Robotics and Computer–Integrated Manufacturing 48 (2017) 121–131

Contents lists available at ScienceDirect

Robotics and Computer–Integrated Manufacturing

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

Pure cycles in two-machine dual-gripper robotic cells

Hakan Gultekin[a,*], Özden O. Dalgıç[b], M. Selim Akturk[c]

[a] Department of Industrial Engineering, TOBB University of Economics and Technology, Ankara, Turkey
[b] Department of Management Sciences, University of Waterloo, Waterloo, ON, Canada
[c] Department of Industrial Engineering, Bilkent University, Ankara, Turkey

ARTICLE INFO

Keywords:
Robotic cell scheduling
Pure cycle
Dual-gripper robot
Throughput maximization

ABSTRACT

We consider a robotic cell served by a dual-gripper robot that consists of identical CNC machines placed linearly and a material handling robot loading/unloading the machines and transporting the parts between them. Identical parts are to be processed in this system and the CNC machines are capable of performing all the operations that a part requires. We consider the problem of sequencing activities of the robot in order to maximize the throughput rate. As a consequence of the flexibility of the CNC machines, a new class of robot move sequences, named as pure cycles, arises. In a pure cycle, the robot loads and unloads each machine once and each part is processed on exactly one of the machines. Thereby, the problem is to determine the best pure cycle that maximizes the throughput rate. We first determine the feasibility conditions for the pure cycles and prove some basic results that reduces the number of feasible pure cycles to be investigated. We analyze 2-machine robotic cells in detail and prove that five of the cycles among a huge number of feasible pure cycles dominate the rest. We determine the parameter regions in which each of the five cycles is optimal. We also analyze the performance improvement that can be attained by using a dual gripper robot and provide managerial insights.

1. Introduction

A robotic cell is a type of production system in which there are a number of machines as well as a robot loading and unloading the machines and transporting the parts between the machines. There is an input and an output buffer providing material flow to and departure from the cell, respectively. An m-machine robotic cell can be seen in Fig. 1. Robotic cells are extensively used in many industries such as electronics and metal cutting. With the recent developments in technology, machines used in these cells highly improved in terms of their capabilities. Especially, those utilizing computer numerical control (CNC) machines provide a huge flexibility for production systems. A CNC machine can perform a large variety of operations as long as the necessary cutting tools are loaded in their tool magazines. Besides the capabilities of the machines, the capability of the robot is another concern in robotic cells. Particularly, the number of grippers on a robot can have a substantial impact on its capability. A gripper is a component of a robot that holds parts to be processed while the robot performs handling operations such as transporting a part, or loading a machine with a part. The number of grippers that a robot has, determines the number of parts that can be handled by the robot at the same time. In this study, we have m identical machines (except the input and the output buffers) which have an in-line robotic cell layout. There are no buffer areas in between the machines. Each of the parts are identical and there are a number of operations to be performed by the machines. Each machine is also identical and flexible enough to perform all the operations required by the parts. There is a dual gripper robot loading and unloading the machines and transporting the parts between the machines. Dual gripper feature provides robots the capability of using both grippers while loading/unloading the machines and transporting the parts. It is assumed that the robot is capable of switching its grippers while it travels between the machines. In the course of this study, we tackle the problem of minimizing the throughput rate.

The studies in the literature are mostly on flow shop robotic cells where parts visit each machine in the same order. Most of these studies consider cyclic schedules where the same sequence of robot moves are repeated indefinitely. It is practical to consider such schedules since the moves of the robot are coded by a finite set of computer instructions. Furthermore, Dawande et al. [4] proved that optimal throughput rates are attained by cyclic robot moves. One of the very first studies in the robotic cell scheduling literature is the paper by Sethi et al. [20]. They considered 2 and 3-machine robotic cells and determined the best 1-unit cycles for both cases. An n-unit cycle is defined as the robot move

[*] Corresponding author.
E-mail addresses: hgultekin@etu.edu.tr (H. Gultekin), oodalgic@uwaterloo.ca (Ö.O. Dalgıç), akturk@bilkent.edu.tr (M.S. Akturk).

http://dx.doi.org/10.1016/j.rcim.2017.03.004
Received 20 June 2016; Received in revised form 28 January 2017; Accepted 18 March 2017
Available online 30 March 2017

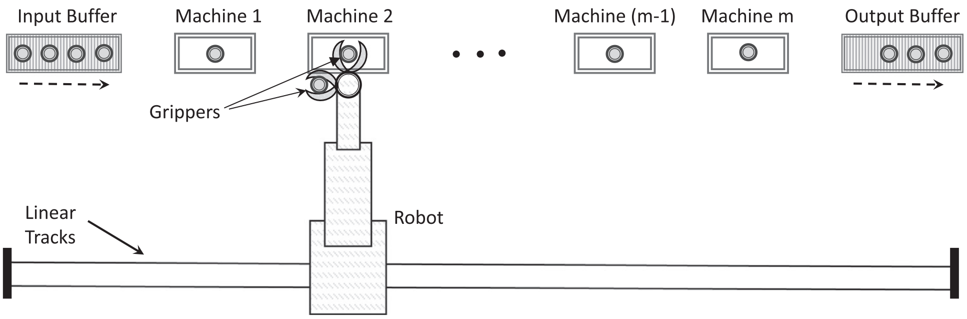

Fig. 1. *m*-machine robotic cell with a dual-gripper robot.

cycle in which each machine is loaded and unloaded exactly n times and n parts are produced at the end of one repetition. They proved that the set of 1-unit cycles are optimal for the 2-machine case. Crama and van de Klundert [3] developed an $O(m^3)$ time algorithm that provides the best 1-unit cycle in an m-machine cell. Hall et al. [16] showed that 1-unit cycles are also optimal in 3-machine robotic cells. However, Brauner and Finke [2] proved that 1-unit cycles need not be optimal for cells with $m \geq 4$. The problem of finding the optimal n-unit cycle for an arbitrary n in an m-machine robotic cell is still open.

The technological progress in robotics led to new research problems. One of these is the consideration of the dual gripper robots. Zhang et al. [23] conducted a sensitivity analysis on different robotic cell design parameters including the gripper types. The problems concerning dual gripper robotic cells are much more intricate than the ones concerning single gripper robotic cells. Even if the number of machines is kept in small sizes, the number of cycles may increase drastically. Sethi et al. [21] showed that the total number of 1-unit cycles in a 2-machine dual gripper robotic cell is 52 in contrast to only 2 in a single gripper counterpart. They showed that the use of a dual gripper robot instead of a single gripper one may reduce the cycle time by 50%. Geismar et al. [11] considered a dual-gripper robotic cell with parallel machines at each stage and determined the optimal cycle under practical assumptions. They also assessed the potential benefits of using dual gripper robots over single gripper ones for the considered cell. Drobouchevitch et al. [5] developed a polynomial time algorithm to determine the optimal 1-unit cycle working under some specific assumptions on parameter values. Geismar et al. [12] provided approximation algorithms to compare single gripper robots with dual gripper robots in a circular layout under the constant travel time assumption. Jung et al. [19] studied the same problem with additive inter-machine travel times and introduced a new class of schedules, which were referred to as epi-cyclic cycles. Foumani et al. [9] considered a cell in which a hub machine is revisited after each secondary operation is performed by a separate secondary machine located sequentially. The material handling is performed by a dual-gripper robot. They proposed a 2-unit cycle and proved its optimality. Galante and Passannanti [10] studied the problem of minimizing the cycle time in serial manufacturing systems with multiple dual-gripper robots. A comparison of robotic cells with parallel machines and multiple dual gripper robots can be found in Geismar et al. [13].

Robotic cells can be utilized in different industries including chemical, electronics, automotive, and metal cutting industries. In order the cell to operate without the interaction of manual operators, the machines, the robot, and the cell controller must communicate with each other. This necessitates the machines used in the cell to be CNC machines. These machines are highly flexible that can perform a large variety of operations. Another research topic is the consideration of such flexible robotic cells. Akturk et al. [1] considered a flexible robotic cell in which there are two identical CNC machines and showed that a 2-unit cycle can also be optimal even in 2-machine cells. Gultekin et al. [14] proposed a new class of robot move cycles that emanate from the capability of the CNC machines for performing a large variety of operations. They showed that these cycles provide higher throughput rates than flowshop type robot move cycles in 2-machine and 3-machine robotic cells. Gultekin et al. [15] considered robotic cells with m identical CNC machines and named the new class of robot move cycles as *pure cycles*. In a pure cycle, all operations of a part are performed by a single machine and the parts are not transferred between the machines. Practical examples of robotic cells utilizing pure cycles include air conditioner shaft manufacturing, engine manufacturing, automobile camshaft and crankshaft manufacturing, and gear manufacturing in which metal cutting operations are performed by CNC machines. In pure cycles, all machines are loaded and unloaded exactly once. Hence, a pure cycle produces exactly m parts in an m-machine cell. Gultekin et al. [15] proved that two specific pure cycles are optimal for most of the possible parameter values in a single gripper setting. For the remaining cases, they provided the worst case performances of these two cycles. They also proved that pure cycles dominate all flow shop type cycles studied in the literature. Therefore, restricting the problem to the flowshop type robot move cycles may result in suboptimal solutions for the cells utilizing flexible machines. A similar analysis is performed by Yildiz et al. [22] for the pure cycles in robot-centered cells, where the machines are located around the robot that makes rotational movements. The authors also considered the joint optimization of the cycle time and the total manufacturing cost assuming that the processing times are controllable. For the robotic cells where the robot has the swap ability, Foumani and Jenab [7] analyzed re-entrant robotic cells under different configurations, whereas Foumani and Jenab [8] studied pure cycles strategy with two machines and identical parts.

In this study, we consider a dual gripper robotic cell with flexible machines. The problem is to determine the best pure cycle that maximizes the throughput rate. The flowshop problem that considers a dual gripper robot is more complex compared to the single gripper problem and the pure cycle problem is more complex compared to the flowshop cell scheduling problem. To the best of our knowledge, this is the first study that considers dual gripper robotic cells and pure cycles jointly which increases the complexity drastically. This study contributes to the literature by developing the necessary analytical framework to analyze pure cycles in dual gripper cells, which can be extended to other complex problems with pure cycles and dual gripper robots. We also contribute to the literature by reducing the number of potentially optimal cycles to five and by determining the parameter regions where each of these cycles are optimal. Moreover, we provide the potential theoretical benefits and quantify the average performance improvement of using a dual gripper robot instead of a single gripper robot.

The problem tackled here is also related with the parallel machine scheduling with a common server problems (see e.g. [17,18,6]). The main differences between our problem and that literature are threefold. Firstly, in that literature, the setup time of the machines is a problem parameter which is arbitrary for each job. In our study, the setup time is a variable which is directly associated with the sequence of the robot activities including transporting the parts between the machines and loading/unloading the machines. Secondly, the unloading of the machines is also performed by the server (robot) in our problem, which is not the case in that literature. Lastly, classical scheduling objectives such as the minimization of the makespan or the total completion time are considered in that literature assuming that a finite number of parts are to be produced. However, we assume that an infinite (or very large) number of identical parts are to be produced as commonly done in the literature on robotic cell scheduling. Therefore, a cyclic scheduling framework is utilized.

In the next section, we provide the notation used throughout the paper and define the problem formally. In Section 3, we consider the 2-machine cells in more detail and find the optimal pure cycles analytically. In Section 4, we provide managerial insights and future research directions.

2. Problem definition and basic results

In this section, we provide some definitions and notations that we have used throughout the paper.

P: Total processing time of a part on any one of the machines.
δ: Robot travel time between two adjacent machines.
ε: Load/unload time of each machine.
θ: Gripper switch time. Resulting from most practical situations it is assumed to be less than the travel time, ($\theta \le \delta$) and the processing time ($\theta \le P$).
w_i: Waiting time of the robot in front of machine i before unloading it. When the robot arrives in front of machine i to unload it, if the processing of the part is already completed, w_i is 0, otherwise it is equal to the remaining processing t-ime.

Our aim is to minimize the cycle time. The cycle time is defined as the long run average time to produce one part. Note that in an m-machine robotic cell a single repetition of a pure cycle produces m parts.

C_i^m : i^{th} pure cycle in an m-machine robotic cell.
$T_{C_i^m}$: Cycle time of cycle C_i^m. Note that, we define the cycle time as the total time required to complete the cycle instead of the time required to produce a single part.

In order to derive all pure cycles in an m machine cell and calculate their cycle times we will use the following definitions.

Definition 1. A **state** is defined as the image of the system. It is an $(m+2)$-tuple which indicates whether the machines are loaded or not (1 for loaded and 0 for empty), and the type of part the grippers have (new (n), processed (f), or none (e)). First m elements denote the machines and the last two denotes the grippers in the state definition.

For example, in a 2-machine cell the state $(0, 0, e, e)$ indicates that both machines are idle and the grippers are empty, whereas $(0, 1, f, n)$ indicates that Machine 1 (M_1) is idle and Machine 2 (M_2) is occupied with a part. Additionally, Gripper 1 (G_1) holds a processed part and Gripper 2 (G_2) holds a new part. There are a total of $3^2 \cdot 2^m$ different states for an m-machine cell (2 different values for each machine status and 3 different values for each of the two grippers). For instance the number possible states for a 2-machine cell is equal to 36.

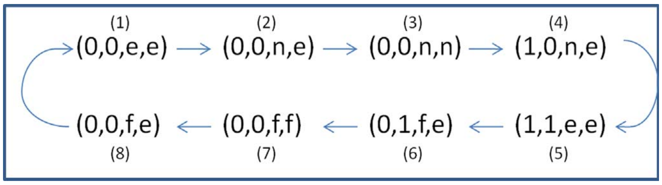

Fig. 2. An example pure cycle, C_2^2, expressed as a permutation of states.

Definition 2. An **action** is the robot motion such as loading and unloading in front of any machine (including the input and the output buffers). When the robot is in front of a machine, only one action can occur and this action depends on the status of that machine (whether it has a part on it or not). When the robot is in front of the input buffer or an occupied machine, the action is **unloading**. When the robot is in front of the output buffer or an idle machine, the action is **loading**.

In an m-machine robotic cell operating under a pure cycle strategy, the robot picks up a part from the input buffer m times and drops m processed parts to the output buffer. Furthermore, each machine is loaded and unloaded exactly once at the end of a pure cycle. All these make a total of 4m actions to be performed by the robot in any pure cycle. Since the system moves from one state to another after the each action, a pure cycle in an m-machine cell can be expressed as a cyclic sequence of 4m states. As an example, the sequence of eight states of one of the pure cycles in a 2-machine cell, named as C_2^2, is illustrated in Fig. 2. In this figure, the numbers above the states indicate the sequence starting from $(0, 0, e, e)$ state.

Although this state definition does not contain the position information of the robot, this can easily be inferred using two consecutive states. For example, consider the two consecutive states, $(0, 0, e, e)$ and $(0, 0, n, e)$. In the first state, both grippers are empty, whereas in the following state G_1 holds a new part. Hence, it can be inferred that the robot is in front of the input buffer to perform this activity.

Not all possible permutations of the states yield a feasible pure cycle. In order to have a feasible cycle, two consecutive states must satisfy a number of feasibility conditions that can be listed as follows:

- The robot actions correspond to loading/unloading operations. These actions change the statuses of the grippers resulting in state changes. The robot cannot use both grippers at the same time to load/unload. Therefore, the status of exactly one gripper must be different in any two consecutive states. As an example, let us consider the $(0, 0, n, n)$ state. The $(0, 0, e, e)$ state cannot be an immediate successor of this state since the statuses of both grippers must be changed from "n" to "e" simultaneously meaning that two loading operations must be performed at the same time. Therefore, in order to move from state $(0, 0, e, e)$ to $(0, 0, n, n)$, either one of the intermediate states $(0, 0, e, n)$ or $(0, 0, n, e)$ must be in between these two states.
- The status of the grippers cannot change from "n" to "f" or "f" to "n" directly. In order to change the status from "n" to "f" or "f" to "n", the robot must first drop the current part from the gripper which means the state must first become "e".
- The changes "e" to "n" or "f" to "e" mean that the robot picks up a part from the input buffer using this gripper, or the robot drops a processed part to the output buffer, respectively. Hence, the status of all machines must remain the same.
- If the change is "e" to "f", then the robot unloads a machine with this gripper. Therefore, the status of exactly one of the occupied machines must change from "1" to "0".
- If the change is "n" to "e", then the robot loads an idle machine with this gripper. For this reason, the status one of the idle machines must change from "0" to "1".

Before applying these feasibility conditions, the total number of feasible and infeasible cycles for the m-machine case equals to $(9 \cdot 2^m)^{4m}$. As an example, for the 2-machine case there are about $3 \cdot 10^{12}$ feasible and infeasible cycles in total. However, if we apply these feasibility conditions, the number of feasible pure cycle reduces to 2.674. As it can be seen, the number of feasible cycles is still very large even for only two machines. By using the following definitions, we will be able to reduce the necessary number of feasible cycles to examine from 2.674 to 266.

We will use the following definition of Gultekin et al. [15] to represent the robot move cycles.

Definition 3. An **activity** is defined as the action of the robot in front of a specific machine (including input and output buffer), that leads to a change of state. These activities can be listed as follows:

I: Unloading a part from the input buffer (repeated m times in a pure cycle),
L_i: Loading a part to M_i, $i = 1, 2, .., m$ (repeated once for each machine in a pure cycle),
U_i: Unloading a part from M_i, $i = 1, 2, .., m$ (repeated once for each machine in a pure cycle),
D: Dropping the processed part to the output buffer (repeated m times in a pure cycle).

All pure cycles can be expressed as a sequence of a specific number of these activities and there is a correspondence between the states and these activities. Let us consider the consecutive states $(0, 0, e, e)$ and $(0, 0, n, e)$. These states indicate that the robot picks up a part from the input buffer with G_1. This means that the robot performs activity I. Similarly, the consecutive states $(1, 1, e, e)$ and $(0, 1, f, e)$ indicate that the robot performs activity U_1. Since $4m$ states are necessary to express a pure cycle, there are $4m$ corresponding activities to express these pure cycles of in an m-machine cell. For instance, the activity sequence corresponding to the state sequence of cycle C_2^2 presented in Fig. 2 is $I - I - L_1 - L_2 - U_1 - U_2 - D - D$. Now let us consider the following state transitions that correspond to feasible pure cycles:

- $(0, 0, e, e) - (0, 0, e, n) - (0, 0, n, n) - (1, 0, e, n) - (1, 1, e, e) - (0, 1, f, e) - (0, 0, f, f) - (0, 0, e, f)$
- $(0, 0, e, e) - (0, 0, e, n) - (0, 0, n, n) - (1, 0, n, e) - (1, 1, e, e) - (0, 1, e, f) - (0, 0, f, f) - (0, 0, e, f)$
- $(0, 0, e, e) - (0, 0, e, n) - (0, 0, n, n) - (1, 0, n, e) - (1, 1, e, e) - (0, 1, f, e) - (0, 0, f, f) - (0, 0, f, e)$

The corresponding activity sequence to each of these cycles is the same, i.e., $I - I - L_1 - L_2 - U_1 - U_2 - D - D$. This is due to the fact that the activities do not define the movements of the robot explicitly. Such situations occur in pure cycles when both of the grippers have the same status such as (e,e), (n,n), and (f,f) in a given state. Consequently, one sequence of activities corresponds to many feasible permutations of states. However, in the following, we will prove that one of these state sequences dominates the others having the same activity sequence.

Definition 4. Active gripper is the gripper that is used while an activity takes place.

Consider state $(0, 0, e, e)$. If activity "I" is performed, there are two alternative states for the following activity. If it is $(0, 0, n, e)$ then the active gripper is G_1, if it is $(0, 0, e, n)$ then the active gripper is G_2.

Lemma 1. *If the status of the robot's grippers has just reached to one of the states* (e,e), (f,f) *or* (n,n), *then there is an optimal cycle in which the succeeding activity is performed by the active gripper.*

Proof. After any one of these states, both grippers are available for performing the next activity. If the active gripper is not used then an additional θ time units is required to switch the grippers. Depending on the duration of the parallel activities that can be performed simultaneously with switching (i.e., switching and processing on the machines or switching and traveling from one machine to another), this situation may increase the cycle time. However, by using the active gripper this extra switching time can be prevented.□

As a consequence of this lemma, we now know which gripper to use for consecutive activities. This is necessary to determine the activity sequence corresponding to a given state sequence. However, it is not sufficient alone. We also need to know the initial active gripper at the start of the cycle. The following definition is necessary for this purpose.

Definition 5. A **complementary cycle** of a pure cycle is another pure cycle in which the active grippers of each activity is replaced with the other gripper. Therefore, the original and the complementary pure cycles have the same activity sequences but complementary state sequences yielding the same cycle time value.

To clear this definition, let us consider the following two pure cycles.

- $(0, 0, e, e) - (0, 0, n, e) - (0, 0, n, n) - (1, 0, e, n) - (1, 1, e, e) - (0, 1, f, e) - (0, 0, f, f) - (0, 0, e, f)$
- $(0, 0, e, e) - (0, 0, e, n) - (0, 0, n, n) - (1, 0, n, e) - (1, 1, e, e) - (0, 1, e, f) - (0, 0, f, f) - (0, 0, f, e)$

The active gripper sequence of these two cycles are $G1 - G2 - G1 - G2 - G1 - G2 - G1 - G2$ and $G2 - G1 - G2 - G1 - G2 - G1 - G2 - G1$ which are the completely opposite of each other. This means that these two pure cycles are complimentary to each other. Since loading and unloading times are identical for both grippers, the cycle times of one cycle and its complementary cycle are equal to each other. Therefore, from now on we will consider only the cycles that the active gripper of their first activity (transition from state 1 to state 2) is G_1 without loss of generality.

Using Definitions 4 and 5 and Lemma 1, we can further make the following definition which will help us reduce the number of cycles to evaluate in order to find the optimal cycle.

Definition 6. An **active cycle** is the cycle in which the active gripper of its first activity is G_1 and satisfy Lemma 1.

As a consequence, since we know the initial active gripper and gripper transitions until the end of the pure cycle, there is a unique active cycle corresponding to a given activity sequence. For example, consider the sequence of activities $I - I - L_1 - L_2 - U_1 - U_2 - D - D$ for the 2-machine case. Since two "I" activities are consecutive, both grippers as well as both machines are initially empty. That is, the cycle starts with the state $(0, 0, e, e)$. In order to perform the first "I" activity, G_1 is used as a consequence of Definition 6. Consequently, the second state must be $(0, 0, n, e)$. The second activity is again taking a part from the input buffer, which is performed by G_2. Therefore, the third state must be $(0, 0, n, n)$. The robot moves to M_1 to load one of the parts. Since G_2 is the active gripper and it has a part that can be loaded to M_1, as a consequence of Lemma 1, the robot loads the part using G_2. So, the fourth state must be $(1, 0, n, e)$. Then, the robot moves to M_2 to load it using G_1. It returns back to M_1 to unload it. The state is $(1, 1, e, e)$ and both grippers can be used to unload M_1. Since G_1 is the active gripper, the robot unloads M_1 with this gripper. The corresponding state is $(0, 1, f, e)$. The robot moves to M_2 to unload it using G_2, which leads to state $(0, 0, f, f)$. It moves to the output buffer to drop the processed part. Both grippers hold a part to drop the processed parts. Since the active gripper is G_2, the robot first drops the part on this gripper. The corresponding state is $(0, 0, f, e)$. Finally, the robot drops the part on G_1 and returns back to the input buffer.

As a consequence of this analysis, we can state that the optimal pure cycle will be an active cycle. When all non-active cycles are eliminated, the potentially optimal robot move cycles for the 2-machine case reduces to 266.

We can calculate the cycle time of a given cycle by following the activities performed by the robot. Let us calculate the cycle time of C_2^2 illustrated in Fig. 2 as an example. Initially, the robot is in front of the input buffer and picks up a part that takes ε amount of time (states 1 and 2). The robot switches its grippers and picks up another part from the input buffer, $(\theta + \varepsilon)$. It moves to M_1 and load this machine using G_2 $(\delta + \varepsilon)$. It moves to M_2 and loads it using G_1. Since the robot is capable of switching its grippers while traveling, it takes $\max(\theta, \delta) + \varepsilon$ amount of time. The robot moves to M_1, waits until machine completes its processing, and unloads it with G_1 ($\delta + w_1 + \varepsilon$). It moves to M_2 and switches its grippers while moving, waits if necessary, and unloads M_2 with G_2 ($\max(\theta, \delta) + w_2 + \varepsilon$). It moves to the output buffer, drops the part on G_2, switches grippers, drops the part on G_1 ($\delta + \varepsilon + \theta + \varepsilon$). Finally, the robot returns back to the input buffer to complete the cycle (3δ). As a result, the cycle time to produce2 parts becomes $T_{C_2^2} = 8\varepsilon + 6\delta + 2\theta + 2\max(\theta, \delta) + w_1 + w_2$, where $w_1 = \max\{0, P - (\varepsilon + \delta + \max\{\delta, \theta\})\}$ and $w_2 = \max\{0, P - (\varepsilon + \delta + \max\{\delta, \theta\} + w_1)\}$. Since we assumed that $\theta \le \delta$, after some algebraic operations we calculate the final cycle time as $T_{C_2^2} = 8\varepsilon + 8\delta + 2\theta + \max\{0, P - (\varepsilon + 2\delta)\}$.

In the following lemma we derive a lower bound for the cycle time of any pure cycle in an m-machine cell.

Lemma 2. *In an m-machine robotic cell, the cycle time of any feasible pure cycle is no less than T_{LB}^m where,*

$$T_{LB}^m = \begin{cases} \max\{4m\varepsilon + (m+1)^2\delta + m\min\{P, \delta\} + (m-1)\theta, P & \text{if } m \text{ is odd,} \\ \quad + 2\varepsilon + \theta\}, & \\ \max\{4m\varepsilon + m(m+1)\delta + m\min\{P, \delta\} & \text{if } m \text{ is even.} \\ \quad + m\theta, P + 2\varepsilon + \theta\}, & \end{cases}$$

Proof. The first term of the max operator of the lower bound comes from multiple components of the robot's operations in any pure cycle. Firstly, the robot loads and unloads each machine once. It also picks up m parts from the input buffer and drops m parts to the output buffer. These handling operations make a total of $4m\varepsilon$.

After the end of one repetition of the cycle to the end of the succeeding repetition, the robot needs to deliver m parts from the input buffer to the output buffer. Therefore, in any cycle, the robot travels from the output buffer to the input buffer a number of times in which the grippers have a total of at least m empty slots. Bearing in mind that the robot has two grippers, it can carry up to two parts (or up to two empty slots) at a time. In the case that m is even, the robot travels from the output buffer to the input buffer at least $\frac{m}{2}$ times with two empty gripper slots. If m is odd, the robot can travel from the input buffer to the output buffer $\frac{m-1}{2}$ times carrying with two empty gripper slots, and one more time for the residual part making a total of $\frac{m+1}{2}$ times. With a similar approach, we conclude that the robot carries m parts with at least $\frac{m}{2}$ trips from the input buffer to the output buffer if m is even, and with at least $\frac{m+1}{2}$ trips if m is odd. Since the distance between the input buffer and the output buffer is $(m+1)\delta$, these trips make a total of $m(m+1)\delta$ if m is even, and $(m+1)^2\delta$ if m is odd. On the other hand, minimizing the number of visits to the input buffer as suggested above may gain one 2δ per visit but an additional switch time is incurred. Since $\theta \le \delta$, this is beneficial. Therefore, if m is an even number, the total gripper switch time in front of the input buffer is $(m/2)\theta$. The same applies to the output buffer. Consequently, we have $m\theta$ if m is even. When m is odd this switching can occur at most $(m-1)\theta/2$ times since in the final visit only a single part will be picked up or dropped that will not result in a switch time.

Finally, in any pure cycle, after loading a machine the robot either waits in front of it for the part to be processed or travels to another machine, which takes at least $\min\{P, \delta\}$.

The second term in the max operator of the lower bound expresses the minimum time required for two successive loadings of the same machine. After loading a machine, the earliest time that it can be unloaded is after its completion, which requires P amount of time. Then in order to reload this machine again in the earliest possible time, the robot unloads the machine (ε), switches its grippers (θ), and loads the machine (ε). As a result, the total required time is $P + 2\varepsilon + \theta$.□

In the following, section we consider 2-machine robotic cells in detail and determine optimal pure cycles.

3. Optimal pure cycles in 2-machine robotic cells

In robotic cells, the number of machines in the system is usually small. This is especially true in the metal cutting industry where CNC machines are used. Since these machines require a large amount of physical space, usually a small number of machines is used. In parallel with this, most studies in the literature considered 2- or 3-machine cells. In this section, we also consider 2-machine cells in detail.

In the previous section, we have defined a new state definition, and derived a number of feasibility conditions using this new state definition to reduce the search space. After applying these feasibility conditions, the number of feasible pure cycle reduces to 2.674. As it can be seen, the number of feasible cycles is still very large even for only two machines. Therefore, we have introduced a new class of cycles, denoted as active cycles, and have shown that the optimal pure cycle must be an active cycle. Consequently, after eliminating all non-active cycles, we were able to reduce the potentially optimal robot move cycles to be examined from 2.674 to 266. We present an active cycle generation algorithm for an m-machine cell together with a methodology for calculating their cycle times for the 2-machine case in Appendix A. In order to compare these 266 potentially optimal robot move cycles, we initially coded these algorithms in Java programming language and performed an extensive numerical analysis with different parameter combinations of $(P, \varepsilon, \delta, \theta)$. In this analysis, we computed the cycle times of all active cycles and determined the best cycle for each parameter combination. It is important to note that there could be alternative pure cycles giving the best cycle time value for a certain combination. One of our aims is to identify the best ones for the overall problem. In all these scenarios, one of the following five cycles given in Eqs. (1)–(5) appeared to be the optimal cycle. In each equation, we give the activity sequence along with the cycle time value in terms of the problem parameters. Although we have a good starting point, this analysis does not guarantee the optimality of these cycles. Consistent with this observation, in the sequel we will analytically show that depending on the problem parameters, $(P, \varepsilon, \delta, \theta)$, one of these cycles is the optimal cycle for a given parameter combination. As a result, we were able to reduce 266 potentially optimal robot move cycles to five.

$$C_1^2\text{:}\ I - I - L_1 - U_1 - L_2 - U_2 - D - D\text{:}\ 8\varepsilon + 6\delta + 2\theta + 2P \tag{1}$$

$$C_2^2\text{:}\ I - I - L_1 - L_2 - U_1 - U_2 - D - D\text{:}$$
$$8\varepsilon + 8\delta + 2\theta + \max\{0, P - (\varepsilon + 2\delta)\} \tag{2}$$

$$C_3^2\text{:}\ I - I - L_1 - U_2 - L_2 - U_1 - D - D\text{:}$$
$$8\varepsilon + 8\delta + 3\theta + \max\{0, P - (2\varepsilon + 2\delta + \theta)\} \tag{3}$$

$$C_4^2\text{:}\ I - L_1 - I - D - U_2 - D - L_2 - U_1\text{:}$$
$$8\varepsilon + 10\delta + \max\{0, P - (5\varepsilon + 8\delta)\} \tag{4}$$

$$C_5^2\text{:}\ I - U_1 - L_1 - I - D - U_2 - L_2 - D\text{:}$$
$$8\varepsilon + 10\delta + 2\theta + \max\{0, P - (6\varepsilon + 10\delta + \theta)\} \tag{5}$$

In order to determine the optimal cycles, we will use the lower bound derived in Lemma 2. Using this lemma, the lower bound for pure cycles in 2-machine cells can be calculated as follows:

$$T_{LB}^2 = \max\{8\varepsilon + 6\delta + 2\min\{P, \delta\} + 2\theta, P + 2\varepsilon + \theta\}. \tag{6}$$

In the following lemma, we determine the parameter values for which C_1^2, C_2^2, and C_5^2 are optimal by comparing their cycle times with the lower bound.

Lemma 3. *In a 2-machine robotic cell we have the following:*

1. *If $P \le \delta$, C_1^2 is the optimal pure cycle,*
2. *Else if $\delta < P \le \varepsilon + 2\delta$, C_2^2 is the optimal pure cycle,*
3. *Else if $10\delta + 6\varepsilon + \theta \le P$, C_5^2 is the optimal pure cycle.*

Proof.

1. If $P \le \delta$, the lower bound in Equation (6) is equal to $8\varepsilon + 6\delta + 2P + 2\theta$, which is equal to the cycle time of C_1^2 given in Eq. (1).
2. If $\delta < P \le \varepsilon + 2\delta$, the lower bound in Equation (6) is equal to $8\varepsilon + 8\delta + 2\theta$, which is equal to $T_{C_2^2}$ in Equation (2).
3. If $10\delta + 6\varepsilon + \theta \le P$, $T_{C_5^2} = 8\varepsilon + 10\delta + 2\theta + \max\{0, P - (6\varepsilon + 10\delta + \theta)\} = P + 2\varepsilon + \theta = T_{LB}^2$.

This completes the proof.□

With Lemma 3 we have determined the optimal pure cycles for $P \in [0, \varepsilon + 2\delta] \cup [6\varepsilon + 10\delta + \theta, \infty)$. In the following, we examine the remaining region where $P \in (\varepsilon + 2\delta, 6\varepsilon + 10\delta + \theta)$.

3.1. Classification of the 2-machine pure cycles

In the following, we first classify the pure cycles into different categories. Afterwards, we determine the best cycles in each category for all possible parameter values within that category. Finally, we compare the best cycles of different categories with each other to determine the optimal cycle. We use the total travel time of the robot in a cycle as the first classification criterion. As it is mentioned in Equation (6), the robot has to travel at least 6δ in any pure cycle. Any additional movement between the machines will increase this value. Since the robot travels in a cyclic manner any addition to this travel time can be an even multiple of δ (the total travel distance in all pure cycles are even multiples of δ). Therefore, in any pure cycle the travel time of the robot can be expressed as $(6 + 2k)\delta$, where $k = 0, 1, \ldots$.

In the following series of lemmas, we determine the best pure cycle in each category.

Lemma 4. *If $P \in (\varepsilon + 2\delta, 6\varepsilon + 10\delta + \theta)$, then C_5^2 dominates all pure cycles that have a total travel time $(6 + 2k)\delta$ for $k \ge 3$.*

Proof. The cycle time of C_5^2 is given in Equation (5) as $T_{C_5^2} = 8\varepsilon + 10\delta + 2\theta + \max\{0, P - (6\varepsilon + 10\delta + \theta)\}$. Therefore, for $\varepsilon + 2\delta < P < 6\varepsilon + 10\delta + \theta$, $T_{C_5^2} = 8\varepsilon + 10\delta + 2\theta$. Since $\theta \le \delta$, we can write $T_{C_5^2} \le 8\varepsilon + 12\delta$. Since both machines must be loaded and unloaded once, two parts must be picked up from the input buffer, and two parts must be dropped to the output buffer, any pure cycle contains 8ε as the total loading and unloading time. On the other hand, if the travel time is $(6 + 2k)\delta$ and $k \ge 3$, then the minimum cycle time for these cycles is $8\varepsilon + 12\delta$, which is greater than or equal to $T_{C_5^2}$.□

As a consequence of Lemma 4, the categories can be limited to $k = 0, 1, 2$ with respect to the travel time. In the following, we will analyze each of these cases separately;

Case 1: k=0. There is only one feasible pure cycle in this category, which is C_1^2 and its cycle time is given in Equation (1). Therefore, it is the only cycle within this category.

Case 2: k=1. There are several pure cycles in this class where the robot travel time is equal to 8δ. The following remark provides an effective search strategy to find the best cycle within this class.

Remark 1. The cycle times of the pure cycles where k=1, in which the input buffer operations (I) or the output buffer operations (O) are not successive, are not less than $8\varepsilon + 8\delta + P$.

If the robot does not pick up two parts consecutively from the input buffer once it is in front of the input buffer, then it must return back to pick up the second part. This requires at least δ additional time over 6δ. That is, the travel time is at least 7δ in the cycle time expression. On the other hand, after loading a machine, if the robot does not wait in front of it to unload it but moves to another machine to make another activity, this will again require at least δ amount of time. Since we are considering k=1 case, in order not to exceed 8δ, the robot has to wait in front of at least one of the machines. This contributes P amount of time to the cycle time. Therefore, the cycle times of the pure cycles in which two (I) operations are not successive are greater than or equal to $8\varepsilon + 8\delta + P$. The same situation can also be considered for the output buffer operations.

Table 1
Remaining Pure Cycles For k=1 With Successive Input and Output Buffer Operations.

Cycle	Cycle time
$I - I - L_1 - L_2 - U_1 - U_2 - D - D$	$8\varepsilon + 8\delta + 2\theta + \max\{0, P - (\varepsilon + 2\delta)\}$
$I - I - L_1 - U_2 - L_2 - U_1 - D - D$	$8\varepsilon + 8\delta + 3\theta + \max\{0, P - (2\varepsilon + 2\delta + \theta)\}$
$I - I - L_1 - L_2 - U_2 - U_1 - D - D$	$8\varepsilon + 8\delta + 2\theta + P$
$I - I - L_2 - U_1 - L_1 - U_2 - D - D$	$8\varepsilon + 8\delta + 3\theta + \max\{0, P - (2\varepsilon + 2\delta + \theta)\}$
$I - I - L_2 - L_1 - U_1 - U_2 - D - D$	$8\varepsilon + 8\delta + 2\theta + P$
$I - I - L_2 - U_2 - L_1 - U_1 - D - D$	$8\varepsilon + 8\delta + 2\theta + 2P$
$I - I - L_2 - L_1 - U_2 - U_1 - D - D$	not suitable for the class of k=1
$I - I - L_1 - U_1 - L_2 - U_2 - D - D$	not suitable for the class of k=1
$I - I - L_1 - U_1 - U_2 - L_2 - D - D$	not feasible
$I - I - L_1 - U_2 - U_1 - L_2 - D - D$	not feasible
$I - I - L_2 - U_2 - U_1 - L_1 - D - D$	not feasible
$I - I - L_2 - U_1 - U_2 - L_1 - D - D$	not feasible

For the cases that two (I) operations and and two (O) operations are successive, we can assume that the cycles start with activities $I - I$ and ends with activities $D - D$, without losing generality. This is because, when the robot picks up two parts from the input buffer, it must load and then, unload the machines before dropping the parts to the output buffer. As a consequence, the total number of activity sequences is equal to the total number of permutations of $\{L_1; L_2; U_1; U_2\}$, which is equal to 24. However, as the robot picks up two parts from the input buffer, it cannot unload a machine before dropping one of the parts from one of its grippers. Therefore, to satisfy the feasibility, either L_1 or L_2 should follow the $I - I$ pair. As a consequence, 12 pure cycles that are listed in Table 1 remain.

The first activity sequence listed in Table 1 corresponds to the C_2^2 cycle and the second one corresponds to the C_3^2 cycle. The last four activity sequences do not satisfy the feasibility conditions since they try to unload a machine despite both of its grippers are occupied. There are two more cycles listed in this table that are feasible but do not satisfy the k=1 condition. These cycles are investigated under the corresponding category. If the remaining six feasible cycles are compared with each other one can easily conclude that C_2^2 and C_3^2 < dominate the rest.

Lemma 5. *If $P \in [\varepsilon + 2\delta, 6\varepsilon + 10\delta + \theta]$ and k=1, C_2^2 dominates all pure cycles, in which two (I) or two (O) operations are not successive.*

Proof. It is shown in Remark 1 that the minimum cycle time of the pure cycles in which two (I) or two (O) operations are not successive is $8\varepsilon + 8\delta + P$. On the other hand, in the given interval, the cycle time of C_2^2 is equal to $7\varepsilon + 6\delta + 2\theta + P$, which is less than or equal to $8\varepsilon + 8\delta + P$.□

As a consequence, only C_2^2 and C_3^2 cycles remain undominated. These cycles are compared with each other in the following lemma.

Lemma 6. *If $\varepsilon + 2\delta \le P \le \varepsilon + 2\delta + \theta$, C_2^2 dominates all the cycles in the class of k=1. Else if $\varepsilon + 2\delta + \theta < P \le 6\varepsilon + 10\delta + \theta$, C_3^2 dominates dominates all the cycles in the class of k=1.*

Proof. We compare the cycles in the following intervals.

1. If $\varepsilon + 2\delta \le P \le \varepsilon + 2\delta + \theta$, then $T_{C_2^2} = 7\varepsilon + 6\delta + 2\theta + P \le 8\varepsilon + 8\delta + 3\theta = T_{C_3^2}$.
2. If $\varepsilon + 2\delta + \theta < P \le 2\varepsilon + 2\delta + \theta$, then $T_{C_2^2} = 7\varepsilon + 6\delta + 2\theta + P \ge 8\varepsilon + 8\delta + 3\theta = T_{C_3^2}$.
3. $2\varepsilon + 2\delta + \theta < P \le 6\varepsilon + 10\delta + \theta$, then $T_{C_2^2} = 7\varepsilon + 6\delta + 2\theta + P \ge 6\varepsilon + 6\delta + 2\theta + P = T_{C_3^2}$.

This completes the proof.□

Case 3: k=2. In this class, the robot travel time is equal to 10δ. With the following series of lemmas we show that either C_4^2 or C_5^2 is optimal in this class.

Lemma 7. *If $\varepsilon + 2\delta \le P \le 5\varepsilon + 8\delta$, then C_4^2 dominates all the pure cycles in the class of k=2.*

Proof. From Equation (4), $T_{C_4^2} = 8\varepsilon + 10\delta$ for the provided P values, which is equal to the minimum possible cycle time value in this class.□

In order to determine the best cycle in this category, we need a secondary classification criterion. We use the number of gripper switches in a cycle for this purpose. Since the robot is assumed to be capable of switching its grippers while traveling from one machine to another and since we assume $\theta \le \delta$, in some cycles there is no switching time in their cycle time expressions. However, in the following situations there exist switching times.

- The robot picks up two parts from the input buffer consecutively. The activity sequence here is $I - I$.
- The robot drops two parts to the output buffer consecutively. The activity sequence here is $O - O$.
- The robot unloads and loads machine i consecutively in the stated order. The activity sequence here is $U_i - L_i$, $i = 1, 2$.

In these situations the robot cannot avoid the switching time because it has to make the operations in front of the same machine. However, the following lemma bounds the total switching time in a pure cycle.

Lemma 8. *The cycle time expression of any pure cycle can contain at most 3θ.*

Proof. In order to have 4θ in the cycle time expression, all of the four situations stated above $(I - I, O - O, U_1 - L_1, U_2 - L_2)$, must be performed by the robot in the same cycle. This means that at the beginning of the cycle the robot picks up two parts from the input buffer consecutively $(I - I)$. However, at this point, since both grippers are occupied, it cannot unload a machine as the next activity. Therefore, the following activity must be L_i for some i=1 or 2. This prevents the occurrence of one $U_i - L_i$ sequence in the cycle. As a consequence, the cycle time expression cannot contain more than 3θ.□

This lemma proves that the number of switches cannot be greater than three. On the other hand, we can show that it is possible to have 3θ in a pure cycle with an example. Consider C_3^2 for which the activity sequence is $I - I - L_1 - U_2 - L_2 - U_1 - D - D$. Since there are $I - I$, $D - D$, and $U_2 - L_2$ pairs in this cycle, its cycle time given in Equation (3) contains 3θ. Therefore, we can express the classification of the cycles with respect to the number of switches as $t\theta$, $t = 0, 1, 2, 3$.

Lemma 9. *When k=2 and $t \le 1$, the cycle time of the pure cycles is not less than $3\varepsilon + 2\delta + P$.*

Proof. If $t \le 1$, then for at least one of the machines, the robot must perform another activity after unloading it. To perform this activity and return back in front of the same machine takes at least $\varepsilon + 2\delta$. During this time, the machine is idle. On the other hand, this machine is loaded and unloaded in a cycle that takes 2ε amount of time, and processes the part for P amount of time. Therefore, the cycle time must be greater than or equal to $3\varepsilon + 2\delta + P$.□

Lemma 10. *If $5\varepsilon + 8\delta < P \le 6\varepsilon + 10\delta + \theta$, then C_4^2 dominates all the pure cycles in the class of k=2 and $t \le 1$.*

Proof. If $5\varepsilon + 8\delta < P \le 6\varepsilon + 10\delta + \theta$, then $T_{C_4^2} = 3\varepsilon + 2\delta + P$. This value is equal to the lower bound of the cycle times in this category proved in Lemma 9.□.

Lemma 11. *If $5\varepsilon + 8\delta < P \le 6\varepsilon + 10\delta + \theta$, then C_5^2 dominates all the pure cycles in the class of k=2 and $t \ge 2$.*

Proof. In the given interval, $T_{C_5^2} = 8\varepsilon + 10\delta + 2\theta$. This value is a lower bound for the pure cycles when k=2 and $t \ge 2$.□

As a consequence of Lemmas 7–11, if $P \in [\varepsilon + 2\delta, 6\varepsilon + 10\delta + \theta]$, then C_4^2 and C_5^2 are the two nondominated cycles in the class of pure cycles with k=2. The following lemma compares these two cycles with each other.

Lemma 12. *If $\varepsilon + 2\delta \le P \le 5\varepsilon + 8\delta + 2\theta$, then C_4^2 dominates C_5^2 in the class of k=2. Else if $5\varepsilon + 8\delta + 2\theta \le P \le 6\varepsilon + 10\delta + \theta$, then C_5^2 dominates C_4^2.*

Proof. We compare the cycle times of C_4^2 and C_5^2 in the following intervals.

1. If $\varepsilon + 2\delta \le P \le 5\varepsilon + 8\delta$, the dominance of the cycle C_4^2 is shown in Lemma 7.
2. If $5\varepsilon + 8\delta < P \le 5\varepsilon + 8\delta + 2\theta$, then $T_{C_4^2} = 3\varepsilon + 2\delta + P \le 8\varepsilon + 10\delta + 2\theta = T_{C_5^2}$.
3. $5\varepsilon + 8\delta + 2\theta < P \le 6\varepsilon + 10\delta + \theta$, then $T_{C_4^2} = 3\varepsilon + 2\delta + P \ge 8\varepsilon + 10\delta + 2\theta = T_{C_5^2}$.

This completes the proof.□

We now completed analyzing all categories and determined the best cycles within each category for $P \in (\varepsilon + 2\delta, 6\varepsilon + 10\delta + \theta)$. The results indicate that $C_1^2, C_2^2, C_3^2, C_4^2$, and C_5^2 are the five nondominated cycles in this interval. From these cycles, C_1^2 is in k=0 class; C_2^2 and C_3^2 are in k=1 class; and C_4^2 and C_5^2 are in k=2 class. In Lemmas 13–14, we compare these cycles with each other to determine the parameter regions where each of these cycles is optimal.

Lemma 13. *If $P \ge \delta$, then C_2^2 dominates C_1^2.*

Proof. If $P \ge \delta$, then $T_{C_1^2} = 8\varepsilon + 6\delta + 2\theta + 2P \ge 8\varepsilon + 8\delta + 2\theta + \max\{0, P - (\varepsilon + 2\delta)\} = T_{C_2^2}$.□.

As a consequence of Lemma 13, the number of nondominated cycles is reduced to four.

Lemma 14. *When the nondominated cycles C_2^2, C_3^2, C_4^2, and C_5^2 are compared with each other we have the following regions of optimality*:

1. *If $\varepsilon + 2\delta < P \le \min\{\varepsilon + 2\delta + \theta, \varepsilon + 4\delta - 2\theta\}$, then C_2^2 is optimal,*
2. *Else if $\varepsilon + 4\delta - 2\theta < P \le \varepsilon + 2\delta + \theta$, then C_4^2 is optimal,*
3. *Else if $\varepsilon + 2\delta + \theta < P \le 2\varepsilon + 2\delta + \theta$, then If $\theta \le \frac{2}{3}\delta$, then C_3^2 is optimal, Else if $\theta > \frac{2}{3}\delta$, then C_4^2 is optimal,*
4. *Else if $2\varepsilon + 2\delta + \theta < P \le 2\varepsilon + 4\delta - 2\theta$, then C_3^2 is optimal,*
5. *Else if $2\varepsilon + 4\delta - 2\theta < P \le 5\varepsilon + 8\delta + 2\theta$, then C_4^2 is optimal,*
6. *Else if $5\varepsilon + 8\delta + 2\theta < P < 6\varepsilon + 10\delta + \theta$, then C_5^2 is optimal.*

Proof. If $\varepsilon + 2\delta < P \le \varepsilon + 2\delta + \theta$, then from Lemma 6, C_2^2 dominates C_3^2, and from Lemma 12, C_4^2 dominates C_5^2. Therefore, we only need to compare C_2^2 with C_4^2. In this interval, $T_{C_2^2} = 7\varepsilon + 6\delta + 2\theta + P$ and $T_{C_4^2} = 8\varepsilon + 10\delta$. Thus, if $P \le \varepsilon + 4\delta - 2\theta$, then C_2^2 is optimal. Otherwise, C_4^2 is optimal.

If $\varepsilon + 2\delta + \theta < P \le 2\varepsilon + 2\delta + \theta$, then from Lemma 6, C_3^2 dominates C_2^2, and from Lemma 12, C_4^2 dominates C_5^2. In this interval, $T_{C_3^2} = 8\varepsilon + 8\delta + 3\theta$ and $T_{C_4^2} = 8\varepsilon + 10\delta$. Hence, if $\theta \le \frac{2}{3}\delta$, then C_3^2 is optimal. Otherwise, C_4^2 is optimal.

If $2\varepsilon + 2\delta + \theta < P \le 5\varepsilon + 8\delta + 2\theta$, again C_3^2 and C_4^2 are the nondominated cycles as a consequence of Lemmas 6 and 12. $T_{C_3^2} = 6\varepsilon + 6\delta + 2\theta + P$. If $P \le 5\varepsilon + 8\delta$, then $T_{C_4^2} = 8\varepsilon + 10\delta$. Hence, if $P \le 2\varepsilon + 4\delta - 2\theta$, then C_3^2 is optimal. Otherwise, C_4^2 is optimal. On the other hand, if $5\varepsilon + 8\delta < P \le 5\varepsilon + 8\delta + 2\theta$, then $T_{C_4^2} = 3\varepsilon + 2\delta + P$. Therefore, in this interval C_4^2 is optimal.

If $5\varepsilon + 8\delta + 2\theta < P \le 6\varepsilon + 10\delta + \theta$, then C_3^2 and C_5^2 are the nondominated cycles as a consequence of Lemmas 6 and 12. $T_{C_3^2} = 6\varepsilon + 6\delta + 2\theta + P$ and $T_{C_5^2} = 8\varepsilon + 10\delta + 2\theta$. Since $P > 5\varepsilon + 8\delta + 2\theta$, C_5^2 is optimal in this interval.□

Table 2
Comparison of dual gripper and single gripper pure cycles.

Interval		Cycle	LB_{sin}/CT_{opt}	*BC*
$0 \le P \le \delta$		C_1^2	$(8\varepsilon + 12\delta)/(8\varepsilon + 6\delta + 2P)$	2
$\delta < P \le \varepsilon + 2\delta$		C_2^2	$(8\varepsilon + 12\delta)/(8\varepsilon + 8\delta + 2\theta)$	3/2
$\varepsilon + 2\delta < P \le \varepsilon + 2\delta + \theta$		C_2^2	$(8\varepsilon + 12\delta)/(7\varepsilon + 6\delta + 2\theta + P)$	3/2
$0 \le \theta \le (2/3)\delta$	$\varepsilon + 2\delta + \theta \le P \le 2\varepsilon + 2\delta + \theta$	C_3^2	$(8\varepsilon + 12\delta)/(8\varepsilon + 8\delta + 3\theta)$	3/2
	$2\varepsilon + 2\delta + \theta \le P \le 2\varepsilon + 4\delta - 2\theta$	C_3^2	$(8\varepsilon + 12\delta)/(6\varepsilon + 6\delta + 2\theta + P)$	3/2
	$2\varepsilon + 4\delta - 2\theta \le P \le 4\varepsilon + 6\delta$	C_4^2	$(8\varepsilon + 12\delta)/(8\varepsilon + 10\delta)$	6/5
	$4\varepsilon + 6\delta \le P \le 5\varepsilon + 8\delta$	C_4^2	$(4\varepsilon + 6\delta + P)/(8\varepsilon + 10\delta)$	6/5
	$5\varepsilon + 8\delta \le P \le 5\varepsilon + 8\delta + 2\theta$	C_4^2	$(4\varepsilon + 6\delta + P)/(3\varepsilon + 2\delta + P)$	7/5
$(2/3)\delta < \theta \le \delta$	$\varepsilon + 2\delta \le P \le \varepsilon + 4\delta - 2\theta$	C_2^2	$(8\varepsilon + 12\delta)/(7\varepsilon + 6\delta + 2\theta + P)$	3/2
	$\varepsilon + 4\delta - 2\theta \le P \le 4\varepsilon + 6\delta$	C_4^2	$(8\varepsilon + 12\delta)(8\varepsilon + 10\delta)$	9/7
	$4\varepsilon + 6\delta \le P \le 5\varepsilon + 8\delta$	C_4^2	$(4\varepsilon + 6\delta + P)/(8\varepsilon + 10\delta)$	6/5
	$5\varepsilon + 8\delta \le P \le 5\varepsilon + 8\delta + 2\theta$	C_4^2	$(4\varepsilon + 6\delta + P)/(5\varepsilon + 8\delta + P)$	7/5
$5\varepsilon + 8\delta + 2\theta \le P \le 8\varepsilon + 10\delta + \theta$		C_5^2	$(4\varepsilon + 6\delta + P)/(8\varepsilon + 10\delta + 2\theta)$	7/5
$8\varepsilon + 10\delta + \theta \le P$		C_5^2	$(4\varepsilon + 6\delta + P)/(2\varepsilon + \theta + P)$	8/5

Combining the findings in Lemmas 3 and 14, we now provide the main result of this paper in the following theorem. This theorem is one of the main contributions of this paper, in which we determine the optimal pure cycle for all possible parameter values.

Theorem 1. *The following cycles are the optimal ones among other* 2-machine *pure cycles within the corresponding intervals.*

1. *If* $0 < P \le \delta$, *then* C_1^2 *is optimal,*
2. *Else if* $\delta < P \le \varepsilon + 2\delta$, C_2^2 *is the optimal pure cycle,*
3. *Else if* $\varepsilon + 2\delta < P \le \min\{\varepsilon + 2\delta + \theta, \varepsilon + 4\delta - 2\theta\}$, *then* C_2^2 *is optimal,*
4. *Else if* $\varepsilon + 4\delta - 2\theta < P \le \varepsilon + 2\delta + \theta$, *then* C_4^2 *is optimal,*
5. *Else if* $\varepsilon + 2\delta + \theta < P \le 2\varepsilon + 2\delta + \theta$, *then If* $\theta \le \frac{2}{3}\delta$, *then* C_3^2 *is optimal, Else if* $\theta > \frac{2}{3}\delta$, *then* C_4^2 *is optimal,*
6. *Else if* $2\varepsilon + 2\delta + \theta < P \le 2\varepsilon + 4\delta - 2\theta$, *then* C_3^2 *is optimal,*
7. *Else if* $2\varepsilon + 4\delta - 2\theta < P \le 5\varepsilon + 8\delta + 2\theta$, *then* C_4^2 *is optimal,*
8. *Else if* $5\varepsilon + 8\delta + 2\theta < P < 6\varepsilon + 10\delta + \theta$, *then* C_5^2 *is optimal.*
9. *Else if* $P \ge 5\varepsilon + 8\delta + 2\theta$, C_5^2 *is optimal.*

4. Managerial insights

In this section, we compare dual gripper robots with single gripper robots in a 2-machine cell and derive managerial insights. We quantify the increase in the throughput rate by using a dual gripper robot. There are previous studies that compared the performance of the dual gripper robots with single gripper ones in different problem environments. For example, Sethi et al. [21] compared single and dual gripper robots in flowshop type robotic cells assuming 1-unit cycles with additive travel times. Geismar et al. [11] performed a similar analysis on robotic cells with parallel machines and constant travel times and Drobouchevitch et al. [5] compared the performances of a dual gripper robotic cell with a single gripper robotic cell with output buffers on machines. The main reason of comparing single and dual gripper robots is the higher investment cost of dual gripper robots. Note that, a dual gripper robot can perform any activity that a single gripper robot performs. Therefore, all feasible single gripper cycles are also feasible dual gripper cycles. However, there are several dual gripper cycles that require a dual gripper robot and are not feasible for single gripper robots. Therefore, it is straightforward to conclude that dual gripper robots perform better than single gripper ones. However, it is important to quantify this benefit to economically justify the necessary investment cost of dual gripper robots. The reason that different studies performed similar comparisons is that the results are specific to the considered system. The quantified values as well as the corresponding parameter regions are different in different systems. As far as we know, our study is the first one to perform this analysis for robotic cells with flexible machines. In the following, we first prove that the dual gripper pure cycles dominate the single gripper ones. Afterwards, we quantify their benefits for different parameter values. In this analysis, we will make use of the following lower bound on the cycle time of single gripper pure cycles.

Table 3
Factors and levels.

Factor	Low	Medium	High
P/δ	0.1	5.0	10.0
θ/δ	0.1	0.5	1.0
θ/ε	0.1	1	2

Remark 2 ([15]). For a 2-machine robotic cell, the cycle time of any single gripper pure cycle is no less than $LB_{sin} = \max\{8\varepsilon + 12\delta, 4\varepsilon + 6\delta + P\}$.

The following lemma proves that the dual gripper cycles that utilize both grippers of the robot always perform better than the single gripper cycles.

Theorem 2. The dual gripper pure cycle C_4^2 *dominates all single gripper pure cycles.*

Proof.

1. If $P \le 5\varepsilon + 8\delta$, $T_{C_4^2} = 8\varepsilon + 10\delta \le \max\{8\varepsilon + 12\delta, 4\varepsilon + 6\delta + P\} = LB_{single}$,
2. Else if $5\varepsilon + 8\delta < P$, $T_{C_4^2} = 3\varepsilon + 2\delta + P \le 4\varepsilon + 6\delta + P = LB_{single}$.□

As a consequence of this theorem we know that dual gripper robots perform better than single gripper ones. However, in order to conduct an economical analysis we have to quantify the possible reduction in cycle times. For this purpose, we first compare the optimal dual gripper cycles of the previous section with the lower bound of the single gripper cycle time. This will provide us the best case performance of dual gripper cycles.

Table 4
Summary of the computational study.

P/δ	θ/δ	θ/ε	**CT$_{op}$**	**LB$_{sin}$**	**LB$_{sin}$/CT$_{opt}$**
L	L	L	144	200	1.39
		M	72	128	1.78
		H	68	124	1.82
	M	L	94.4	104	1.10
		M	22.4	32	1.43
		H	18.4	28	1.52
	H	L	88.2	92	1.04
		M	16.2	20	1.23
		H	12.2	16	1.31
M	L	L	172	200	1.16
		M	108	134	1.24
		H	104	132	1.27
	M	L	98	104	1.06
		M	28	32	1.14
		H	24	28	1.17
	H	L	90	92	1.02
		M	18	20	1.11
		H	14	16	1.14
H	L	L	180	220	1.22
		M	110	184	1.67
		H	106	182	1.72
	M	L	99	104	1.05
		M	28	40	1.43
		H	25.5	38	1.49
	H	L	90	92	1.02
		M	18	22	1.22
		H	14	20	1.43
Average					**1.30**

This analysis is given in Table 2. In this table we calculate the ratio of LB_{sin} to the optimal dual gripper cycle denoted by CT_{opt}. Since we want to calculate the best case performance denoted as BC, we take the maximum value of this ratio, $BC = \max\{\frac{LB_{sin}}{CT_{opt}}\}$. In order to calculate the maximum, we use the limiting values of the parameters. As an example let us consider the case where $0 \leq P \leq \delta$. In this interval we have

$$\frac{LB_{sin}}{CT_{opt}} = \frac{8\varepsilon + 12\delta}{8\varepsilon + 6\delta + 2P}.$$

This ratio takes its maximum value for the following parameter values.

$$BC = \lim_{\substack{\delta\to\infty\\ \varepsilon\to 0\\ \theta\to 0\\ P\to 0}} \frac{8\varepsilon + 12\delta}{8\varepsilon + 6\delta + 2P} = \frac{12}{6} = 2. \tag{7}$$

For each interval in this table, a similar analysis is performed.

As it is seen in Table 2, a dual gripper cycle can perform two times better than single gripper cycles in the best case. In order to quantify the average performance improvement of using a dual gripper cycle, we performed a computational study. We set up a design of experiment, in which 3 different factors that can have an effect on the cycle times are used. These are P/δ, θ/δ, and θ/ε. Since rather than the magnitudes of the parameters, their ratios affect the cycle times, we use these ratios in the computational study.

We used 3 different levels for each of these factors. Since $\theta \leq \delta$ is assumed, the maximum value of θ/δ is 1. In Table 3, the factors and their levels are listed. For each generated data, the optimal dual pure cycle time, the lower bound of single gripper cycles, and their ratio are calculated. The results of this computational study are summarized in Table 4. The minimum, average, and maximum performance improvements of dual gripper robots are appeared to be 1.02, 1.3, and 1.82, respectively. It can also be observed that, as the θ/δ ratio increases, the benefits of dual grippers reduces. This is meaningful because, the time advantage of dual gripper robots reduces if gripper switch time increases. On the other hand, as the θ/ε ratio increases, the benefits of dual gripper robots also increase. This analysis provides the preferable parameter values for an investment in dual gripper robots.

5. Conclusion

In this study, we consider a manufacturing cell consisting of a number of flexible machines and a material handling robot that produces identical parts. The robot is assumed to have dual grippers. The flexibility of the machines leads to a new class of robot move cycles, named as pure cycles, in which each part is completely produced on one of the machines. To the best of the author's knowledge, this study is the first one to consider pure cycles together with dual gripper robots. The problem is to determine the optimal pure cycle that maximizes the system throughput. We first develop the necessary framework to analyze this problem. We analyze 2-machine robotic cells in detail and prove that despite the large number of cycles, only five of them are potentially optimal. We identify the parameter values for which each cycle is optimal. We also provide managerial insights by comparing the performances of dual gripper and single gripper cycles. We show that the throughput rate of a dual gripper robot can be at most twice better than a single gripper robot depending on the problem parameters. We also perform a computational study using different parameter values, which demonstrates that the average and maximum performance improvements of dual gripper robots on single gripper robots are 1.3 and 1.82, respectively.

This study contributes to the literature by developing a general framework for analyzing pure cycles in dual gripper robotic cells that can be utilized in more complex problems such as robotic cells with larger number of machines. The approach used in this study forms an analytical basis for finding the optimal pure cycles corresponding to given input parameters. Moreover, by quantifying the maximum possible and average benefits of using a dual gripper robot rather than a single gripper one, this study supports management decisions on investing in dual-gripper robots.

This study can be extended by considering multiple part types. In this case, the robot sequencing and part sequencing problems must be solved simultaneously which increases the problem complexity. Another direction for future research is to analyze robotic cells with larger number of machines. However, the number of feasible pure cycles increases drastically as the number of machines increases. Mathematical programming based exact or heuristic solution procedures can be developed.

Appendix A. Active cycle generation

There are a total of 4 m activities in an active cycle (one L and one U activity for each of the m machines plus m copies of I and D activities). To derive all active cycles for an m-machine cell, we first find all possible permutations of activities. Considering the cyclic permutations and eliminating the repeated cycles resulted from m copies of I and D, the total number of all permutations can be calculated as $\frac{(4m-1)!}{m!m!}$. Note that not all circular permutations of the activities are active cycles. Algorithm 1 derives all active cycles from the set of all circular activity permutations using the following conditions:

- Before an unloading (I or U), at least one of the grippers must be empty.
- Before a D activity, the robot must have at least one processed part in one of its grippers.
- Before a L activity, the robot must have at least one new part in one of its grippers.

Algorithm 1. Algorithm for deriving all active cycles.

Using Algorithm 1, we calculate the total number of active cycles to be 266. Once we generate all active cycles, we can calculate their cycle times using the matrix in Table A1, which shows the necessary time the robot must spent to perform an activity succeeding a particular activity. However, w_j, the waiting time before unloading machine j, is not constant, and hence it depends on the sequence of the activities in the active cycle. Therefore, to be able derive the cycle time expression in terms of the problem parameters, we must derive w_j. For the 2-machine case, Table A2 illustrates the expressions for w_1 and w_2 for different orderings of L_1, L_2, U_1, and L_2. In these expressions, the empty spaces, (...), are filled with the suitable number of I and D activities considering the feasibility conditions to have an active cycle. Since the schedules are cyclic, without loss of generality we assume L_1 activity to precede L_2, U_1, and L_2 activities in any cycle. T_1 (T_2) denotes the total time of activities except any waiting time in between the

Table A1
Necessary time between pair of activities.

	L_j	U_j	I	D
L_i	$\lvert i-j\rvert\delta+\varepsilon$	$\lvert i-j\rvert\delta+w_j+\varepsilon$	$i\delta+\varepsilon$	$(m+1-j)\delta+\varepsilon$
U_i	$\max\{\lvert i-j\rvert\delta,\ \theta\}+\varepsilon$	$\lvert i-j\rvert\delta+w_j+\varepsilon$	$i\delta+\varepsilon$	$(m+1-j)\delta+\varepsilon$
I	$j\delta+\varepsilon$	$j\delta+w_j+\varepsilon$	$\theta+\varepsilon$	$m\delta+\varepsilon$
D	$(m+1-j)\delta+\varepsilon$	$(m+1-j)\delta+w_j+\varepsilon$	$m\delta+\varepsilon$	$\theta+\varepsilon$

Table A2
Patterns of the L_1, L_2, U_1, and U_2 in all active cycles and corresponding waiting time expressions.

	Pattern	$\mathbf{w}_1$	$\mathbf{w}_2$
1	$\ldots-\mathbf{L_1}-\ldots-\mathbf{L_2}-\ldots-\mathbf{U_1}-\ldots-\mathbf{U_2}-\ldots$	$\max\{P-T_1,\ 0\}$	$\max\{P-T_2-w_1,\ 0\}$
2	$\ldots-\mathbf{L_1}-\ldots-\mathbf{L_2}-\ldots-\mathbf{U_2}-\ldots-\mathbf{U_1}-\ldots$	$\max\{P-T_1-w_2,\ 0\}$	$\max\{P-T_2,\ 0\}$
3	$\ldots-\mathbf{L_1}-\ldots-\mathbf{U_1}-\ldots-\mathbf{L_2}-\ldots-\mathbf{U_2}-\ldots$	$\max\{P-T_1,\ 0\}$	$\max\{P-T_2,\ 0\}$
4	$\ldots-\mathbf{L_1}-\ldots-\mathbf{U_1}-\ldots-\mathbf{U_2}-\ldots-\mathbf{L_2}-\ldots$	$\max\{P-T_1,\ 0\}$	$\max\{P-T_2-w_1,\ 0\}$
5	$\ldots-\mathbf{L_1}-\ldots-\mathbf{U_2}-\ldots-\mathbf{L_2}-\ldots-\mathbf{U_1}-\ldots$	$\max\{P-T_1-w_2,\ 0\}$	$\max\{P-T_2-w_1,\ 0\}$
6	$\ldots-\mathbf{L_1}-\ldots-\mathbf{U_2}-\ldots-\mathbf{U_1}-\ldots-\mathbf{L_2}-\ldots$	$\max\{P-T_1-w_2,\ 0\}$	$\max\{P-T_2,\ 0\}$

```
Permutations: Set of all possible circular permutations of activities.
ActiveCycles: Set of active cycles derived by the algorithm. Set ActiveCycles = ∅
gStatus=[gs1, gs2]: gs1 is the number of new parts and gs2 is the number of processed parts on robot's grippers. Since we assume all cycles
  start with "I" activity, gStatus can be equal to [0,0], [0,1], or [1,0] at the beginning of any active cycle.
for $perm \in Permutations$ do
    for $[gs1, gs2] \in \{[0,0], [0,1], [1,0]\}$ do
        $cycleFeasible = True$
        for $activity \in perm$ do
            if $activity$ is "I" then
                gs1 ← gs1 + 1
            else if $activity$ is "L" then
                gs1 ← gs1 − 1
            else if $activity$ is "U" then
                gs2 ← gs2 + 1
            else
                (if $activity$ is "D" gs2 ← gs2 − 1
            end
            if $gs1 < 0$ or $gs1 < 0$ or $(gs1 + gs2) > 2$ then
                $cycleFeasible = False$
                break;
            end
        end
        if $cycleFeasible = True$ then
            $ActiveCycles = ActiveCycles \cup \{perm\}$
        end
    end
end
```

L_1 (L_2) activity and the U_1 (U_2) activity.

As a consequence, the cycle time of an active cycle for the 2-machine case can be calculated using Tables A1 and A2.

References

[1] M.S. Akturk, H. Gultekin, O.E. Karasan, Robotic cell scheduling with operational flexibility, Discret. Appl. Math. 145 (3) (2005) 334–348.

[2] N. Brauner, G. Finke, Cycles and permutations in robotic cells, Math. Comput. Model. 34 (5–6) (2001) 565–591.

[3] Y. Crama, J. van de Klundert, Cyclic scheduling of identical parts in a robotic cell, Oper. Res. 45 (6) (1997) 952–965.

[4] M. Dawande, H.N. Geismar, S.P. Sethi, C. Sriskandarajah, Sequencing and scheduling in robotic cells: recent developments, J. Sched. 8 (5) (2005) 387–426.

[5] I.G. Drobouchevitch, S.P. Sethi, C. Sriskandarajah, Scheduling dual gripper robotic cell: one-unit cycles, Eur. J. Oper. Res. 171 (2) (2006) 598–631.

[6] E. Edis, C. Oguz, I. Ozkarahan, Parallel machine scheduling with additional resources: notation, classification, models and solution methods, Eur. J. Oper. Res. 230 (3) (2013) 449–463.

[7] M. Foumani, K. Jenab, Cycle time analysis in reentrant robotic cells with swap ability, Int. J. Prod. Res. 50 (22) (2012) 6372–6387.

[8] M. Foumani, K. Jenab, Analysis of flexible robotic cells with improved pure cycle, Int. J. Comput. Integr. Manuf. 26 (3) (2013) 201–215.

[9] M. Foumani, M.Y. Ibrahim, I. Gunawan, Scheduling dual gripper robotic cells with a hub machine. In Industrial Electronics (ISIE), 2013 IEEE International Symposium on, (2013) pp. 1–6. IEEE.

[10] G. Galante, G. Passannanti, Minimizing the cycle time in serial manufacturing systems with multiple dual-gripper robots, Int. J. Prod. Res. 44 (4) (2006) 639–652.

[11] H.N. Geismar, M. Dawande, C. Sriskandarajah, Throughput optimization in constant travel-time dual gripper robotic cells with parallel machines, Prod. Oper. Manag. 15 (2) (2006) 311–328.

[12] H.N. Geismar, L. Mui, A. Chan, M. Dawande, C. Sriskandarajah, Approximations to optimal k-unit cycles for single-gripper and dual-gripper robotic cells, Prod. Oper. Manag. 17 (5) (2008) 551–563.

[13] H.N. Geismar, M. Pinedo, C. Sriskandarajah, Robotic cells with parallel machines and multiple dual gripper robots: a comparative overview, IIE Trans. 40 (12) (2008) 1211–1227.

[14] H. Gultekin, M.S. Akturk, O.E. Karasan, Scheduling in robotic cells: process flexibility and cell layout, Int. J. Prod. Res. 46 (8) (2008) 2105–2121.

[15] H. Gultekin, O.E. Karasan, M.S. Akturk, Pure cycles in flexible robotic cells, Comput. Oper. Res. 36 (2) (2009) 329–343.

[16] N.G. Hall, H. Kamoun, C. Sriskandarajah, Scheduling in robotic cells: classification, two and three machine cells, Oper. Res. 45 (3) (1997) 421–439.

[17] N.G. Hall, C.N. Potts, C. Sriskandarajah, Parallel machine scheduling with a common server, Discret. Appl. Math. 102 (3) (2000) 223–243.

[18] K. Hasani, S. Kravchenko, F. Werner, Simulated annealing and genetic algorithms for the two-machine scheduling problem with a single server, Int. J. Prod. Res. 52 (13) (2014) 3778–3792.

[19] K.S. Jung, H.N. Geismar, M. Pinedo, C. Sriskandarajah, Approximations to optimal sequences in single-gripper and dual-gripper robotic cells with circular layouts, IIE Trans. 47 (6) (2015) 634–652.

[20] S.P. Sethi, C. Sriskandarajah, G. Sorger, J. Blazewicz, W. Kubiak, Sequencing of Parts and Robot Moves in a Robotic Cell, Int. J. Flex. Manuf. Syst. 4 (1992) 331–358.

[21] S.P. Sethi, J.B. Sidney, C. Sriskandarajah, Scheduling in dual gripper robotic cells for productivity gains, IEEE Trans. Robot. Autom. 17 (3) (2001) 324–341.

[22] S. Yildiz, O.E. Karasan, M.S. Akturk, An analysis of cyclic scheduling problems in robot centered cells, Comput. Oper. Res. 39 (6) (2012) 1290–1299.

[23] J. Zhang, X. Fang, L. Qi, Sensitivity-analysis based method in single-robot cells cost-effective design and optimization, Robot. Comput.-Integr. Manuf. 38 (2016) 9–15.