with the above analysis resulting in the following corollary:

Corollary 1. *If we assume that $P^{M1} < P^{M2}$ all cases of Theorem* 5 *are still valid except case* 2.2.1 *which should be replaced with the following*:

(2.2.1) *If $P^{M1} + 2P^{M2} + P \leqslant 2\epsilon + 6\delta$, then $S_{12}S_{21}$ gives the minimum cycle time.*

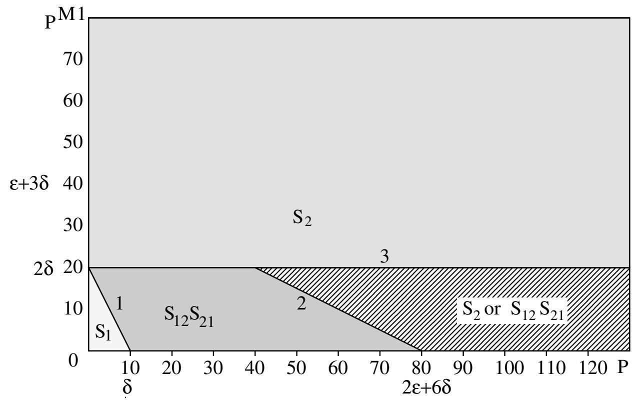

Fig. 3. Optimal Regions for S_1, S_2 and $S_{12}S_{21}$.

3.3. Sensitivity analysis

In this subsection we will perform a sensitivity analysis on parameters such as the robot transportation time δ, and the loading (or unloading) time of the machines ϵ, and show how the regions of optimality change with a change in these parameters. We represent P^{M2} as αP^{M1}, where $0 \leqslant \alpha \leqslant 1$. The following example will be useful in order to analyze the parameters graphically.

Example 3. Assume that $\epsilon = \delta = 10$ and $\alpha = 0$, which means that $P^{M2} = 0$. We can show the regions for this case as a graph of P versus P^{M1} as in Fig. 3.

Consider parameters ϵ, δ, and α one at a time. Theorem 5, case 1 states that if $P^{M1} + P^{M2} \geqslant 2\delta$, then S_2 gives the minimum cycle time. From here we conclude that if the transportation time is zero or negligible, then S_2 always gives the minimum cycle time. This is logical since we can consider the robot as a third machine which is the bottleneck one. Then, the main concern is to minimize the waiting time of the robot in front of the machines. In order to achieve this, in cycle S_2, while processing of a part continues on one of the machines, the robot makes other activities (such as transportation, loading or unloading of the other machine, waiting in front of the other machine, etc.) without being late.

The definition of n-unit cycles states that every machine is loaded and unloaded exactly n times. Thus the loading/unloading times are equivalent for all cycles. However, in cycles S_2 and $S_{12}S_{21}$, while processing continues on one of the machines, the robot does not wait in front of the machine and makes other activities and when the robot returns back to the machine to unload there is a partial waiting time in front of this machine. The loading/unloading time affects these partial waiting times.

We defined α as P^{M2}/P^{M1} and assumed that $0 \leqslant \alpha \leqslant 1$. When we increase P^{M2} from 0 to P^{M1}, this is in favor of S_2 because in cycle S_2 the optimal allocation is the one which balances the processing times on both machines and when P^{M2} is close to P^{M1}, the ability to balance the processing times increases.

4. Conclusion

In this paper, we studied the 2-machines, identical parts robotic operation allocation problem with tooling constraints. The operation allocation flexibility is a direct consequence of assuming that the machines in the robotic cell are CNC machines as is the case for machining operations. The problem is to find the allocation of the operations to the machines and the corresponding robot move cycle that jointly minimize the cycle time. As a solution to this problem, we proved in Theorem 4 that the optimal solution is either a 1-unit or a 2-unit cycle. In Theorem 5, we presented the regions of optimality for these robot move cycles. We showed that the study of Sethi et al. (1992) becomes a special case of our study. We conducted a sensitivity analysis on parameters. Suggested future research directions include considering non-identical machines so that the time to complete a specific operation is different for the two machines. Another topic is to consider a dual-gripper robot which manages loading and unloading at the same time.

Acknowledgments

The authors would like to acknowledge the three anonymous referees for their helpful guidance on improving the presentation and the contents of the paper. This work was supported in part by the Scientific and Technical Research Council of Turkey under grant #2211.

Appendix A. Cycle time calculations

We will find the cycle times of S_1, S_2 and $S_{12}S_{21}$ for a given k-allocation type. Let us start with S_1. Sethi et al. (1992) proved the cycle time of S_1 to be:

$$T_{S_1} = 6\epsilon + 6\delta + a + b,$$

where a and b are the processing times of the part on the first and second machines respectively. With our notation this cycle time is the following

$$T_{S_1} = 6\epsilon + 6\delta + P + P^{M1} + P^{M2} \tag{A.1}$$

and it does not depend on the allocation type. So whatever the allocation is, this cycle time is the same.

Now consider S_2. Sethi et al. (1992) also provided that:

$$T_{S_2} = 6\epsilon + 8\delta + \max\{0, a - (2\epsilon + 4\delta), b - (2\epsilon + 4\delta)\},$$

where a and b are defined as above. With our definitions the above equation is the cycle time of S_2 with one-allocation type where, $a = P^{M1} + P_1(\sigma_1)$ and $b = P^{M2} + P - P_1(\sigma_1)$. Then the cycle time of a one-allocation for cycle S_2 with our notation is the following:

$$T_{S_2(\sigma_1)} = 6\epsilon + 8\delta + \max\{0, P^{M1} + P_1(\sigma_1) - (2\epsilon + 4\delta), P^{M2} + P - P_1(\sigma_1) - (2\epsilon + 4\delta)\}. \tag{A.2}$$

Now consider a two-allocation type for the cycle S_2: for a part with the first allocation type, the processing time on the first machine is $P^{M1} + P_1(\sigma_2)$ and on the second machine is $P^{M2} + P - P_1(\sigma_2)$. For a part with the second allocation type, the processing times on the first and the second machines are $P^{M1} + P_2(\sigma_2)$ and $P^{M2} + P - P_2(\sigma_2)$ respectively. Then, the long run average cycle time to produce one part with cycle S_2 with a specific two-allocation type σ_2 is:

$$\begin{aligned} T_{S_2(\sigma_2)} = 6\epsilon + 8\delta &+ 1/2(\max\{0, P^{M1} + P_1(\sigma_2) - (2\epsilon + 4\delta), P^{M2} + (P - P_2(\sigma_2)) - (2\epsilon + 4\delta)\}) \\ &+ 1/2(\max\{0, P^{M1} + P_2(\sigma_2) - (2\epsilon + 4\delta), P^{M2} + (P - P_1(\sigma_2)) - (2\epsilon + 4\delta)\}). \end{aligned} \tag{A.3}$$

We can easily generalize this for a k-allocation type for the cycle S_2 as follows:

$$\begin{aligned} T_{S_2(\sigma_k)} = 6\epsilon + 8\delta &+ 1/k(\max\{0, P^{M1} + P_1(\sigma_k) - (2\epsilon + 4\delta), P^{M2} + (P - P_k)(\sigma_k) - (2\epsilon + 4\delta)\}) \\ &+ 1/k(\max\{0, P^{M1} + P_2(\sigma_k) - (2\epsilon + 4\delta), P^{M2} + (P - P_1(\sigma_k)) - (2\epsilon + 4\delta)\}) \\ &+ 1/k(\max\{0, P^{M1} + P_3(\sigma_k) - (2\epsilon + 4\delta), P^{M2} + (P - P_2(\sigma_k)) - (2\epsilon + 4\delta)\}) + \cdots \\ &+ 1/k(\max\{0, P^{M1} + P_k(\sigma_k) - (2\epsilon + 4\delta), P^{M2} + (P - P_{(k-1)}(\sigma_k)) - (2\epsilon + 4\delta)\}). \end{aligned} \tag{A.4}$$

Now let us consider cycle $S_{12}S_{21}$. The long run average cycle time for one-allocation type for the cycle $S_{12}S_{21}$ can be derived as follows: The activity sequence of this robot move cycle is $A_0A_1A_0A_2A_1A_2$. Initially both machines are empty and the robot is in front of the input buffer just starting to take a part. The robot takes a part from the input buffer transports it to the first machine and loads it, $(\epsilon + \delta + \epsilon)$; waits in front of the machine to finish the processing of the part, $(P^{M1} + P_1(\sigma_1))$; unloads the part transports it to the second machine and loads it, $(\epsilon + \delta + \epsilon)$; returns back to the input buffer, takes another part, transports it to the first machine and loads it, $(2\delta + \epsilon + \delta + \epsilon)$; travels to the second machine, (δ); waits if necessary, (w_{21}); unloads the machine, transports the part to the output buffer and drops it $(\epsilon + \delta + \epsilon)$; travels to the first machine, (2δ); waits if necessary, (w_{11}); unloads the machine, transports the part to the second machine and loads it, $(\epsilon + \delta + \epsilon)$; waits for the machine to finish the processing, $(P^{M2} + (P - P_1(\sigma_1)))$; unloads the machine, transports it to the output buffer, drops the part and returns back to the input buffer so that the initial and the final states are the same, $(\epsilon + \delta + \epsilon + 3\delta)$. As a result, total time to produce two parts is:

$$T_{S_{12}S_{21}(\sigma_1)} = 12\epsilon + 14\delta + P + P^{M1} + P^{M2} + w_{11} + w_{21},$$

where $w_{11} = \max\{0, P^{M1} + P_1(\sigma_1) - (2\epsilon + 4\delta + w_{21})\}$ and $w_{21} = \max\{0, P^{M2} + P - P_1(\sigma_1) - (2\epsilon + 4\delta)\}$.

In other words, $w_{11} + w_{21} = \max\{0, P^{M1} + P_1(\sigma_1) - (2\epsilon + 4\delta), P^{M2} + P - P_1(\sigma_1) - (2\epsilon + 4\delta)\}$.

Hence the cycle time for one-allocation for the cycle $S_{12}S_{21}$ is:

$$T_{S_{12}S_{21}(\sigma_1)} = \frac{12\epsilon + 14\delta + P + P^{M1} + P^{M2} + \max\{0, P^{M1} + P_1(\sigma_1) - (2\epsilon + 4\delta), P^{M2} + P - P_1(\sigma_1) - (2\epsilon + 4\delta)\}}{2}. \tag{A.5}$$

Now consider the case when we have two-allocation types:

$$T_{S_{12}S_{21}(\sigma_2)} = \frac{12\epsilon + 14\delta + P^{M1} + P^{M2} + P_1(\sigma_2) + (P - P_2(\sigma_2)) + w_{12} + w_{21}}{2},$$

where $w_{12} = \max\{0, P^{M1} + P_2(\sigma_2) - (2\epsilon + 4\delta + w_{21})\}$ and $w_{21} = \max\{0, P^{M2} + P - P_1(\sigma_2) - (2\epsilon + 4\delta)\}$. In other words,

$$\begin{aligned} T_{S_{12}S_{21}(\sigma_2)} = {} & 1/2(12\epsilon + 14\delta + P^{M1} + P^{M2} + P_1(\sigma_2) + (P - P_2(\sigma_2))) + 1/2(\max\{0, P^{M1} + P_2(\sigma_2) \\ & - (2\epsilon + 4\delta), P^{M2} + P - P_1(\sigma_2) - (2\epsilon + 4\delta)\}). \end{aligned} \tag{A.6}$$

Appendix B. Proof of Theorem 4

The following definition will play a crucial role in the forthcoming context.

Definition 5. *State* 0 is defined to be the state in which both machines are empty and the robot is in front of the input buffer.

Consider any n-unit robot move cycle during which State 0 is encountered. Clearly, the allocations immediately preceding and those immediately following this state can be treated as completely independent from each other. As far as the cycle time computations are involved, the contribution of the portion of the cycle between two State 0's is simply additive. This observation deserves further attention:

Fact 1. The average cycle time of an n-unit robot move cycle is simply the average of the cycle time corresponding to a sub-cycle between two State 0's and the cycle time of the remaining cycle after the sub-cycle is extracted.

After inspecting the cycles we have introduced thus far, it is easy to state that:

Fact 2. S_1 cycle starts and ends with State 0. S_{12} sequence starts with State 0 and S_{21} sequence ends with State 0.

With these observations, we are now ready to proceed with the proof of Theorem 4.

Proof. Hall et al. (1997) showed that any n-unit robot move cycle can be represented by the four robot move sequences: S_1, S_2, S_{12} and S_{21}. However, not all of these sequences can follow any other since otherwise the basic feasibility assumptions of Crama et al. (2000) stating that the robot cannot load an already loaded machine and cannot unload an already empty machine may be violated. Fig. 1 depicts the feasible transitions from one sequence to another. Now let us consider any n-unit robot move cycle with the optimal allocation type. In its generality, we assume this cycle to contain at least one of these four robot move sequences. The allocation of the operations does not affect the loading/unloading and transportation times but only affects the processing times which in turn affect the waiting times of the robot in front of the machines. Let us first analyze one of the S_1 cycles within this n-unit robot move cycle. Using Facts 1 and 2 above we conclude that the allocation for this cycle does not affect the remaining part of the cycle. Also, as mentioned previously, there is no allocation problem for S_1 cycle, and the waiting time for S_1 is $P + P^{M1} + P^{M2}$, independent of the allocation. When we remove the activity sequence $A_0A_1A_2$, corresponding to S_1 from the activity sequence of the n-unit robot move cycle, we get a new feasible $(n-1)$-unit robot move cycle for which the optimal allocation for the remaining parts does not change.

Assume that there are a total of z S_1 cycles within the n-unit robot move cycle where $z = 0, 1, \ldots, n$. Let T_{n-z} be the cycle time of the new $(n-z)$-unit robot move cycle, say C_{n-z}, which is attained by removing the activity sequences of all S_1 cycles. Hence, the cycle time of our n-unit robot move cycle is:

$$T_n = \frac{z * T_{S_1} + (n-z) * T_{n-z}}{n}.$$

This equation states that $T_n \geqslant \min\{T_{S_1}, T_{n-z}\}$.

Now let us consider C_{n-z}. From Fig. 1 we see that between two S_{12} sequences one S_{21} sequence must be performed. Between one S_{12} and one S_{21} any number of S_2 sequences can be performed. Thus we can partition C_{n-z} into sub-cycles each of which starts with S_{12} performs a number of S_2's and ends up with S_{21}. Furthermore, as stated as Fact 1 above, the allocation for such cycles does not affect the allocation for the remaining part of the cycle and is not affected by the allocation of the remaining part of the cycle. The cycle time of the whole n-unit robot move cycle is a convex combination of the cycle times of these sub-cycles. Let T_i be the cycle time of the ith sub-cycle with the optimal allocation type then, clearly, $T_{n-z} \geqslant \min_i\{T_i\}$. Henceforth, $T_n \geqslant \min\{T_{S_1}, \min_i\{T_i\}\}$.

Now let us consider any such cycle which starts with S_{12}, performs a total of l S_2's $l = 0, 1, \ldots$ and ends up with S_{21}. Note that this is an $(l+2)$-unit cycle. Let us represent this cycle as $S_{12}(l - S_2)S_{21}$. Since in every cycle each machine is loaded and unloaded an equal number of times, the average loading/unloading time to produce one part for all cycles are equal to each other, namely, 6ϵ. Repeating cycle S_2 l times requires a total of $8l\delta$. Repeating S_{12} one time, which has an activity sequence of $A_0A_1A_0$, requires 5δ and the following activity of this sequence can only be A_2 which requires an additional δ to travel between final state of S_{12} (first machine) to the initial state of the next sequence (second machine). In a similar way repeating S_{21} ($A_2A_1A_2$) one time requires 5δ and an additional 3δ to travel between the last state of S_{21} (output buffer) to initial state of the following sequence which must start with activity A_0 (input buffer). Then the average travel time to produce one part with the cycle $S_{12}(l - S_2)S_{21}$ is:

$$\frac{6\delta + 8l\delta + 8\delta}{l+2} = \frac{8l + 14}{l+2}\delta.$$

The last component in our cycle time representations is the total waiting time of the robot in front of the machines. We can find the waiting time as follows: Assume for the cycle $S_{12}(l - S_2)S_{21}$ using k allocation types is optimal and consider the specific allocation σ_k. The first part will be produced according to the cycle

S_{12} which implies a full waiting time in front of the first machine: $P^{M1}+P_1(\sigma_k)$, and will be processed according to S_2 on the second machine which means a partial waiting time in front of the second machine: $w_{21}=\max\{0,P^{M2}+(P-P_1(\sigma_k))-(2\epsilon+4\delta)\}$. The next l parts will be processed according to S_2 cycle on both machines. The partial waiting time for the second part on the first machine is: $w_{12}=\max\{0,P^{M1}+P_2(\sigma_k)-(2\epsilon+4\delta+w_{21})\}$. Then we have:

$$w_{12}+w_{21}=\max\{0,P^{M1}+P_2(\sigma_k)-(2\epsilon+4\delta),P^{M2}+(P-P_1(\sigma_k))-(2\epsilon+4\delta)\}.$$

Proceeding in the same way for all parts, total waiting time to produce $l+2$ parts is:

$$\begin{aligned}&P^{M1}+P^{M2}+P_1(\sigma_k)+(P-P_{(l+2)}(\sigma_k))\\&\quad+\max\{0,P^{M1}+P_2(\sigma_k)-(2\epsilon+4\delta),P^{M2}+(P-P_1(\sigma_k))-(2\epsilon+4\delta)\}\\&\quad+\max\{0,P^{M1}+P_3(\sigma_k)-(2\epsilon+4\delta),P^{M2}+(P-P_2(\sigma_k))-(2\epsilon+4\delta)\}\\&\quad+\max\{0,P^{M1}+P_4(\sigma_k)-(2\epsilon+4\delta),P^{M2}+(P-P_3(\sigma_k))-(2\epsilon+4\delta)\}+\cdots\\&\quad+\max\{0,P^{M1}+P_{(l+1)}(\sigma_k)-(2\epsilon+4\delta),P^{M2}+(P-P_l(\sigma_k))-(2\epsilon+4\delta)\}\\&\quad+\max\{0,P^{M1}+P_{(l+2)}(\sigma_k)-(2\epsilon+4\delta),P^{M2}+(P-P_{(l+1)}(\sigma_k))-(2\epsilon+4\delta)\}.\end{aligned}$$

In the above equation let us insert $P_1(\sigma_k)$ and $(P-P_{(l+2)}(\sigma_k))$ into the first and last max terms respectively. By this change all other max terms remain the same but these two become:

$$\begin{aligned}&\max\{P_1(\sigma_k),P^{M1}+P_2(\sigma_k)+P_1(\sigma_k)-(2\epsilon+4\delta),P^{M2}+P-(2\epsilon+4\delta)\}\quad\text{and}\\&\max\{P-P_{(l+2)}(\sigma_k),P^{M1}+P-(2\epsilon+4\delta),P^{M2}+(P-P_{(l+1)}(\sigma_k))+(P-P_{(l+2)}(\sigma_k))-(2\epsilon+4\delta)\}.\end{aligned}$$

Under the optimal allocation we must have $P_1(\sigma_k^*)=0$ and $P_{(l+2)}(\sigma_k^*)=P$. This means for the first part allocating all operations that are in set O to the second machine and for the last part allocating them to the first machine. Furthermore, since we assumed that $P^{M1}\geqslant P^{M2}$, letting $P_{(l+1)}(\sigma_k^*)=0$ does not change the cycle time. As a result of these, the last max term of the above equation becomes:

$$\max\{0,P^{M1}+P-(2\epsilon+4\delta),P^{M2}+P-(2\epsilon+4\delta)\}=\max\{0,P^{M1}+P-(2\epsilon+4\delta)\}.$$

After including the loading/unloading and travel times, the cycle time of $S_{12}(l-S_2)S_{21}$ under optimal allocation σ_k^* becomes:

$$6\epsilon+\frac{(14+8l)}{(l+2)}\delta+\frac{1}{(l+2)}(P^{M1}+P^{M2}) \tag{B.1}$$
$$+1/(l+2)(\max\{0,P^{M1}+P_2(\sigma_k^*)-(2\epsilon+4\delta),P^{M2}+P-(2\epsilon+4\delta)\}) \tag{B.2}$$
$$+1/(l+2)(\max\{0,P^{M1}+P_3(\sigma_k^*)-(2\epsilon+4\delta),P^{M2}+(P-P_2(\sigma_k^*))-(2\epsilon+4\delta)\}) \tag{B.3}$$
$$+1/(l+2)(\max\{0,P^{M1}+P_4(\sigma_k^*)-(2\epsilon+4\delta),P^{M2}+(P-P_3(\sigma_k^*))-(2\epsilon+4\delta)\}) \tag{B.4}$$
$$+\cdots$$
$$+1/(l+2)(\max\{0,P^{M1}-(2\epsilon+4\delta),P^{M2}+(P-P_l(\sigma_k^*))-(2\epsilon+4\delta)\}) \tag{B.$(l+1)$}$$
$$+1/(l+2)(\max\{0,P^{M1}+P-(2\epsilon+4\delta)\}). \tag{B.$(l+2)$}$$

Let $W(S_{12}S_{21})$ represent the waiting time of two-allocation for cycle $S_{12}S_{21}$ with optimal allocation σ_k^*, which in Eq. (3) was found to be: $W(S_{12}S_{21})=\frac{1}{2}(\max\{0,P+P^{M1}-(2\epsilon+4\delta)\})$. Then, the (B.($l$+2))nd component in the above representation is $\frac{2}{(l+2)}W(S_{12}S_{21})$. Consider lines (B.2)–(B.(l+1)) above. They share the same pattern as the waiting time component corresponding to an l-allocation type for the cycle S_2 in (A.4). Let $W(S_2)$ be this waiting time component in (A.4) when we plug in the respective allocations from (B.2)–(B.(l+1)). The summation of the components (B.2)–(B.(l+1)) is then $\frac{l}{l+2}W(S_2)$. In conclusion,

$$T_{S_{12}(l-S_2)S_{21}} = \frac{2(6\epsilon + 7\delta + \frac{P^{M1}+P^{M2}}{2} + W(S_{12}S_{21})) + l(6\epsilon + 8\delta + W(S_2))}{l+2}.$$

In particular, the cycle time of $S_{12}(l-S_2)S_{21}$ is a convex combination of the cycle times of two-allocation for $S_{12}S_{21}$ and l-allocation for S_2. Clearly,

$$T_{S_{12}(l-S_2)S_{21}} \geqslant \min\{T_{S_2(\mathrm{opt})}, T_{S_{12}S_{21}(\mathrm{opt})}\}.$$

Finally, we have managed to show that the cycle time of the n-unit cycle we started out with, $T_n \geqslant \min\{T_{S_1}, T_{S_2(\mathrm{opt})}, T_{S_{12}S_{21}(\mathrm{opt})}\}$. □

References

Brauner, N., Finke, G., 1997. Final Results on the One-cycle Conjecture in Robotic Cells. Internal note, Laboratoire LEIBNIZ, Institut IMAG, Grenoble, France.

Crama, Y., van de Klundert, J., 1997. Cyclic scheduling of identical parts in a robotic cell. Operations Research 45, 952–965.

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

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

Dawande, M., Geismar, H.N., Sethi, S.P., 2002a. Dominance of cyclic solutions and challenges in the scheduling of robotic cells. Working paper, University of Texas at Dallas, to appear in SIAM Review.

Dawande, M., Sriskandarah, C., Sethi, S.P., 2002b. On throughput maximization in constant travel-time robotic cells. Manufacturing and Service Operations Management 4, 296–312.

Dawande, M., Geismar, H.N., Sethi, S.P., Sriskandarajah, C. 2003. Sequencing and scheduling in robotic cells: Recent developments. Working paper, University of Texas at Dallas, to appear in Journal of Scheduling.

Garey, M.G., Johnson, D.S., 1976. Computers and Intractability: A Guide to the Theory of *NP*-completeness. W.H. Freeman.

Gray, A.E., Seidmann, A., Stecke, K.E., 1993. A synthesis of decision models for tool management in automated manufacturing. Management Science 39, 549–567.

Hall, N.G., Kamoun, H., Sriskandarajah, C., 1997. Scheduling in robotic cells: Classification, two and three machine cells. Operations Research 45, 421–439.

Hall, N.G., Kamoun, H., Sriskandarajah, C., 1998. Scheduling in robotic cells: Complexity and steady state analysis. European Journal of Operations Research 109, 43–65.

King, R.E., Hodgson, T.J., Chafee, F.C., 1993. Robot task scheduling in a flexible manufacturing cell. IIE Transactions 25, 80–87.

Kise, H., Shioyama, T., Ibaraki, T., 1991. Automated two-machine flowshop scheduling: A solvable case. IIE Transactions 23, 10–16.

Miller, R.K., Walker, T.C., 1990. FMS/CIM Systems Integration Handbook. The Fairmont Press, Lilburn, GA.

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

Sriskandarajah, C., Hall, N.G., Kamoun, H., 1998. Scheduling large robotic cells without buffers. Annals of Operations Research 76, 287–321.