The Multiple Part Type Cyclic Flow Shop Robotic Cell Scheduling Problem: A Novel and Comprehensive Mixed Integer Linear Programming Approach

Atabak Elmi[a], Asef Nazari[b,*], Dhananjay Thiruvady[a]

[a]*School of Mathematical Sciences, Monash University, Australia*
[b]*School of Information Technology, Deakin University, Australia*

Abstract

This paper considers the problem of cyclic flow shop robotic cell scheduling deploying several single and dual gripper robots. In this problem, different part types are successively processed on multiple machines with different pickup criteria including free pickup, pickup within time-windows and no-waiting times. The parts are transported between the machines by the robots. We propose a novel mixed integer programming model which simultaneously determines the optimal sequence of the parts for the cyclic schedule and the optimal sequencing of the robots' movements, which in turn maximises the throughput rate. The proposed mathematical model is validated on a number of randomly generated test instances. These problem instances are constructed by varying the number of machines, part types, robots and gripper options. This allows a detailed analysis of the model including how it scales with increasing numbers of machines, part types, robots and grippers. The experimental results show that model generates good feasible solutions for up to 20 machines and 10 part types.

Keywords: Scheduling, Mixed Integer Programming, Cyclic Flow Shop, Multiple Part Type, Multiple Robots, Single/Dual Gripper.

Introduction

In this study, we focus on the sequencing and scheduling of an automated material handling system in cellular manufacturing, namely, cyclic flow shop robotic cells. The aim of the problem is to minimise the cycle time while sequencing a number of parts (same or multiple types) to be processed repetitively. The

*Corresponding author
Email addresses: atabak.elmi@monash.edu (Atabak Elmi), asef.nazari@deakin.edu.au (Asef Nazari), dhananjay.thiruvady@monash.edu (Dhananjay Thiruvady)

parts enter the system via an input device. They are then processed on a number of machines, in sequence, and exit the system via an output device. To move between machines, a crew of identical programmable robots are used. There is no buffer for intermediate storage between machines, and hence, each machine can hold only one part at a time. Thus, the robotic cell scheduling problem is, in essence, a cyclic flow shop problem with blocking constraints. The cyclic aspect is introduced due to the requirements of processing a number of parts repetitively in order to achieve a production target. In doing so, robots (single or dual gripper) perform a series of material handling operations, also in a repetitive manner. Due to customer demands and competition, manufacturers are gradually improving their production systems to make them capable of processing multi-type parts in small batches concurrently. The production requirements of multi-type parts are usually partitioned into identical copies of the smallest set, called the minimal part set (MPS) [1]. In each production cycle, a single MPS is sent into a production line and exactly one MPS is completed and unloaded from it. The MPS consists of multiple-part types, and hence, a sequence of parts to be processed must be identified.[1] In order to satisfy customers, the MPS is repeatedly processed (with a reduced cycle time) thereby maximising throughput. Another aspect complicating the problem is *blocking*. This occurs when a part has been processed on a machine but another part on a subsequent machine has not been finished. Hence the part on the first machine must wait until the subsequent machine becomes free. This is important, since the restrictions introduced by the blocking constraints can significantly increase the cycle time. Hence, dealing with blocking efficiently is of vital importance. Although in real systems MPSs need to be scheduled, previous studies have been confined to developing single part type cyclic schedules in robotic systems. This study proposes a novel comprehensive mixed integer programming (MIP) model that simultaneously sequences multiple parts of the cycle and finds the optimal sequence of robots' relocations corresponding to a cyclic schedule. In doing so, optimal sequences generated by the model also maximise the *throughput rate*.[2] Additionally, the gripper options (single or dual) and collision constraints are incorporated within the model. Given the processing requirements, the objective of our model is the productivity maximisation of cells, which is in accordance with manufacturers' interests.

[1] Compared to a multiple-part type schedule, in a single type cyclic schedule each part is completed when the robot or robots complete the sequence of relocations without sequencing the parts.

[2] A natural and widely used measure of productivity in robotic cells is the throughput rate defined as the number of finished parts produced per unit of time as mentioned in Dawande et al. [2].

1. Literature Review

Basic robotic flow shop scheduling with time windows has been studied in depth [3, 4, 5, 6]. However, most previous work considers the robotic flow shop scheduling problem with a single part-type. For example, Sethi et al. [3] analytically study the flow shop robotic cell consisting of two and three machines with a single robot for one-degree cyclic schedules. Logendran and Sriskandarajah [4] consider different configurations of cells and more general time-frames for the actions of robots in the single and different part type systems. Crama and Van De Klundert [5] and Dawande et al. [6] develop polynomial-time algorithms to find the optimal one-unit cycles considering the number of machines. Agnetis [7] study a more complex variant of the problem, the two- and three-machine one-degree no-wait cyclic robotic cells with a single robot. They develop polynomial algorithms which are able to find good solutions to the problem. Levner et al. [8] further extended the same problem to *m*-machines using the notion of prohibited intervals. They also consider the usage of more than one robot in large robotic cells in order to eliminate material handling bottlenecks. Moreover, Liu and Jiang [9] considered one-degree cyclic scheduling of two hoists in a no-wait cell. They propose a polynomial algorithm that identifies thresholds for the cycle time and then checks feasibility of solution in increasing order of time. Many researchers have noted and verified that multi-degree cyclic schedules usually have a larger throughput than one-degree cyclic schedules. Kats et al. [10] studied the *k*-degree cyclic robotic cells with a single robot, and they propose a sieve method based algorithm to optimize the cycle time. On the basis of their approach, Che et al. [11] propose an exact algorithm using a branch-and-bound procedure for no-wait robotic cells. They note that the algorithm is polynomial for a fixed *k*-degree ($k > 1$), but becomes exponential if k is assumed arbitrary. More recently, Che and Chu [12] show that the no-wait multi-degree cyclic robotic cell scheduling problem with two parallel robots can be also solved in polynomial time for a fixed *k*-degree ($k > 1$), but the complexity of the problem is exponential if k is arbitrary. However, this algorithm cannot be mechanically extended to the multi-robot case because it explores some properties specific to cells with two robots. Che and Chu [13] consider the *k*-degree cyclic flow shop robotic cell scheduling problem with multiple robots in a no-wait system. They propose a polynomial algorithm to find the minimum number of robots for all *k*-degree ($k > 1$) cyclic schedules and consequently the optimal cycle time for any given number of robots. On the basis of this study, Che et al. [14] extend the method to the cyclic scheduling of a no-wait re-entrant robotic flow shop with multiple robots. Interestingly, research related to the minimal number of robots first appeared in the seventies. For example, Karzanov and Livshits [15] study a system with parallel tracks where robots run along their respective tracks, and they propose an algorithm to find the minimal number of robots for a given cycle time. Zhou et al. [16] consider a different approach by proposing a MIP model for the fixed k-degree cyclic flow shop robotic cells with a single robot considering time window constraints. Brauner et al. [17] consider the single-robot cyclic cell scheduling problem with processing time windows and non-Euclidean robot travel times. They also propose a MIP model to minimize the cycle time and robot empty travel times. Hasani et al. [18] investigate the cyclic project scheduling problem and suggest that their proposed graph algorithm can be extended to solve the multi-degree and multiple-robot cyclic scheduling problems. Che et al. [19] and Zhou et al. [16] propose a branch-and-bound algorithm and a MIP approach, respectively, for the so-called multi-degree cyclic robotic flow shop scheduling problem with time window constraints. Here, multiple parts of the same type enter and leave the system during a cycle. Kats et al. [20] and Kats and Levner [21] propose polynomial algorithms for this simplified version of the problem with given robot move sequences. Other attempts to formulate the problem considered here as a MIP model are studies by El Amraoui et al. [22] and Kats [23]. Kats [23] considered the single-track multi-hoist cyclic scheduling problem with a fixed k-degree. Both approaches minimize the cycle time while El Amraoui et al. [22] study the fixed k-degree cyclic single-hoist scheduling problem for heterogeneous part jobs. Kogan and Levner [24] propose a branch-and-bound algorithm for the cyclic scheduling of a robotic flow shop with multiple part types and processing time windows. Moreover, in two different studies, Elmi and Topaloglu [25, 26] investigate the hybrid flow shop robotic cell scheduling problem with one robot and multiple robots, respectively. They propose MIP models and a simulated annealing based solution to minimize the makespan for the processing of multiple part types. These approaches assign the part types to the machines at each stage and sequence the part types along with the robots' transportation operations simultaneously to minimize the makespan. All the above studies handled the robotic scheduling problem with a single part type or non-cyclic multiple part types. So far, little work has been done for cyclic scheduling of the robotic flow shop problem with multiple part-types due to its significant complexity. Lei and Liu [27] developed a branch-and-bound algorithm for the scheduling problem with two part-types. Later, Mateo and Companys [28] and El Amraoui et al. [29] also proposed a branch-and-bound algorithm and a MIP approach for the same problem as Lei and Liu [27], respectively. Recently, El Amraoui et al. [30] extended their MIP approach for the production line with a complex configuration. For a problem with more than two part-types, Varnier and Jeunehomme [31] constructed a feasible multi-part robot move schedule through a combination of single-part robot move schedules. To reduce the complexity of the problem with multiple part types, a few studies address the problem so that either the part input sequence or the robot's movement sequence is given. For instance, Hindi and Fleszar [32], and Paul et al. [33] proposed heuristic algorithms for the problem with given part input sequence with an objective minimizing the makespan. More thorough reviews of the cyclic robotic flow shop cells can be found in surveys by Hall [34], Crama et al. [35], Dawande et al. [36], and in the book by Dawande et al. [2]. To the best of the authors' knowledge, there is no comprehensive mathematical modeling formulation in the literature for the cyclic flow shop robotic cell scheduling (CFS-RCS) problem. In this research, we investigate the following cyclic robotic cell scheduling problem depicted in Figure 1. We formulate the CFS-RCS considering the number of robots, either single or multiple, and different gripping options. The pickup criteria can be free, time-window or no-wait. Also, we examine travel times from different perspectives including additive, constant or Euclidian distance based travel times. Moreover, most of the cyclic problems based on the settings illustrated in Figure 1 have not been investigated in the literature and, for those that are, there are no proposed mathematical models due to the complexity of the problems.

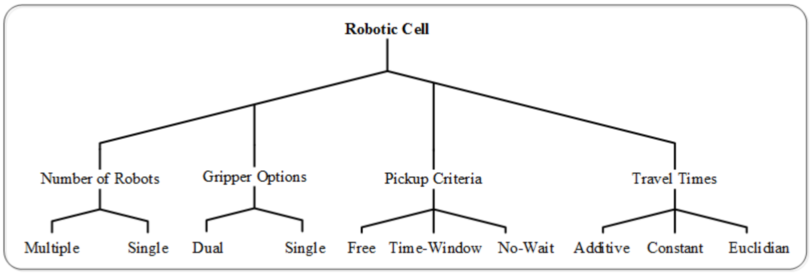

FIGURE 1: Real life robotic configurations considered in the proposed mathematical model.

2. A Comprehensive Mixed Integer Program

The robotic flow shop problem consists of multiple computer-controlled robots for material handling. There are M machines placed in series ($m \in \{m_1, m_2, \dots, m_M\}$), where m_1 and m_M are the loading and unloading stations, respectively. The parts are of the single or multiple types and are processed in the robotic flow shop by visiting the machines consecutively. The flow of parts can be described as follows; After a part is moved into the robotic flow shop from the loading station m_1, it is processed sequentially through machines $m \in \{m_2, m_3, \dots, m_{M-1}\}$, and finally leaves the flow shop from the unloading station m_M. In the multi-part cyclic flow shop problem, the parts of the same type have the same conditions in consecutive cycles. In other words, the parts that would be processed in a cycle are indexed as types $\{1, 2, \dots, p\}$. In contrast to the single part type, the sequence of parts have to be determined in multiple part type to minimize the cycle time. Due to the cyclic processing requirements, some of the consecutive processing operations of parts may continue in the upcoming cycles. This means that only one part from the same type is completed per cycle. The following cyclic flow shop robotic cell characteristics underpin the problem considered here:

- Each machine can process only one part at a time.
- The parts cannot be processed on more than one machine at the same time.
- There is no intermediate buffer between the machines.
- Each part is transported to the next machine based on the pickup criterion after completing its processing on the previous machine.
- Each single or dual gripper robot can transport only one part at a time.
- Each transportation operation is performed by only one robot.
- The time required for the robots to transport the parts between machines and for their empty moves are deterministic.

Carrying part j from machine m to machine $m+1$ by a robot is defined as a transportation operation $[j,m]$. A transportation operation $[j,m]$ is made up of three simple robot sub-operations: (1) unloading part j from machine m, (2) transporting part j to the next machine $m+1$, and (3) loading it onto machine $m+1$. Thus, the total number of transportation operations in a cyclic schedule with J parts is $J \times (M-1)$. Under this consideration, we show the time required for any robot to perform the transportation operation $[j,m]$ (such an action is also called a complete move) as θ_m, and the time required for any robot to travel from machine m to machine n without carrying any part (such a travel is also called a void move $\llbracket m,n \rrbracket$) as $\delta_{\llbracket m,n \rrbracket}$. Note that a cyclic schedule can be described by the starting times of all moves in the system. All the robots are considered to be free at the start of a cycle, whereas the processing machines might be free or occupied at the start time of the cycle. The following sets, parameters and variables are used to build the mathematical model of the problem.

Indices and sets:

J The number of parts that are processed during a cycle, $j \in \{1,2,\ldots,J\}$,

M The number of machines including input and output devices, $m \in \{1,2,\ldots,M\}$,

R The number of robots, $r \in \{1,\ldots,R\}$,

$[j,m]$ A complete robot move to transport a part of type j from machine m to machine $m+1$,

$\llbracket m,n \rrbracket$ A void robot move to travel from machine m to machine n without carrying any part.

Parameters:

$\mathrm{PLo}_{j,m}$ The lower bound of the processing time of part j on machine m,
assuming it is equal to zero for all the parts at the input and output devices,

$\mathrm{PUp}_{j,m}$ The upper bound of the processing time of part j on machine m,
assuming it is equal to zero for all the parts at the input and output devices,

θ_m Time required for a robot to perform a complete move $[j,m]$

$\delta_{[\![m,n]\!]}$ Time required for a robot to perform a void move $[\![m,n]\!]$

G_r 1 if the r^{th} robot is a single gripper, 2 if it is a dual gripper, $r \in \{1,\ldots,R\}$,

$\mathscr{M}$ A sufficiently large positive value.

Decision variables:

$X_{j,j'}$ 1 if part j immediately precedes part j'; Otherwise 0,

$A_{j,m}$ 1 if machine m is occupied by part j at the start of the cycle; Otherwise 0 ,

$B_{j,m}$ 0 if the subsequent part of j^{th} part starts its processing operation on machine
m after the j^{th} part within the cycle; Otherwise 1,

$Z_{[j,m],r}$ 1 if a complete move $[j,m]$ is performed by the r^{th} robot; Otherwise 0,

$Y_{[j,m],[j',m']}$ 1 if a complete move $[j,m]$ is performed before another complete move
$[j',m']$; Otherwise 0,

$S_{[j,m]}$ Starting time of a complete move $[j,m]$,

T The total cycle time for all the parts that have entered the system.

In the following section, we provide details of the objective function and constraints of the CFS-RCS problem.

2.1. Sequence of Parts

Due to the cyclic characteristics, when a machine is occupied at the start of the cycle, the sequence of the parts changes cyclically once. Figure 2 illustrates an example Gantt chart solution for a problem with 3 part types and 12 machines including input and output devices. As illustrated in Figure 2 the sequence of parts on machines 1 to 3 is {1,2,3}. Meanwhile, machine 4 is occupied by part type 3 at the start time of the cycle. Therefore, the sequence of part types changes cyclically as {3,1,2} until the next occupied machine. Moreover, the sequence of part types has changed to {2,3,1} and {1,2,3} on occupied machines 5 and 6, respectively. It is obvious from Figure 2 that the cyclic sequence of part types would be changed at the start time of the cycle if machine m is occupied. As mentioned earlier, the sequence of the parts changes cyclically at each machine that is occupied at the start of the cycle. Therefore, the immediate successor of

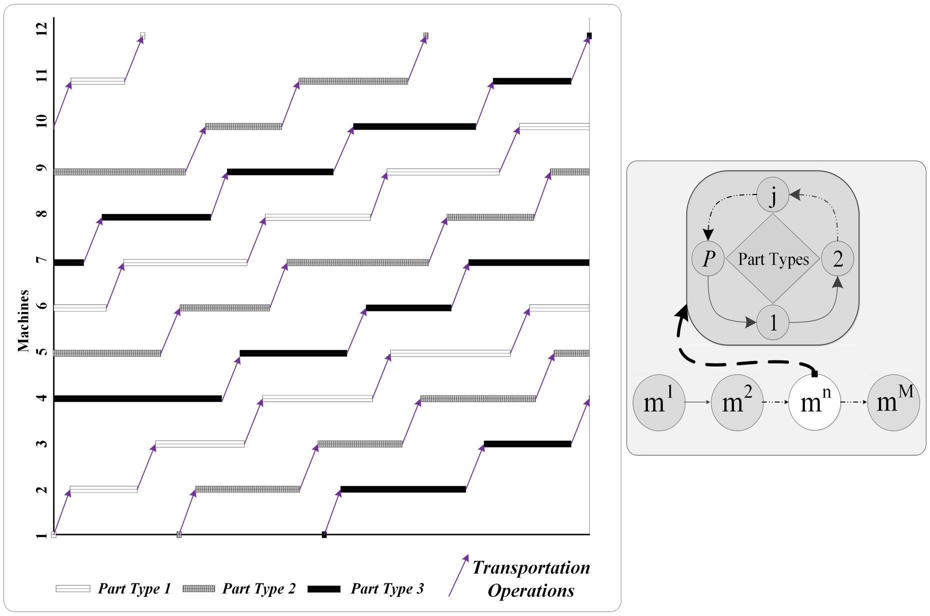

FIGURE 2: The example Gantt Chart for a cyclic solution with 3 part types; The cyclic change on the sequence of part types after any occupied machine at the start of the cycle.

each part has to be determined. The following sets of constraints are used to determine the sequence of parts and the immediate successor of each part.

$$X_{j,j''} \leq 2 - X_{j,j'} - X_{j',j''}, \qquad \forall \{j, j', j'' | j, j', j'' \in \{1, 2, \ldots, J\}; j \neq j' \neq j''\} \tag{1}$$

$$\sum_{j' \in J; j' \neq j} X_{j,j'} = 1, \qquad \forall j \in \{1, 2, \ldots, J\} \tag{2}$$

$$\sum_{j \in J; j \neq j'} X_{j,j'} = 1, \qquad \forall j' \in \{1, 2, \ldots, J\} \tag{3}$$

Constraint sets Eq. 1, Eq. 2, and Eq. 3 determine each part's preceding and subsequent part.

2.2. *Processing Time Constraints*

In flow line production systems, the processing of a part on a machine starts when it completes processing on the previous machine. Thereupon, if machine m is free at the beginning of the cycle, then the processing time constraints are as follows:

$$S_{[j,m-1]} + \theta_{m-1} + \mathrm{PLo}_{j,m} \leq S_{[j,m]} \tag{4}$$

$$S_{[j,m-1]} + \theta_{m-1} + \text{PUp}_{j,m} \geq S_{[j,m]} \tag{5}$$

where Eq. 4 and Eq. 5 state that part j requires at least $\text{PLo}_{j,m}$ time to be processed and can stay at most $\text{PUp}_{j,m}$ time ($\text{PLo}_{j,m} \leq \text{PUp}_{j,m}$) on machine m, respectively. On the other hand, if machine m is occupied at the beginning of the cycle, the start time of the related processing operation is determined with respect to the previous cycle as follows:

$$S_{[j,m-1]} + \theta_{m-1} + \text{PLo}_{j,m} \leq S_{[j,m]} + T \tag{6}$$

$$S_{[j,m-1]} + \theta_{m-1} + \text{PUp}_{j,m} \geq S_{[j,m]} + T. \tag{7}$$

Variables $A_{j,m}$ are used to determine whether the related machine m is occupied by part j at the start of the cycle. As each machine can only be occupied by just one part we have

$$\sum_{j \in J} A_{j,m} \leq 1, \qquad \forall\, m \in \{1, \ldots, M-1\}. \tag{8}$$

The processing time restrictions (Eq. 4 - 7) are linked to the variables $A_{j,m}$ as follows:
$\forall \{j,m | j \in \{1,2,\ldots,J\}, m \in \{2,\ldots,M-1\}\}$,

$$\begin{cases} S_{[j,m-1]} + \theta_{m-1} + \text{PLo}_{j,m} - \mathscr{M} \times (A_{jm}) \leq S_{[j,m]} \\ S_{[j,m-1]} + \theta_{m-1} + \text{PLo}_{j,m} - \mathscr{M} \times (1 - A_{jm}) \leq S_{[j,m]} + T, \end{cases} \tag{9}$$

$\forall \{j,m | j \in \{1,2,\ldots,J\}; m \in \{2,\ldots,M-1\}\}$,

$$\begin{cases} S_{[j,m-1]} + \theta_{m-1} + \text{PUp}_{j,m} + \mathscr{M} \times (A_{jm}) \geq S_{[j,m]} \\ S_{[j,m-1]} + \theta_{m-1} + \text{PUp}_{j,m} + \mathscr{M} \times (1 - A_{jm}) \geq S_{[j,m]} + T, \end{cases} \tag{10}$$

where constraint sets Eq. 9 and Eq. 10 guarantee the pickup criteria for lower and upper bounds, respectively.

2.3. Operational Sequence Constraints for Machines

The operational sequence constraints prevent the formation of conflicts between parts on the machines. Each machine should be unloaded before the next part to be processed is loaded on it. It should be also noted that since there is no buffer between machines, completed parts have to remain on the machines until both the next machine and any of the robots become available. Considering the sequence of parts, if j' is the successor of part j then

$$S_{[j,m]} \leq S_{[j',m-1]} + \theta_{m-1}. \tag{11}$$

Considering the cyclic nature of the problem, the operational sequence of the last (j) and first (j') parts on machine m can be stated as:

$$S_{[j,m]} \leq S_{[j',m-1]} + \theta_{m-1} + T \tag{12}$$

Constraint set Eq.13 guarantees that the operational sequence of all parts follow the Eq.11 if machine m is occupied at the start of the cycle. In addition, Eq. 14 states the operational sequence of each pair of part types (part j and its successor j') based on inequalities Eq.11 and Eq.12 considering the cyclic sequence of all part types on machines.

$$\sum_{j\in J} B_{j,m} + \sum_{j\in J} A_{j,m} = 1, \qquad \forall\, m \in \{2,\dots,M-1\} \tag{13}$$

$$\begin{cases} S_{[j,m]} - \mathscr{M} \times (1 + B_{j,m} - X_{j,j'}) \leq S_{[j',m-1]} + \theta_{m-1} \\ S_{[j,m]} - \mathscr{M} \times (2 - B_{j,m} - X_{j,j'}) \leq S_{[j',m-1]} + \theta_{m-1} + T \\ \forall \{j, j' \in \{1,2,\dots,J\}; m \in \{2,\dots,M-1\}\} \end{cases} \tag{14}$$

2.4. Robot Assignment and Transportation Operation Sequences

In this study, robotic cells with single or multiple robots are considered, and therefore in the case of multiple robots any one of the robots could be assigned to perform a transportation operation. The constraint set Eq. 15 is formulated to assign an appropriate robot for each transportation operation.

$$\sum_{r=1}^{R} Z_{[j,m],r} = 1 \qquad \forall \{j \in \{1,2,\dots,J\}; m \in \{1,\dots,M-1\}\} \tag{15}$$

Considering that there are a limited number of robots, and each robot can perform only one transportation operation at a time, the transportation operations assigned to a robot in a cycle should be sequenced to avoid conflicts. The following inequality (Eq.16) represents the operational sequence of the transportations assigned to a robot;

$$S_{[j,m]} + \theta_m + \delta_{[\![m+1,m']\!]} \leq S_{[j',m']}, \tag{16}$$

where both $[j,m]$ and $[j',m']$ transportation operations are performed by the same robot, and $[j,m]$ precedes $[j',m']$. In order to determine the operational sequence of all transportation operations in a cycle, constraint set Eq. 17 should be satisfied.

$$Y_{[j,m],[j'm']} + Y_{[j',m'],[j,m]} = 1 \quad \forall \left\{ j, j' \in \{1,2,\ldots,J\}; m,m' \in \{1,\ldots,M-1\}; j \neq j' \vee m \neq m' \right\} \tag{17}$$

Eq. 16 is extended to constraint sets Eq. 18 and Eq. 19 to take into account the sequence of transportation operations and corresponding robot assignments. Thereupon, these constraints ensure the completion of transportation operation $[j,m]$ and repositioning, if required, before the next transportation operation $[j',m']$. Moreover in Eq. 18, $m+1 \neq m'$ implies that machine $m+1$ is not the next machine to be unloaded after machine m in this set of constraints.

$$\begin{cases} S_{[j,m]} + \theta_m + \delta_{\llbracket m+1,m' \rrbracket} - \mathscr{M} \times (3 - Y_{[j,m],[j',m']} - Z_{[j,m],r} - Z_{[j',m'],r}) \leq S_{[j',m']} \\ \forall \{j, j' \in \{1,2,\ldots,J\}; m,m' \in \{1,\ldots,M-1\}; r \in \{1,2,\ldots,R\}; j \neq j' \wedge \; m+1 \neq m'\} \end{cases} \tag{18}$$

It is important to note that the presence of dual gripper options makes it possible to unload two consecutive machines m and $m+1$ without any empty move by the same robot. Therefore, this idea is captured in constraint set Eq. 19, which is dropped in Eq. 18 (by $m+1 \neq m'$) to cover a more general case.

$$\begin{cases} S_{[j,m]} + \theta_m + (2 - G_r) \times \theta_{m+1} - \mathscr{M} \times (3 - Y_{[j,m],[j',m+1]} - Z_{[j,m],r} - Z_{[j',m+1],r}) \leq S_{[j',m+1]} \\ \forall \{j, j' \in \{1,2,\ldots,J\}; m \in \{1,\ldots,M-1\}; r \in \{1,2,\ldots,R\}; j \neq j'\} \end{cases} \tag{19}$$

2.5. Robot Collision Avoidance Constraints

Robots can visit all machines in both robot centered and flow line systems. However, they operate in the the same area and collisions are possible. Therefore, scheduling multiple robots is significantly more difficult than scheduling a single robot due to the added complexity of collisions. In this research, robots are considered to share the same rail to move between machines. Thus, the following constraints (Eq. 20 and 21) are defined to avoid collisions between robots in multiple robotic cells.

$$\begin{cases} S_{[j,m]} + \theta_m + \delta_{\llbracket m+1,m' \rrbracket} - \mathscr{M} \times (3 - Y_{[j,m],[j'm']} - Z_{[j,m],r} - Z_{[j',m'],t}) < S_{[j',m']} \\ S_{[j',m']} + \theta_{m'} + \delta_{\llbracket m'+1,m \rrbracket} - \mathscr{M} \times (3 - Y_{[j',m'],[j,m]} - Z_{[j,m],r} - Z_{[j',m'],t}) < S_{[j,m]} \\ \forall \{j, j' \in \{1,2,\ldots,J\}; m,m' \in \{1,\ldots,M-1\}; r \in \{1,2,\ldots,R\}; j \neq j' \vee m > m'\} \end{cases} \tag{20}$$

$$\begin{cases} S_{[j,m]} + \theta_m + \delta_{\llbracket m+1,m' \rrbracket} - \mathscr{M} \times (3 - Y_{[j,m],[j'm']} - Z_{[j,m],r} - Z_{[j',m'],t}) < S_{[j',m']} \\ S_{[j',m']} + \theta_{m'} + \delta_{\llbracket m'+1,m \rrbracket} - \mathscr{M} \times (3 - Y_{[j',m'],[j,m]} - Z_{[j,m],r} - Z_{[j',m'],t}) < S_{[j,m]} \\ \forall \{j, j' \in \{1,2,\ldots,J\}; m,m' \in \{1,\ldots,M-1\}; r \in \{1,2,\ldots,R\}; j \neq j' \vee m < m'\} \end{cases} \tag{21}$$

2.6. *Proposed Mathematical Programming Model*

The proposed mathematical programming model can be defined as:

$$[MIP]\begin{cases}\text{minimize} & T\\ \text{subject to} & \\ & \text{Eq.s } 1-3, 8-10, 13-15, 17-21\\ & X, R, B, Z, Y \in \{0,1\}\\ & T, S \geq \mathbf{0}\end{cases}$$

3. Investigating the Efficacy of the Model

We investigate the performance of proposed MIP model in the scheduling of cyclic robots with single and dual grippers. In order to demonstrate the impact of the number of robots and gripper types, we use generate problem instances with one, two and three robots, along with ten processing machines and input and output devices. The free pickup criteria is considered for the instances as it is a comprehensive form of the no-wait and the time-window pickup criteria. The performance on the CFS-RCS problems are dependent on the number of robots and gripper types. A solution found by the model provides the sequence of parts, the robot assignments for the transportation operations, and the sequence of transportation operations assigned to each robot. Tables 1 and 2 provide the related data including the processing time of each part type on each machine and empty and loaded movement times of robots between the machines. In particular, the transportation time (θ) in Table 2 shows the loaded transportation times between two consecutive machines as the parts travel through the machines in increasing order.

TABLE 1: The processing time in seconds of different part types on machines (free pickup).

Parts	Machines including input and output devices											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	74	62	89	38	34	67	96	61	36	57	0
2	0	42	90	45	90	97	60	47	86	84	33	0
3	0	38	63	36	85	47	75	92	61	71	52	0

TABLE 2: The required time in seconds for the robots to perform transportation operations and empty moves between machines.

Data		Machines											
		1	2	3	4	5	6	7	8	9	10	11	12
Transportation time (θ)		18	20	20	20	21	19	20	18	22	19	20	0
Empty move (σ)	1	0	2	4	7	9	14	16	18	20	24	27	29
	2	2	0	2	5	7	12	14	16	18	22	25	27
	3	4	2	0	3	5	10	12	14	16	20	23	25
	4	7	5	3	0	2	7	9	11	13	17	20	22
	5	9	7	5	2	0	5	7	9	11	15	18	20
	6	14	12	10	7	5	0	2	4	6	10	13	15
	7	16	14	12	9	7	2	0	2	4	8	11	13
	8	18	16	14	11	9	4	2	0	2	6	9	11
	9	20	18	16	13	11	6	4	2	0	4	7	9
	10	24	22	20	17	15	10	8	6	4	0	3	5
	11	27	25	23	20	18	13	11	9	7	3	0	2
	12	29	27	25	22	20	15	13	11	9	5	2	0

3.1. Single Gripper Robots

In this section, we focus on single gripper robots. Figures 3-5 present the Gantt charts for solutions to the problem with one, two and three single gripper robots, respectively. Figure 3 illustrates the schedule found with a single gripper robot. The robot moves are represented with blue arrows. Additionally, solid and dashed arrows are used to show loaded and empty moves, respectively. The cycle time is $T = 868$ seconds and the sequence of the parts is $\{1,2,3\}$. In this example, machines $\{3,5,6,8,10,11\}$ are busy at the start of the cycle. As there is only one single gripper robot in this example, the collision constraints are not relevant. The second solution is presented in Figure 4 which illustrates the case with two single gripper robots. Their moves are represented by blue and red arrows with solid and dashed arrows representing loaded and void moves, respectively. The cycle time is $T = 590$ seconds and the sequence of parts is $\{1,2,3\}$. In this example, machines $\{4,5,6,7,9,10\}$ are busy at the start of the cycle. As there are two single gripper robots in this example, the collision constraints are potentially active and these constraints ensure that there are no collisions between the robots.[3] Figure 5 shows the solution for the instance with three single gripper

[3] The collision constraints ensure the robots do not pass each other.

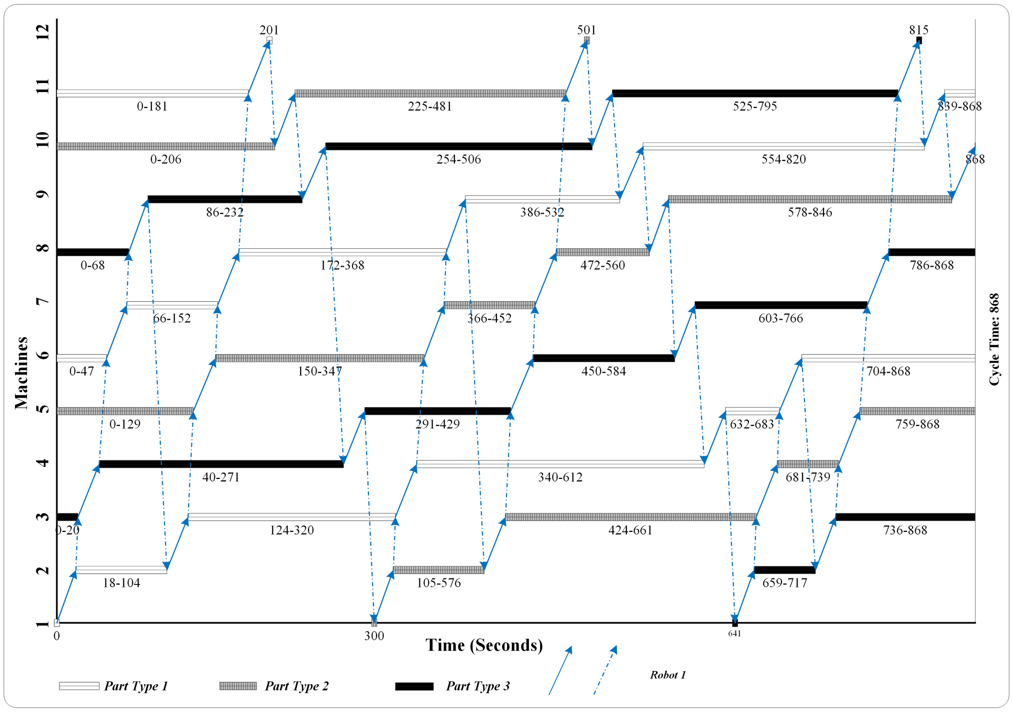

FIGURE 3: The Gantt chart of the solution found for the problem with a single gripper robot.

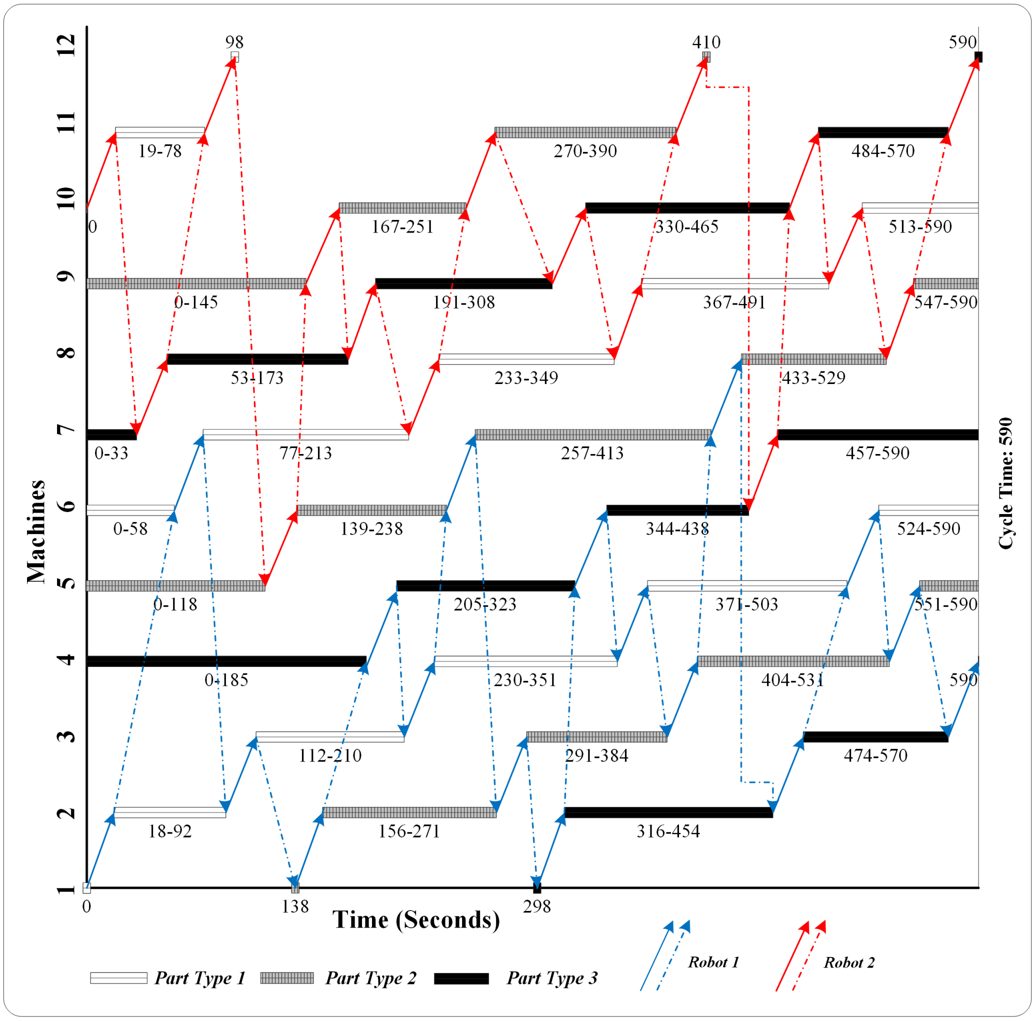

FIGURE 4: The Gantt chart of the solution found for the problem with two single gripper robots.

robots. Solid and dashed lines show loaded and empty moves of the robots, respectively, in different colors to distinguish the robots. The cycle time for this instance is $T = 517$ seconds and the sequence of the parts is $\{1,2,3\}$. In this example, machines $\{3,4,5,7,8,9\}$ are busy at the start of the cycle. As there are three single gripper robots in this example, the collisions are avoided due to the collision constraints. Figures 3-5 illustrate the impact of multiple single gripper robots in decreasing the cycle time. Starting with $T = 868$ seconds for one single gripper robot, the cycle length decreases to $T = 590$ and $T = 517$ by incorporating two and three single gripper robots, respectively. In addition, the figures clearly show that the robots do not collide in all three cases. It is important to note that since single gripper robots are used in these three instances, they are unable to unload and load different parts at the same time on a particular machine.

3.2. Dual Gripper Robots

In this section the problems with one, two and three dual gripper robots are considered. Figures 6-8 illustrate the solutions for these instances. As shown, the robots may reach different machines irrespective of their positions without any conflicts. Moreover, since the robots are dual gripper, they are able to unload a particular part from a machine and load it onto another machine simultaneously, if required. As shown in Figure 6, there is only one dual gripper robot, and its moves are shown with blue arrows. As with the previous charts, solid and dashed arrows are used for loaded and empty moves, respectively. The optimal cycle time is $T = 723$ seconds, and the sequence of the parts is $\{1,2,3\}$. In this example, there are eight occupied machines as $\{2,3,4,6,8,9,10,11\}$ at the start of the cycle. Since there is only one dual gripper robot in this example, the collision constraints are not relevant. It can be seen that dual grippers decrease the cycle time from $T = 868$ seconds in Figure 3 to $T = 723$ seconds in Figure 6. This example clearly illustrates the effect of dual grippers in decreasing the cycle time as a dual gripper robot is capable of unloading the previous part and loading next part simultaneously on machines, if required. There are two dual gripper robots in Figure 7, and their moves are shown with blue and red arrows. The cycle time is found to be $T = 561$ seconds with the sequence of the parts $\{1,2,3\}$. In this example, machines $\{3,4,7,8,9,11\}$ are busy at the start of the cycle. It can be seen that the dual gripper option decreases the cycle time from $T = 590$ seconds in Figure 4 to $T = 561$ seconds in this example. In addition, Figure 7 pictorially explains that robot 1 unloads part 2 from machine 3 and carries it to machine 4 with the first gripper. Then by changing the gripper, it unloads part 1 from machine 4 with the second gripper. Again by changing the gripper, it loads part 2 on machine 4 with the first gripper and then carries part 1 to machine 5 with the second gripper. Furthermore, the same operations take place for other pairs of transportation operations when the dual gripper option is used. The

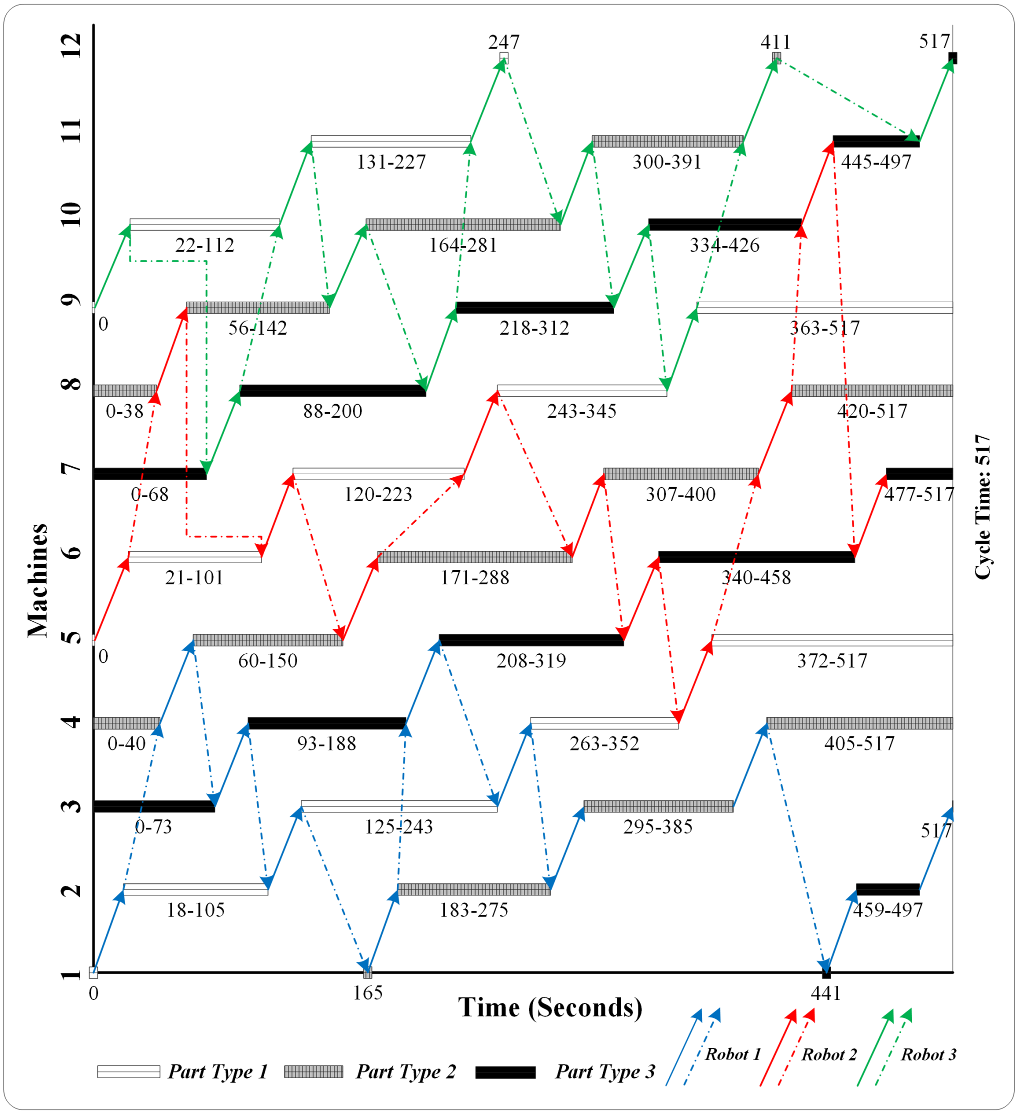

FIGURE 5: The Gantt chart of the solution found for the problem with three single gripper robots.

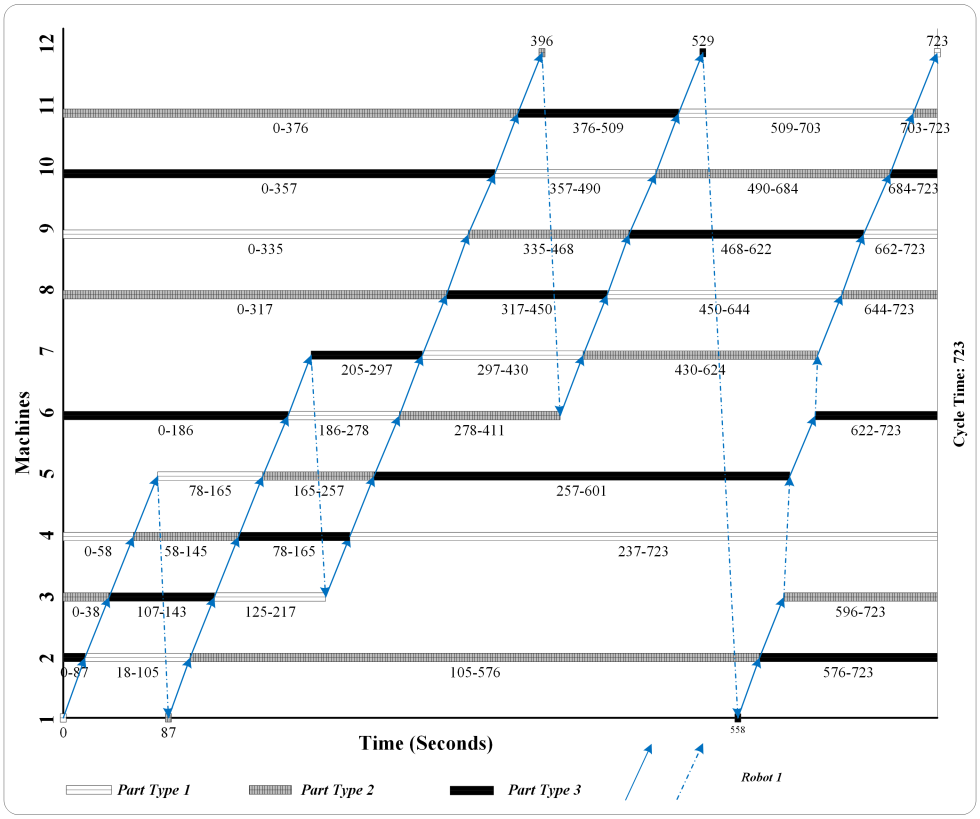

FIGURE 6: The Gantt chart of the solution found for the problem with a dual gripper robot.

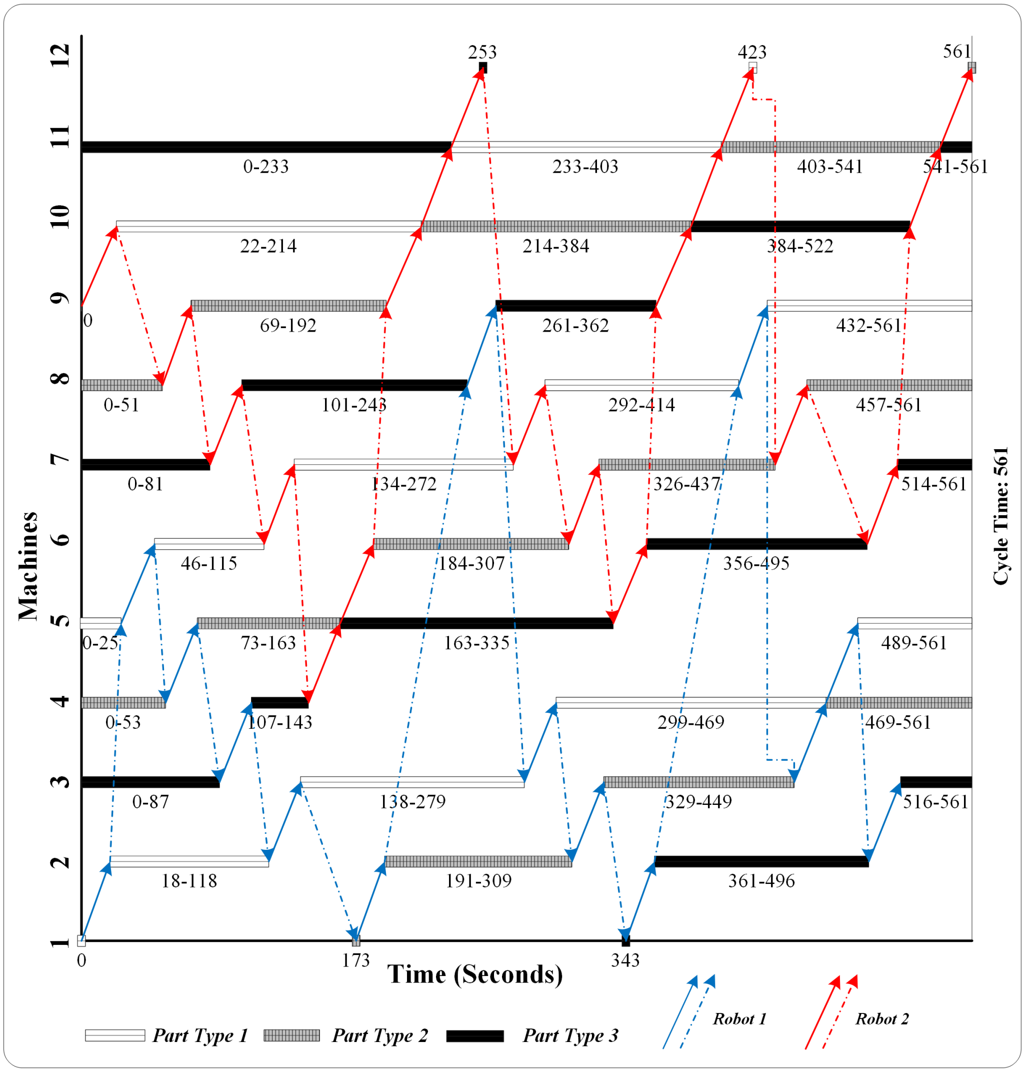

FIGURE 7: The Gantt chart of the solution found for the problem with two dual gripper robots.

pairs of transportation operations (TO) that where dual grippers are utilized are listed in Table 3. As shown in in this table, the second robot has used the dual gripper option more than the first one.

TABLE 3: The pairs of transportation operations (TO) that are done considering dual grippers.

Pairs	First TO	Second TO	Assigned Robot
1	[1,4]	[2,3]	1
2	[1,10]	[2,9]	2
3	[1,11]	[2,10]	2
4	[2,5]	[3,4]	2
5	[2,10]	[3,9]	2
6	[2,11]	[3,10]	2
7	[3,11]	[1,10]	2

The optimal solution in shown in Figure 8 consists of three dual gripper robots. Their moves are shown with red, blue and green arrows, respectively. The optimal cycle time is $T = 517$ seconds, and the sequence of the parts is $\{1,2,3\}$. In this example, machines $\{3,4,5,8,9,10\}$ are occupied at the start of the cycle. It is clear from the figure that the dual gripper option does not change the cycle time, and cycle time is the same as the instance with three single gripper robots in Figure 5.

3.3. Mixed Single and Dual Gripper Robots

In addition to the robots with the same gripper types some real-life manufacturers use mixed gripper types. Here, we consider problems with mixed gripper types for two robots. Due to the collision conditions and its related limitations, the problem has been solved twice by changing the position of robots with different gripper types. The impact of positioning the robots with different gripper types can be seen by comparing the cycle times. The first case is when the first robot is a single gripper and the second robot is a dual gripper (Figure 9) =. Furthermore, Figure 10 illustrates the second case where the first robot is dual gripper and the second robots is a single gripper. Comparing the cycle times, it is clear that the dual gripper option for the second robot is more effective than the first robot. Moreover, this result could be inferred from Table 3 as it is mentioned that the second robot uses dual gripper option more than the first one.

4. Numerical Results

This section investigates the scaling of the MIP model with problem size in terms of number of machines, part types, robots and gripper options with the free pick-up criteria. The test problems are created with up

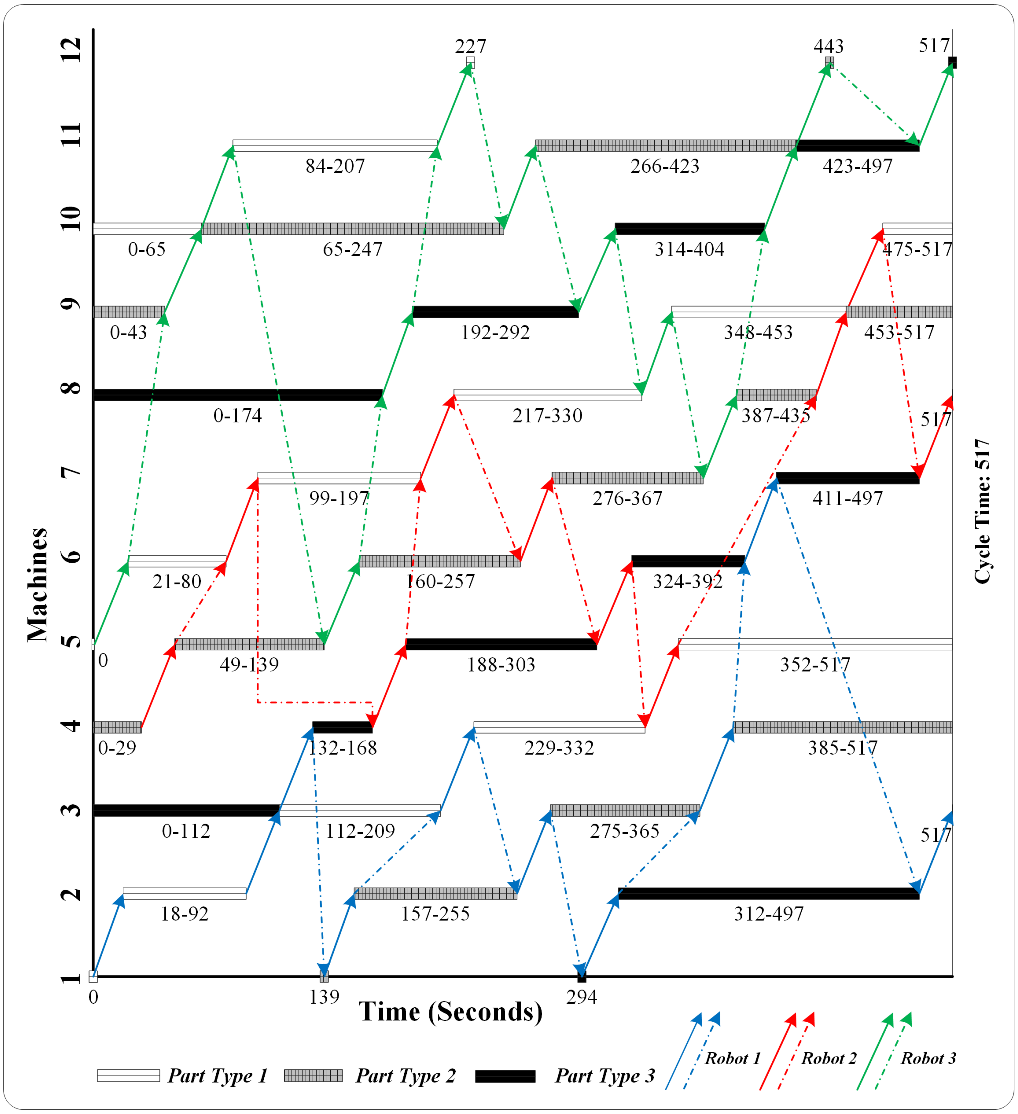

FIGURE 8: The Gantt chart of the solution found for the problem with three dual gripper robots.

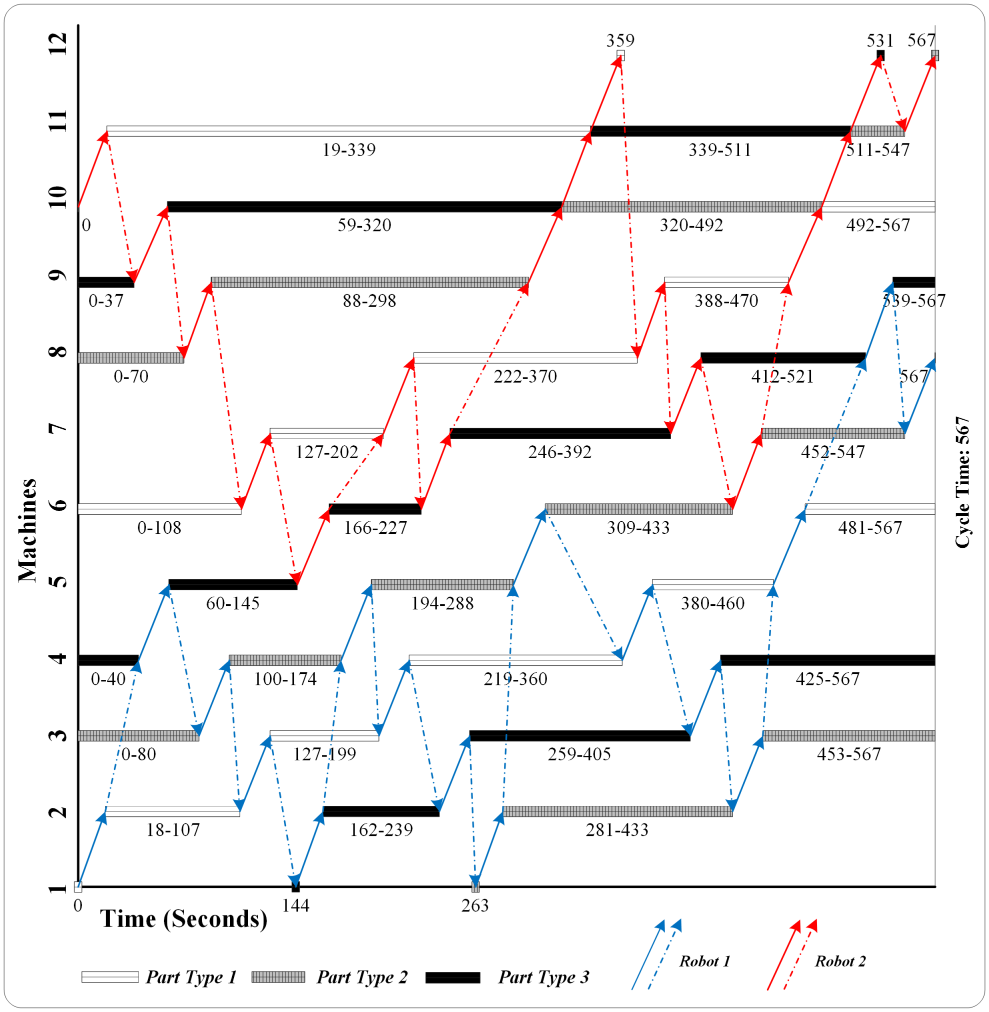

FIGURE 9: The Gantt chart of the solution found for the problem with two robots; the first and second robots are single and dual gripper, respectively.

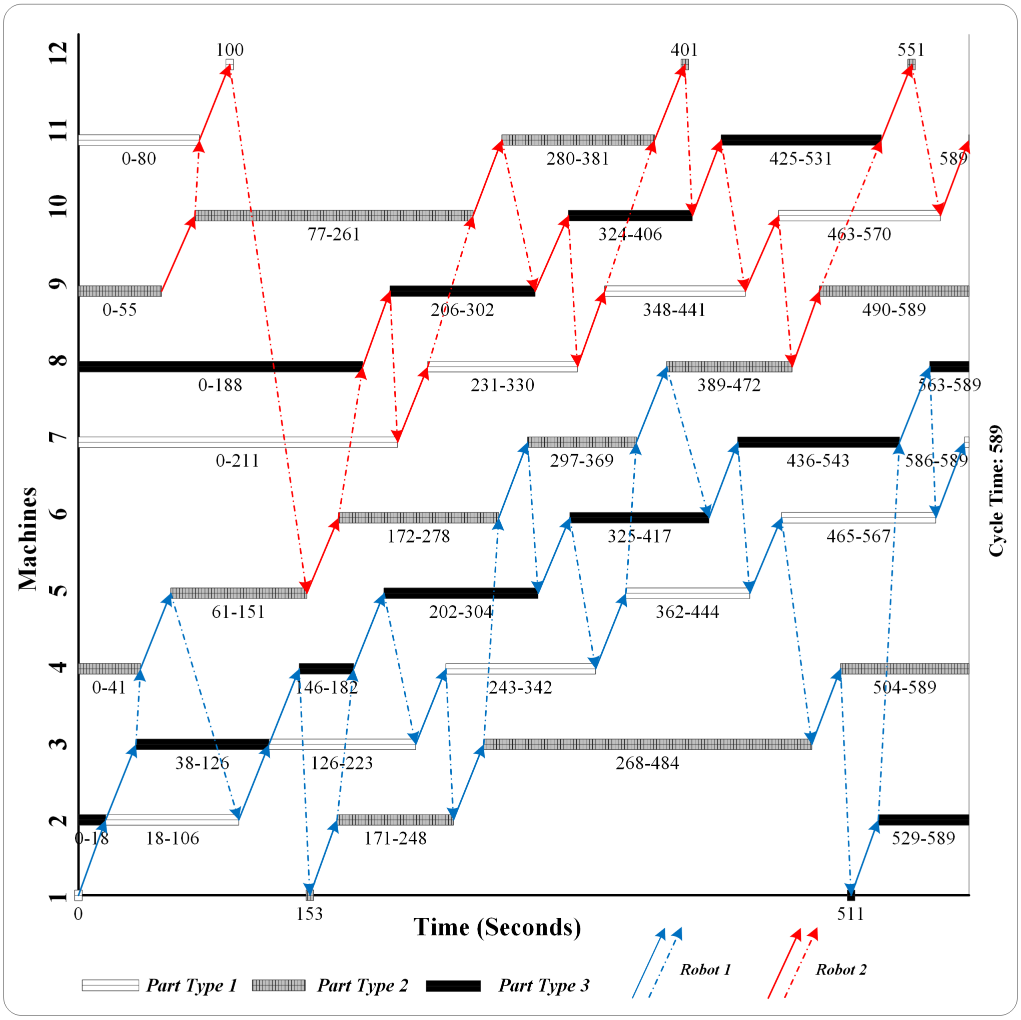

FIGURE 10: The Gantt chart of the solution found for the problem with two robots; the first and second robots are dual and single gripper, respectively.

to 20 machines, 20 parts, 3 robots and single or dual grippers (see Appendix Appendix A, which details the algorithm for generating the instances). The experiments were conducted on MONARCH, a cluster at Monash University [37]. Each run was given 4 cores and 10 GB memory. The MIPs were solved using Gurobi [38]. Each run was given 5 hours of wall clock time and since each run was also given 4 cores, the total CPU time available to each run is 72,000 seconds. The results are presented in the following tables. The grippers can be single (1) or dual (2). The Gap is a % gap from the cycle time to the lower bound found by the MIP. The times are computed as CPU cycles and reported in seconds. The cells with a dash emphasize that no solution was found within the given time period.

TABLE 4: Computational results for 5 machines obtained by the proposed model.

Machines	Parts	Robots	Gripper	Cycle Time	Gap(%)	Time(s)	#Constraints	#Variables	#Non-zeros
5	2	1	1	98	0	3.2	177	145	595
			2	67	0	0.07	177	145	595
		2	1	49	0	0.37	233	155	883
			2	41	0	0.1	233	155	883
		3	1	39	0	0.11	289	165	1171
			2	39	0	0.11	289	165	1171
	4	1	1	205	0	5.59	675	497	2389
			2	149	0	4.63	675	497	2389
		2	1	103	0	41.24	915	517	3605
			2	82	0	6.66	915	517	3605
		3	1	77	0	237.02	1155	537	4821
			2	73	0	1.28	1155	537	4821
	5	1	1	260	0	18.58	1062	751	3781
			2	190	0	28.62	1062	751	3781
		2	1	131	0	414.96	1442	776	5701
			2	96	0	38.39	1442	776	5701
		3	1	95	0	50015.27	1822	801	7621
			2	90	0	5.37	1822	801	7621
	6	1	1	315	0	237.51	1549	1057	5527
			2	231	0	868.1	1549	1057	5527
		2	1	155	0	22940.25	2101	1087	8311
			2	116	0	2620.54	2101	1087	8311
		3	1	110	7	71794.75	2653	1117	11095
			2	102	0	23.18	2653	1117	11095
	7	1	1	370	0	7382.62	2142	1415	7645
			2	272	0	23616.8	2142	1415	7645
		2	1	182	27	71189.98	2898	1450	11453
			2	135	0	24366.41	2898	1450	11453
		3	1	131	11	71311.73	3654	1485	15261
			2	117	0	64.18	3654	1485	15261

Table 4 presents the results for the instances with 5 machines. It is clear that solutions are found for all instances and most of them are optimal (Gap = 0). There are three instances where the optimal solution is not guaranteed and these are the ones which also require the largest run-times. Generally it can be seen that there are increased run-times with increasing problem size, however, the main contributor appears to be the number of robots with single gripper (e.g. Instances 23, 27, 29).

TABLE 5: Computational results for 10 machines obtained by the proposed model.

Machines	Parts	Robots	Gripper	Cycle Time	Gap(%)	Time(s)	#Constraints	#Variables	#Non-zeros
10	4	1	1	459	35	71828.84	2965	1777	10509
			2	321	22	71936.19	2965	1777	10509
		2	1	222	44	71906.88	4225	1817	16845
			2	156	22	71914.19	4225	1817	16845
		3	1	170	39	71843.66	5485	1857	23181
			2	119	15	71879.7	5485	1857	23181
	8	1	1	1000	74	71735.65	11897	6785	42393
			2	673	62	71879.72	11897	6785	42393
		2	1	647	76	71871.2	17009	6865	68025
			2	393	54	71939.68	17009	6865	68025
		3	1	418	59	71828.19	22121	6945	93657
			2	296	38	71934.6	22121	6945	93657
	10	1	1	-	-	71902.9	18727	10501	66711
			2	849	64	71939.7	18727	10501	66711
		2	1	825	78	71884.17	26737	10601	106851
			2	627	71	71925.2	26737	10601	106851
		3	1	584	73	71845.94	34747	10701	146991
			2	397	56	71924.96	34747	10701	146991
	12	1	1	-	-	71897.61	27197	15025	96805
			2	1067	67	71936.06	27197	15025	96805
		2	1	-	-	71887.5	38753	15145	154693
			2	743	75	71889.62	38753	15145	154693
		3	1	731	77	71825.45	50309	15265	212581
			2	511	64	71854.12	50309	15265	212581
	16	1	1	-	-	71843.11	49249	26497	174897
			2	1377	68	71910.12	49249	26497	174897
		2	1	-	-	71781	69841	26657	278001
			2	1143	83	71771.66	69841	26657	278001
		3	1	1050	81	71663.8	90433	26817	381105
			2	813	75	71691.31	90433	26817	381105

Table 5 presents the results for the instances with 10 machines. We see here that the problems are large enough such that the MIP requires the full quota of 72,000 seconds for every problem instance. Despite the run-time allowance, there are a number of instances for which no solution has been found (e.g. 55, 57). Furthermore, when solutions are found, the gaps are often large with the lowest gap being 15% for instance 36.

TABLE 6: Computational results for 15 and 20 machines obtained by the proposed model.

Machines	Parts	Robots	Gripper	Cycle Time	Gap(%)	Time(s)	#Constraints	#Variables	#Non-zeros
15	9	1	1	1779	77	71845.52	34650	18847	122959
			2	1189	67	71910.73	34650	18847	122959
		2	1	1231	87	71882.91	50400	18982	201835
			2	934	80	71854.55	50400	18982	201835
		3	1	862	79	71680.97	66150	19117	280711
			2	729	83	71789.76	66150	19117	280711
	12	1	1	2592	81	71683.09	61887	33265	219565
			2	-	-	71709.46	61887	33265	219565
		2	1	-	-	71741.45	89943	33445	360013
			2	1415	88	71532.13	89943	33445	360013
		3	1	-	-	71495.47	117999	33625	500461
			2	1043	86	71648.33	117999	33625	500461
	15	1	1	-	-	71222.12	97242	51751	344791
			2	-	-	70287.82	97242	51751	344791
		2	1	-	-	71643.77	141132	51976	564451
			2	2922	95	71376.24	141132	51976	564451
		3	1	-	-	71015.79	185022	52201	784111
			2	1317	89	71269.13	185022	52201	784111
20	10	1	1	-	-	71762.02	76947	40901	272311
			2	-	-	71647.79	76947	40901	272311
		2	1	-	-	71611.2	112857	41101	452051
			2	1698	92	71432.34	112857	41101	452051
		3	1	-	-	71018.97	148767	41301	631791
			2	1357	90	71200.46	159540	41361	685713

Table 6 presents the results for 15 and 20 machines. Here the run-times are consistently large and also that there are several instances with no solution. In fact, no feasible solutions were found with more than 15 and 10 parts for instances with 15 and 20 machines, respectively. Figure 11 presents the affects of number of machines, parts and robots on the cycle time. It is partitioned into 4 columns (yellow lines) based on the number of machines. We see that multiple robots are more efficient when the number of machines increases in the manufacturing cell. Moreover, comparing the rate of the reduction on cycle time between different number of robots shows that the rate of reduction between one and two robots is noticeably higher than the reduction rate between two and three robots. This means that the the production rate can be significantly increased by considering two robots compared to a single robot. Additionally, Figure 12 demonstrates the affects of number of robots and gripper options on the cycle time based on the number of machines. For problems with the same number of machines and parts the cycle times of the systems with 1 dual gripper robot and 2 single gripper robots are similar (also between the 2 dual gripper robots and 3 single gripper robots). This helps provide options in utilizing robots in terms of number and gripper options. Figure 12 states that the efficiency of utilizing dual grippers increases with the number of machines and parts.

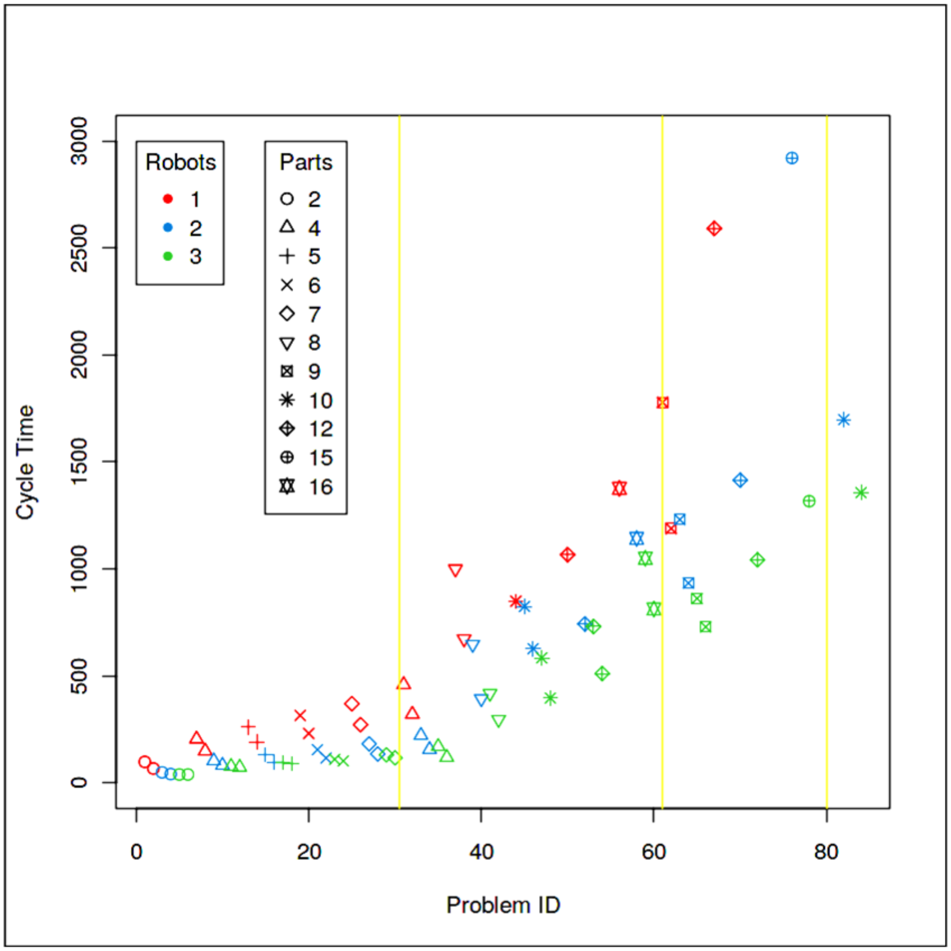

FIGURE 11: The affects of number of machines, parts and robots on the cycle time.

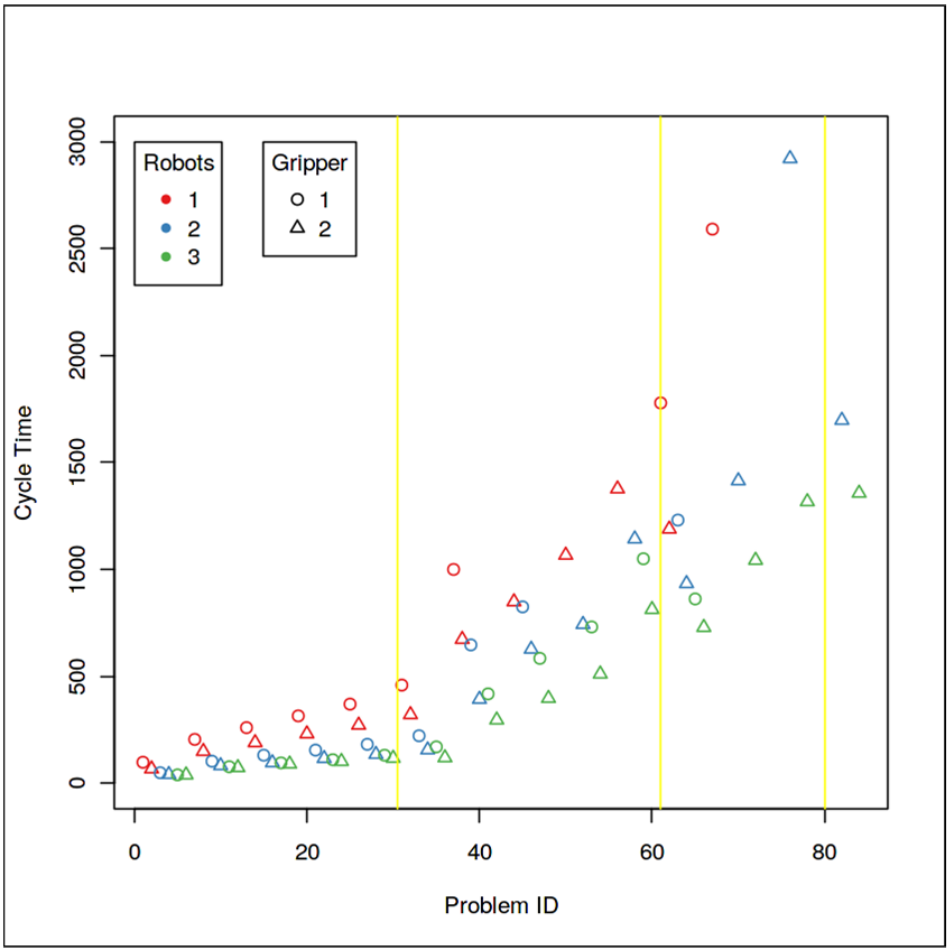

FIGURE 12: The affects of number of robots and gripper options on the cycle time.

5. Conclusion

To authors' knowledge, this is the first study considering the two topics in robotic cell scheduling; (a) multiple part type cyclic flow shop robotic cell scheduling with multiple robots, and (b) multiple dual gripper robots in cyclic flow shop robotic cells. A novel and comprehensive mixed-integer linear programming model is proposed for the multiple part type cyclic flow shop robotic cells scheduling problem considering multiple single and dual gripper robots and their collisions to minimise the cycle time and hence to maximise the throughput rate. In addition, the special characteristics of problem are analysed by generating instances consisting of different types and number of robots. The computational results confirm the efficiency of different types and number of robots on cycle times and on the throughput rate. It is also shown that the proposed model encompasses most of the real-life conditions that are encountered in robotic cells. The advantages of this study and the achievements from the computational study can be outlined as follows.

- The proposed model effectively solves the multiple part type cyclic flow shop robotic cell scheduling problem with different pickup criteria, namely, free, time-window and no-wait.
- The proposed model considers the multiple robots and their collisions in cyclic flow shop robotic cells (the first time this has been addressed to the authors' knowledge).
- The dual gripper is considered as an option for the robots in cyclic flow shop robotic cells and the proposed model addressed this option, also for the first time.
- This study proposes a novel MIP model for the multiple part type sequencing in cyclic flow shop systems.
- Differing number of robots and gripper options affect the optimal cycle time and throughput rate.
- Using the MIP model suggested here, different industrial systems can base their decisions on where to invest on the appropriate number of robots with gripper options.
- Due to the collision constraints, the affect of dual gripper option on the cycle time differs between the robots.

For future work, different aspects of the problem can be considered an extended. For example, the flow shop problem could be considered with an equivalent job shop problem. Mixed integer programming decompositions, such as Lagrangian relaxation or Column generation could prove to be effective in solving solving some of the larger instances in reduced time-frames. Furthermore, due to the large run-time requirements, an exact or heuristic approach could prove to be effective in solving the instances in a shorter time-frames. Additionally, parallel implementations via multi-core shared memory architectures or the message passing interface could also prove to be effective.

Appendix A. Generating Problem Instances

The problem instances were generated with the method in Algorithm 1. Here, the all the data associated with the machines and parts (*nMach* and *nPart*, respectively) are generated. ω serves as an input to the algorithm which allows varying the characteristics of the instances that are generated. First a lower limit (*lValue*) and an upper limit (*uValue*) are defined based on ω. The upper limit is generated uniformly in the interval $[5 \times \omega,\ (nMach + 2) \times \omega]$. Following this, the time-windows for processing the parts on the machines are defined (lines 4-7). This is followed by defining the void moves (lines 8-14) and the loaded moves (lines 15-18).

Algorithm 1 Generate Problems

```
1:  procedure FUNCTION($nMach, nPart, \omega$)
2:      $lValue \leftarrow 2 \times \omega$
3:      $uValue \leftarrow \mathscr{U}[5 \times \omega,\ (nMach + 2) \times \omega]$
4:      for all $j, m$: $(1 \leq m \leq nMach$ and $1 \leq j \leq nPart)$ do
5:          $\mathrm{PLo}_{j,m} \leftarrow \mathscr{U}[lVlaue, uValue]$
6:          $\mathrm{PUp}_{j,m} \leftarrow \mathrm{PLo}_{j,m} + \mathscr{U}[lVlaue, uValue]$
7:      end for                       ▷ generating the time-window processing times
8:      for all $m$: $(1 \leq m < nMach)$ do
9:          $\delta_{m,m+1} \leftarrow \omega + \mathscr{U}[-1, 1]$
10:     end for
11:     for all $m, m'$: $(1 \leq m$ and $m + 1 < m' \leq nMach)$ do
12:         $\delta_{m,m'} \leftarrow \Sigma_{n=m}^{m'-1} \delta_{n,n+1}$
13:         $\delta_{m',m} \leftarrow \delta_{m,m'}$
14:     end for                 ▷ generating the void move times of a robot between machines
15:     for all $m$: $(1 \leq m < nMach)$ do
16:         $\theta_{m,m+1} \leftarrow \omega + \mathscr{U}[-2, 1]$
17:     end for               ▷ generating the loaded move times of a robot between machines
18:     return $(PLo,\ PUp,\ \delta,\ \theta)$
19: end procedure
```

References

[1] K. L. Hitz, Scheduling of flexible flowshops.

[2] M. W. Dawande, H. N. Geismar, S. P. Sethi, C. Sriskandarajah, Throughput optimization in robotic cells, Vol. 101, Springer Science & Business Media, 2007.

[3] S. P. Sethi, C. Sriskandarajah, G. Sorger, J. Blazewicz, W. Kubiak, Sequencing of parts and robot moves in a robotic cell, International Journal of Flexible Manufacturing Systems 4 (3) (1992) 331–358.

[4] R. Logendran, C. Sriskandarajah, Sequencing of robot activities and parts in two-machine robotic cells, International Journal of Production Research 34 (12) (1996) 3447–3463.

[5] Y. Crama, J. Van De Klundert, Cyclic scheduling of identical parts in a robotic cell, Operations Research 45 (6) (1997) 952–965.

[6] M. Dawande, C. Sriskandarajah, S. Sethi, On throughput maximization in constant travel-time robotic cells, Manufacturing & Service Operations Management 4 (4) (2002) 296–312.

[7] A. Agnetis, Scheduling no-wait robotic cells with two and three machines, European Journal of Operational Research 123 (2) (2000) 303–314.

[8] E. Levner, V. Kats, V. E. Levit, An improved algorithm for cyclic flowshop scheduling in a robotic cell, European Journal of Operational Research 97 (3) (1997) 500–508.

[9] J. Liu, Y. Jiang, An efficient optimal solution to the two-hoist no-wait cyclic scheduling problem, Operations Research 53 (2) (2005) 313–327.

[10] V. Kats, E. Levner, L. Meyzin, Multiple-part cyclic hoist scheduling using a sieve method, IEEE Transactions on Robotics and Automation 15 (4) (1999) 704–713.

[11] A. Che, C. Chu, F. Chu, Multicyclic hoist scheduling with constant processing times, IEEE Transactions on Robotics and Automation 18 (1) (2002) 69–80.

[12] A. Che, C. Chu, Multi-degree cyclic scheduling of two robots in a no-wait flowshop, IEEE Transactions on Automation Science and Engineering 2 (2) (2005) 173–183.

[13] A. Che, C. Chu, Multi-degree cyclic scheduling of a no-wait robotic cell with multiple robots, European Journal of Operational Research 199 (1) (2009) 77–88.

[14] A. Che, M. Chabrol, M. Gourgand, Y. Wang, Scheduling multiple robots in a no-wait re-entrant robotic flowshop, International Journal of Production Economics 135 (1) (2012) 199–208.

[15] A. Karzanov, E. Livshits, Minimal quantity of operators for serving a homogeneous linear technological process, Automation and Remote Control 39 (3) (1978) 445–450.

[16] Z. Zhou, A. Che, P. Yan, A mixed integer programming approach for multi-cyclic robotic flowshop scheduling with time window constraints, Applied Mathematical Modelling 36 (8) (2012) 3621–3629.

[17] N. Brauner, G. Finke, W. Kubiak, Complexity of one-cycle robotic flow-shops, Journal of Scheduling 6 (4) (2003) 355–372.

[18] K. Hasani, S. A. Kravchenko, F. Werner, Minimizing total weighted completion time approximately for the parallel machine problem with a single server, Information Processing Letters 114 (9) (2014) 500–503.

[19] A. Che, Z. Zhou, C. Chu, H. Chen, Multi-degree cyclic hoist scheduling with time window constraints, International journal of production research 49 (19) (2011) 5679–5693.

[20] V. Kats, L. Lei, E. Levner, Minimizing the cycle time of multiple-product processing networks with a fixed operation sequence, setups, and time-window constraints, European Journal of Operational Research 187 (3) (2008) 1196–1211.

[21] V. Kats, E. Levner, A faster algorithm for 2-cyclic robotic scheduling with a fixed robot route and interval processing times, European Journal of Operational Research 209 (1) (2011) 51–56.

[22] A. El Amraoui, M.-A. Manier, A. El Moudni, M. Benrejeb, A linear optimization approach to the heterogeneous r-cyclic hoist scheduling problem, Computers & Industrial Engineering 65 (3) (2013) 360–369.

[23] V. B. Kats, An exact optimal cyclic scheduling algorithm for multi-operator service of a production line, Automation and Remote Control 43 (4) (1982) 538–542.

[24] K. Kogan, E. Levner, A polynomial algorithm for scheduling small-scale manufacturing cells served by multiple robots, Computers & operations research 25 (1) (1998) 53–62.

[25] A. Elmi, S. Topaloglu, A scheduling problem in blocking hybrid flow shop robotic cells with multiple robots, Computers & Operations Research 40 (10) (2013) 2543–2555.

[26] A. Elmi, S. Topaloglu, Scheduling multiple parts in hybrid flow shop robotic cells served by a single robot, International Journal of Computer Integrated Manufacturing 27 (12) (2014) 1144–1159.

[27] L. Lei, Q. Liu, Optimal cyclic scheduling of a robotic processing line with two-product and time-window constraints, INFOR: Information Systems and Operational Research 39 (2) (2001) 185–199.

[28] M. Mateo, R. Companys, Hoist scheduling in a chemical line to produce batches with identical sizes of different products, in: Sixième Conférence Francophone de Modélisation et SIMulation, MOSIM, Vol. 6, 2006, pp. 677–684.

[29] A. El Amraoui, M.-A. Manier, A. El Moudni, M. Benrejeb, A mixed linear program for a multi-part cyclic hoist scheduling problem, International Journal of Sciences and Techniques of Automatic control and computer engineering (IJ-STA), Special issue on CEM 2 (2008) 612–623.

[30] A. El Amraoui, M.-A. Manier, A. El Moudni, M. Benrejeb, Resolution of the two-part cyclic hoist scheduling problem with bounded processing times in complex linesß configuration, European Journal of Industrial Engineering 6 (4) (2012) 454–473.

[31] C. Varnier, N. Jeunehomme, A cyclic approach for the multi-product hoist scheduling problem, in: 7th International Workshop on Project Management and Scheduling-PMS (Osnabrueck, Germany, 2000), EURO, Ed, 2000.

[32] K. S. Hindi, K. Fleszar, A constraint propagation heuristic for the single-hoist, multiple-products scheduling problem, Computers & Industrial Engineering 47 (1) (2004) 91–101.

[33] H. J. Paul, C. Bierwirth, H. Kopfer, A heuristic scheduling procedure for multi-item hoist production lines, International Journal of Production Economics 105 (1) (2007) 54–69.

[34] N. G. Hall, Operations research techniques for robotics systems, Handbook of Industrial Robotics, Second Edition (2007) 543–577.

[35] Y. Crama, V. Kats, J. Van de Klundert, E. Levner, Cyclic scheduling in robotic flowshops, Annals of operations Research 96 (1-4) (2000) 97–124.

[36] M. Dawande, H. N. Geismar, S. P. Sethi, C. Sriskandarajah, Sequencing and scheduling in robotic cells: Recent developments, Journal of Scheduling 8 (5) (2005) 387–426.

[37] esolutions service desk, 2017. the monash campus cluster - confluence, monash university. eds: Philip chan and simon michnowicz.

[38] Gurobi optimization, inc., 2015. gurobi optimizer reference manual, url = "http://www.gurobi.com".