Ain Shams Engineering Journal (2013) 4, 465–474

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com

MECHANICAL ENGINEERING

Scheduling and sequencing in four machines robotic cell: Application of genetic algorithm and enumeration techniques

M.M.S. Abdulkader *, M.M. ElBeheiry, N.H. Afia, A.K. El-Kharbotly

Dept. of Design & Production Engineering, Ain Shams University, Faculty of Engineering, Cairo, Egypt

Received 23 May 2012; revised 15 November 2012; accepted 19 November 2012
Available online 26 January 2013

KEYWORDS

Scheduling;
Sequencing;
Robotic cells;
Cellular manufacturing;
Genetic algorithm

Abstract The introduction of robotic cells to manufacturing systems improved the efficiency, productivity and reliability of the system. The main objective of the scheduling problem of multi-item multi-machine robotic cells is the identification of the optimum robot cycle/s and jobs sequencing for certain processing conditions which yield the higher possible production rate. The objective of this work is to solve the scheduling problem in four-machine blocking robotic cells producing identical and different part types while minimizing the cycle time. A genetic algorithm is developed to find the parts sequence that minimizes the robot-moves cycle time for each robot cycle. The results showed that the developed genetic algorithm yields competitive results compared to the results of the full enumeration of all possible parts sequences. The results show also that the ratio between the average processing time of all parts and the robot travel time determines the cycle having the optimal robot-moves.

1. Introduction

In robotic cells, the robot performs a sequence of actions during which k parts are produced. This sequence of robot actions is defined by Sethi et al. [1] as *k-unit* cycle. A special case of the *k-unit* cycle is the one unit cycle in which only one part enters the cell and only one part leaves it while each machine is loaded and unloaded only once. He showed that the number of all possible robot-moves cycles for one unit cycle robotic cell is "$m!$" where m is the number of machines in the cell. For cells producing identical parts from two up to "m" machines were studied for scheduling of robot moves that minimize the robot cycle time. In case of cells producing different part types, the problem was solved for m machines and sequencing of parts such as to minimize the makespan [2,3].

Sethi et al. [1] proved that for two machines cell, the optimal one unit cycle is superior to any k-unit cycle and they conjectured this to be true for "$m \geqslant 3$". Hall et al. [4] proved that for three machines cells producing identical parts, the

* Corresponding author. Tel.: +20 1001807798.
E-mail address: Mohamed.abdulkader@eng.asu.edu.eg (M.M.S. Abdulkader).

Peer review under responsibility of Ain Shams University.

2090-4479 © 2012 Ain Shams University. Production and hosting by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asej.2012.11.002

optimum one unit cycle dominates all two unit cycles. Brauner and Finke [5] confirmed that Sethi et al. [1] conjecture is valid for two and three machines cell. For four machines the one unit cycle dominates the two unit cycle in cells with equidistance machines, and the three-unit cycles dominate all one unit cycles [5].

Most of the research work in the field of scheduling of robotic cells focused on the study of cells with identical parts [6–8]. Other researches considered different part types and few considered Minimal Part Set (MPS) (which is the fixed ratio between the numbers of produced parts every cycle) [2,3,5,9,10]. In case of scheduling different parts in a robotic cell, a focus is made on special category of robot move sequences which is called Concatenated Robot Move sequences (CRM). The CRM is achieved by repeating one unit cycle '*n*' times to produce '*k*' parts [11]. Most studies fix the robot move sequences to be a CRM sequence and then find the MPS part schedule that minimizes the total cycle time. The best part schedules for each CRM sequence are then compared [12].

The robot-move cycles can be classified according to the difficulty of cycle time calculations. The first class includes cycles which are trivially solvable in which their cycle time can be calculated by derived equations. The second class includes other cycles which are either solvable in polynomial time or non-polynomial time. The third class concerns cycles which depend on job sequences, where cycle times cannot be calculated using equations or exact formulation. These cycles are considered to be NP-hard and require heuristics for cycle time calculations as stated by Kamoun et al. [10].

The methods used to solve the robotic cell problem varied between heuristics, CPM–PERT, linear programming, integer programming, dynamic programming, branch and bound, deriving upper and lower bounds while most researchers solved the problem by formulating it as a Traveling Salesman Problem (TSP).

Genetic Algorithm (GA) was used to overcome the complexity of solving large scale problems and NP hard cycles [2,3]. Carlier et al. [2] investigated the problem of "*m*" machines robotic cell with a blocking constraint. They proposed a number of heuristics to minimize the makespan namely; the approximate decomposition algorithm, the exact branch-and-bound algorithm and the genetic algorithm. They reported that the proposed optimization-based approaches deliver high-quality solutions. The genetic algorithm delivers reasonably good solutions while requiring significantly shorter CPU times. Genetic algorithm was also used by Soukhal and Martineau [3] for solving large scale problems minimizing the makespan for *m* machines cells and gave good results.

NP hard cycles were studied by few researchers due to its complexity. NP-hard cycles may result in minimum cycle times when compared to other cycles. Sadek et al. [9] used mathematical simulation to solve the NP hard cycles. In the three machines robotic cells there were two NP hard cycles out of all the six cycles. The two NP hard cycles were proved to be optimal.

Limited work tackled the scheduling problem in four machine cells [9,13]. The problem of scheduling robot moves and sequencing parts simultaneously in cells that produce different part types and contain more than three machines was tackled by Kamoun et al. [10]. They solved the MPS sequencing under different robot moves cycles in three machines robotic cells. They also gave a methodology for extending this heuristic to four machines cells. Due to the difficulty of finding the optimal solutions to the problem of four machines cells; they found a lower bound and tested the heuristic against it.

Kamoun et al. [10]. concluded that when the robot is relatively slow, the cycle in which the loading of machines is done in regular sequence $(1,2,\ldots,m)$ is probably the optimal cycle since it has the minimum travel time. While the cycle in which the loading is done in reversed sequence $(m,m-1,\ldots,1)$ might be the optimal cycle when the robot is relatively fast because it allows each machine as much time for processing as possible. It was recommended that cycle #1 is the optimum robot cycle and superior to other cycles for low ratios between processing time and robot travel time while cycle #24 become the optimum cycle for higher ratios between processing times and robot travel times. The in-between cycles must be explored as they may lead to better solutions than cycles #1 and 24.

The objective of this work is to solve the scheduling problem in four-machine blocking robotic cells producing identical and different parts while minimizing the cycle time. The problem includes scheduling of robot moves and sequencing of parts simultaneously. A genetic algorithm is developed to solve the problem. The solution is based on finding a parts' sequence for each robot move cycle. The sequence with minimum cycle time is considered the best reached one. The performance of the genetic algorithm is measured by comparing the results with the results of full enumeration of all possible parts sequences. The cycle times of the enumerated sequences is calculated for the 24 robot moves cycles. Mathematical simulation is used to track the robot actions and used in calculating the cycle time.

The rest of this paper is organized as follows: Section 2 presents the problem assumptions and configuration. In Section 3, the algorithm is tested against the full enumeration technique. Finally, Section 4 concludes the main results.

2. Problem description

The problem tackled in this paper involves the scheduling of robot moves and sequencing of different parts forming MPS in blocking flow shop robotic cell. The cell consists of four machines, input and output buffers and a robot with single gripper.

The robot is subjected to two types of waits. The first type of wait takes place when the robot loads certain part on one of the machines and waits till the machine completes the processing of that part, and then unloads the part and advance to the next machine. In this case the waiting time of the robot is equal to the processing time of the part. The second type of wait takes place after the robot loads the part as it moves onto perform other activities while the part is being processed. After finishing the other activities, the robot returns to unload the part and wait in front of the machine while the part is still under processing. In the latter case, the waiting time is equal to the processing time of the part after subtracting the time of all activities that were made by the robot while the part is in processing. If the time of these activities is higher than the processing time of the part, the robot will be able to unload the part as soon as it reaches the machine, and hence the wait time is zero.

2.1. Problem assumptions

The following assumptions are considered within the context of this work.

(1) A single robot serving a center-robotic cell with no storage buffers.
(2) The robot and the machines can handle only one part at a time.
(3) All parts are available in the input buffer at the beginning of the cycle (static cell).
(4) The cell is flow shop with no skipping allowed.
(5) The cell produces multiple part types according to certain MPS. In order to process n jobs, CRM sequences are considered in which the same robot-move cycle is repeated n times.
(6) The MPS is produced repetitively at regular intervals.
(7) The travel time of the robot is independent of the part being processed and assumed constant between any two successive machines.
(8) The travel time between any two non-successive machines equals the travel time multiplied by the number of travel distances between the machines in succession.
(9) The processing, loading, and unloading times are deterministic and constant for all parts and they differ according to the jobs and processing machines.

The following notations are considered:

P_i	Part i, $i = 1{:}n$
$S_{m,4}$	Robot-moves cycle m in cell contains four machines
I	Input buffer
O	Output buffer
M_M	Machine M
M_M^+	Unload machine M
M_M^-	Unload machine $M - 1$, move to machine M, load machine M
δ	The travel time between two successive machines
$t_{M,i}$	The processing time of part i on machine M
$L_{M,i}$	The loading time of part i on machine M
$U_{M,i}$	The unloading time of part i from machine M
$W_{M,i}$	The waiting time for part i to be finished on machine M

Both input and output buffers are denoted as machine 0.

2.2. Robot moves cycles and cycle time calculations

2.2.1. Derivation of robot moves cycles

In four-machine cell, there are 4! = 24 robot cycles as shown in Table 1.

For example, the actions of cycle $S_{24,4}$ and their corresponding times notations are listed in Table 2. Fig. 1 shows a schematic representation for robot movements in 4 M/C cell for the same cycle. The following are the descriptions of M/C notations given in Table 2.

- M_4^-: The cycle starts with the robot moving to $M3$ and waits at machine $M3$ till the processing of part P_i is completed. After part P_i is finished on $M3$; the robot unloads it, moves to machine $M4$ and loads the part.
- M_3^-: The robot moves to machine $M2$, and waits till the processing of P_{i+1} part is completed. After part P_{i+1} is finished on $M2$; the robot unloads it, moves to machine $M3$ and loads the part.
- M_2^-: The robot moves to machine $M1$, and waits in front of the machine till part P_{i+2} is completed. After part P_{i+2} is finished the robot unloads it, moves to machine $M2$ and loads the part.
- M_1^- : The robot moves to the input buffer, picks up part P_{i+3} from the input buffer, moves to machine $M1$ and loads the part.
- M_4^+: The robot moves to machine $M4$, and waits in front of the machine till P_i part is completed. After part P_i is finished on $M4$; the robot unloads the part, moves to output buffer and drops the part.

2.2.2. Cycle time calculations

The cycle time CT depends on the robot actions. It is equal to the time taken by the robot to complete one cycle where the robot returns back to its initial position at the beginning of the cycle. Therefore, it is the summation of robot activities time. The CT of cycle $S_{24,4}$ is calculated by Eq. (1). Eqs. (2)–(5) are used to calculate the waiting times. Each waiting time for the robot is the maximum of "zero" and the processing time minus the time of the actions made by the robot after it loads the machine until it returns to unload it again as illustrated in Table 2.

$$T_{24} = \sum_{i=1}^{n}\sum_{M=0}^{4} L_{M,i} + \sum_{i=1}^{n}\sum_{M=0}^{4} U_{M,i} + \sum_{i=1}^{n}(W_{1,i+2} + W_{2,i+1} + W_{3,i} + W_{4,i}) + 16n\delta \tag{1}$$

where

$$W_{1,i+2} = \max\left(0, t_{1,i+2} - \begin{pmatrix} 12\delta + W_{4,i-1} + U_{4,i-1} + L_{0,i-1} + W_{3,i} \\ +U_{3,i} + L_{4,i} + W_{2,i+1} + U_{2,i+1} + L_{3,i+1} \end{pmatrix}\right) \tag{2}$$

Table 1 The cycles available in four machines robotic cell.

$S_{1,4}$	$(M_1^-, M_2^-, M_3^-, M_4^-, M_4^+)$	$S_{13,4}$	$(M_3^-, M_1^-, M_2^-, M_4^-, M_4^+)$
[a]$S_{2,4}$	$(M_1^-, M_2^-, M_4^-, M_3^-, M_4^+)$	$S_{14,4}$	$(M_3^-, M_1^-, M_4^-, M_2^-, M_4^+)$
[a]$_{3,4}$	$(M_1^-, M_3^-, M_2^-, M_4^-, M_4^+)$	[a]$S_{15,4}$	$(M_3^-, M_2^-, M_1^-, M_4^-, M_4^+)$
[a]$S_{4,4}$	$(M_1^-, M_3^-, M_4^-, M_2^-, M_4^+)$	[a]$S_{16,4}$	$(M_3^-, M_2^-, M_4^-, M_1^-, M_4^+)$
[a]$S_{5,4}$	$(M_1^-, M_4^-, M_2^-, M_3^-, M_4^+)$	$S_{17,4}$	$(M_3^-, M_4^-, M_1^-, M_2^-, M_4^+)$
[a]$S_{6,4}$	$(M_1^-, M_4^-, M_3^-, M_2^-, M_4^+)$	[a]$S_{18,4}$	$(M_3^-, M_4^-, M_2^-, M_1^-, M_4^+)$
$S_{7,4}$	$(M_2^-, M_3^-, M_1^-, M_4^-, M_4^+)$	$S_{19,4}$	$(M_4^-, M_1^-, M_2^-, M_3^-, M_4^+)$
$S_{8,4}$	$(M_2^-, M_3^-, M_1^-, M_4^-, M_4^+)$	[a]$S_{20,4}$	$(M_4^-, M_1^-, M_3^-, M_2^-, M_4^+)$
[a]$S_{9,4}$	$(M_2^-, M_4^-, M_1^-, M_3^-, M_4^+)$	[a]$S_{21,4}$	$(M_4^-, M_2^-, M_1^-, M_3^-, M_4^+)$
[a]$S_{10,4}$	$(M_2^-, M_4^-, M_3^-, M_1^-, M_4^+)$	[a]$S_{22,4}$	$(M_4^-, M_2^-, M_3^-, M_1^-, M_4^+)$
$S_{11,4}$	$(M_2^-, M_1^-, M_3^-, M_4^-, M_4^+)$	[a]$S_{23,4}$	$(M_4^-, M_3^-, M_1^-, M_2^-, M_4^+)$
[a]$S_{12,4}$	$(M_2^-, M_1^-, M_4^-, M_3^-, M_4^+)$	[a]$S_{24,4}$	$(M_4^-, M_3^-, M_2^-, M_1^-, M_4^+)$

[a] NP hard cycles.

Table 2 The actions of $S_{24,4}$ with corresponding times.

Actions	Time
$S_{24,4}\left(M_4^-, M_3^-, M_2^-, M_1^-, M_4^+\right)$	
Wait for part P_i at $M3$	$W_{3,i}$
Unload part P_i from $M3$	$U_{3,i}$
Move from $M3$ to $M4$	δ
Load part P_i on $M4$	$L_{4,i}$
Move from $M4$ to $M2$	2δ
Wait for part P_{i+1} at $M2$	$W_{2,i+1}$
Unload part P_{i+1} from $M2$	$U_{2,i+1}$
Move from $M2$ to $M3$	δ
Load part P_{i+1} on $M3$	$L_{3,i+1}$
Move from $M3$ to $M1$	2δ
Wait for part P_{i+2} at $M1$	$W_{1,i+2}$
Unload part P_{i+2} from $M1$	$U_{1,i+2}$
Move from $M1$ to $M2$	δ
Load part P_{i+2} on $M2$	$L_{2,i+2}$
Move from $M2$ to I	2δ
Pick up part P_{i+3} at I	$U_{0,i+3}$
Move from I to $M1$	δ
Load part P_{i+3} on $M1$	$L_{1,i+3}$
Move from $M1$ to $M4$	3δ
Wait for part P_i at $M4$	$W_{4,i}$
Unload part P_i from $M4$	$U_{4,i}$
Move from $M4$ to O	δ
Drop part P_i at O	$L_{0,i}$
Move from O to $M3$	2δ

$$W_{2,i+1} = \max\left(0, t_{2,i+1} - \begin{pmatrix} 12\delta + U_{0,i+2} + L_{1,i+2} + W_{4,i-1} + U_{4,i-1} \\ +L_{0,i-1} + W_{3,i} + U_{3,i} + L_{4,i} \end{pmatrix}\right) \quad (3)$$

$$W_{3,i} = \max\left(0, t_{3,i} - \begin{pmatrix} 12\delta + W_{1,i+1} + U_{1,i+1} + L_{2,i+1} + U_{0,i+2} \\ +L_{1,i+2} + W_{4,i-1} + U_{4,i-1} + L_{0,i-1} \end{pmatrix}\right) \quad (4)$$

$$W_{4,i} = \max\left(0, t_{4,i} - \begin{pmatrix} 12\delta + W_{2,i+1} + U_{2,i+1} + L_{3,i+1} + W_{1,i+2} \\ +U_{1,i+2} + L_{2,i+2} + U_{0,i+3} + L_{1,i+3} \end{pmatrix}\right) \quad (5)$$

The waits of the robot at different machines are interdependent on each other for all cycles. In the present case, to find the robot wait at M_1 ($W_{1,i+2}$) the wait at M_2, M_3 and M_4 ($W_{2,i+1}$, $W_{3,i}$ and $W_{4,i-1}$) must be known. The wait at M_2 depends on the wait on M_3 and M_4, while the wait at M_3 depends on the wait at M_1 and M_4. Finally the wait at M_4 depends on the wait at M_1 and M_2. Therefore, the waits of this cycle cannot be calculated directly. The same can be noticed for the cycles marked with 'a' in Table 1. Accordingly, the waiting times and the CT can only be determined by simulating the robot actions for the robot cycle under consideration.

2.2.3. Simulation of robot cycles

In the beginning of the simulation, all the machines are considered idle where jobs enter the cell one by one. When all the machines in the cell are occupied and simulation is made after reaching the steady state; the cycle time is then calculated. A simulation matrix such as that given in Fig. 2 is composed according to the following steps.

Step 1: The job data is organized in three different matrices (the first for loading, the second for unloading and the third for processing). Each matrix consists of four rows (machines) and *n* columns (jobs). Each cell in the matrix contains the time corresponding to job *n* on machine *m*.
Step 2: Filling the simulation matrix in Fig. 2 with relevant data, each time as stated in the robot moves actions.

1. In the first row; the robot travels between machines without performing any action. The first job is picked up from the input buffer and loaded to the first machine.
2. In the second row; the robot unloads machine 1 and loads the job on machine 2. It picks up the following job and loads it on machine 1.
3. In the third row; the robot unloads machine 2 and loads the job on machine 3, unloads machine 1 and loads the job on machine 2, and picks up the following job and loads it on machine 1.
4. In the fourth row; the robot unloads machine 3 and loads the job on machine 4, unloads machine 2 and loads the job on machine 3, unloads machine 1 and loads the job on machine 2, picks up the following job and loads it on machine 1, and finally unloads machine 4 and drops the job in the output buffer.
5. In each of the following rows; the robot repeats the same actions done in the fourth row, while changing the jobs.

Step 3: The CT is calculated by summing number of rows equal to the number of jobs. The steady state is reached if two successive groups of rows (where the number of rows equal to the number of jobs) have the same total time. The total time in this case is equal the cycle time.

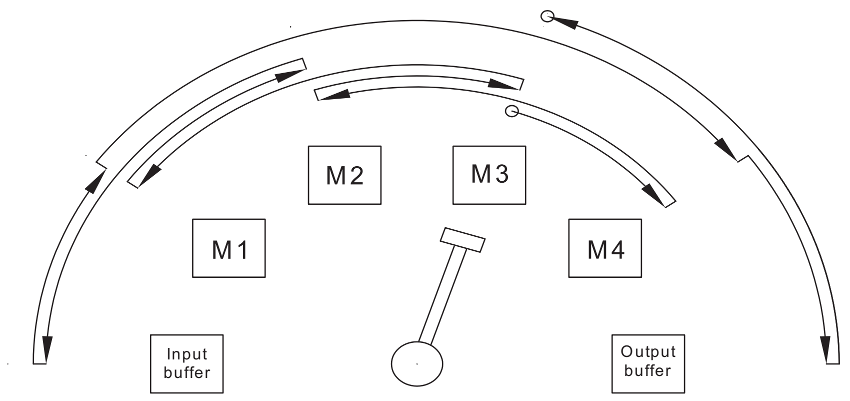

Figure 1 The movements done by the robot in $S_{24,4}$.

$$\begin{bmatrix}
0 & 0 & \delta & 0 & 2\delta & 0 & 0 & \delta & 0 & 2\delta & 0 & 0 & \delta & 0 & 2\delta & U_{0,i+3} & \delta & L_{1,i+3} & 3\delta & 0 & 0 & \delta & 0 & 2\delta \\
0 & 0 & \delta & 0 & 2\delta & 0 & 0 & \delta & 0 & 2\delta & W_{1,i+2} & U_{1,i+2} & \delta & L_{2,i+2} & 2\delta & U_{0,i+3} & \delta & L_{1,i+3} & 3\delta & 0 & 0 & \delta & 0 & 2\delta \\
0 & 0 & \delta & 0 & 2\delta & W_{2,i+1} & U_{2,i+1} & \delta & L_{3,i+1} & 2\delta & W_{1,i+2} & U_{1,i+2} & \delta & L_{2,i+2} & 2\delta & U_{0,i+3} & \delta & L_{1,i+3} & 3\delta & 0 & 0 & \delta & 0 & 2\delta \\
W_{3,i} & U_{3,i} & \delta & L_{4,i} & 2\delta & W_{2,i+1} & U_{2,i+1} & \delta & L_{3,i+1} & 2\delta & W_{1,i+2} & U_{1,i+2} & \delta & L_{2,i+2} & 2\delta & U_{0,i+3} & \delta & L_{1,i+3} & 3\delta & W_{4,i} & U_{4,i} & \delta & U_{0,i} & 2\delta \\
W_{3,i} & U_{3,i} & \delta & L_{4,i} & 2\delta & W_{2,i+1} & U_{2,i+1} & \delta & L_{3,i+1} & 2\delta & W_{1,i+2} & U_{1,i+2} & \delta & L_{2,i+2} & 2\delta & U_{0,i+3} & \delta & L_{1,i+3} & 3\delta & W_{4,i} & U_{4,i} & \delta & U_{0,i} & 2\delta \\
W_{3,i} & U_{3,i} & \delta & L_{4,i} & 2\delta & W_{2,i+1} & U_{2,i+1} & \delta & L_{3,i+1} & 2\delta & W_{1,i+2} & U_{1,i+2} & \delta & L_{2,i+2} & 2\delta & U_{0,i+3} & \delta & L_{1,i+3} & 3\delta & W_{4,i} & U_{4,i} & \delta & U_{0,i} & 2\delta \\
W_{3,i} & U_{3,i} & \delta & L_{4,i} & 2\delta & W_{2,i+1} & U_{2,i+1} & \delta & L_{3,i+1} & 2\delta & W_{1,i+2} & U_{1,i+2} & \delta & L_{2,i+2} & 2\delta & U_{0,i+3} & \delta & L_{1,i+3} & 3\delta & W_{4,i} & U_{4,i} & \delta & U_{0,i} & 2\delta
\end{bmatrix}$$

Figure 2 The robot actions matrix in $S_{24,4}$.

2.3. The developed genetic algorithm

The following steps are taken within the context of GA solution.

Step 1: Encoding the solution and generating the initial population.

A Number of chromosomes equal to the population size are generated to represent the sequences at which the parts enter the cell. The length of each chromosome is n genes, where n equals the sum of the numbers forming the ratios of MPS. Genes are filled into the chromosome randomly while maintaining the number of genes corresponding to each part equal the corresponding ratio of that part in the MPS. As an example: if the ratio of MPS is $(2,3,4)$ then $n = 9$ (chromosome length), the number of units from part 1 is 2, the number of units from part 2 is 3 and the number of part units from 3 is 4. Consequently, the resulted chromosome will be $\{1\,2\,3\,2\,3\,1\,3\,2\,3\}$.

Step 2: Evaluating the chromosomes.

Different chromosomes are evaluated using fitness function representing the CT. The fitness function is calculated from simulation with an objective of minimizing the cycle time.

Step 3: Crossover operator.

Crossover is the process of combining two chromosomes (parents) having good function to produce better chromosomes (children). Set Partition crossover operator is used in the present work.

In set partition cross over operator, the first two non-empty sets of numbers are created and filled randomly with part numbers $(1,2,3,\ldots,n)$; each number should be placed in one set only, while each set is assigned to one of the parents. The genes of the parents are scanned sequentially starting by the first gene in the first parent then the first gene in the second parent then the second gene in the first parent then the second gene of the second parent. If a gene of certain parent matches any number in the set of that parent it will be located in the new individual. After scanning all the genes in the two parents, the child produced will have the same length of parents while maintaining the number of genes corresponds to each part and equals the ratio of that part in the MPS.

Step 4: Mutation operator.

Mutation is the process of introducing some variations in the population in order to change the search area and make sure that the solution is not trapped in local optimum. The mutation operator used in the present work is the Inversion method.

Table 3 Example of inversion mutation operator.

Original gene	2	3	1	4	2	3	1	2	4	3
Modified gene	2	3	1	1	3	2	4	2	4	3

In inversion two positions in the chromosome are selected randomly and the genes between these positions are inverted. An example is illustrated in Table 3.

Step 5: Termination condition.

The stopping criterion used in present work is specified Stull number of generations. Steps 2–4 are repeated until termination condition reached.

The GA Parameters used in the present research are as follow:

- Population size, 20.
- Number of generations, 100.
- Stull generations, 50.
- Crossover rate, 80%.
- Mutation rate, 20%.

2.4. Full enumeration solution

The values are filled into a matrix where each row contains different sequence of parts. The cycle times of all sequences are then calculated using simulation. The sequence which has the minimum CT is considered the optimum sequence for a given robot-move cycle. This process is repeated for all the robot-moves cycles. The optimal values which are obtained from the full enumeration of the all possible sequences are compared with the results obtained by the application of GA to evaluate the performance of the later method.

3. Results and discussion

The proposed GA was used to solve the scheduling robot moves and job sequencing in four machines blocking robotic cells producing identical or different parts to minimize CT. The performance of the algorithm was tested against the results obtained by full enumeration. The effect of processing time to robot travel time ratio on the optimum CT is then investigated.

3.1. Evaluating the performance of the developed genetic algorithm

The results obtained by the proposed GA are evaluated against the results of full enumeration. The deviation from the optimal

Table 4 The results of evaluating the proposed GA.

Number of parts	Number of sequences	Min. value of accuracy attained	Enumeration time	GA time
5	120	100%	2.5 s	44 s
8	40,320	97.32%	15.92 min	1.18 min
9	362,880	98.44%	8.34 h	1.2 min
10 (MPS)	12,600	100%	5.42 h	1.3 min
12 (MPS)	27,720	100%	9 h	1.3 min
10	3,628,800	N/A	N/A	1.28 min

CT and the accuracy of the obtained CT from GA is calculated from the following equations:

$$\text{Deviation} = \text{Calculated cycle time } \mathrm{CT}_c - \text{Optimum cycle time } \mathrm{CT}_o \quad (6)$$

$$\text{Accuracy} = 100\% - \frac{\mathrm{CT}_c - \mathrm{CT}_o}{\mathrm{CT}_o} \times 100 \quad (7)$$

The consistency of the results is evaluated based on the results of solving ten problems (eight parts) five times. The results of the evaluation are given in Table 4.

In general, the GA's run time to reach the best solution is considerably small compared to time taken for full enumeration especially for large number of parts. The GA solution maintains considerably high accuracy when the number of parts increases more than 10 parts. The full enumeration technique fails to reach a solution even for very long run time.

The results of solving six problems using genetic algorithm are shown in Fig. 3. Problems (1–3) are 15 parts problems while problems (4–6) are 20 parts problems. Each problem was solved 10 times. The travel time is 3 unit time and the processing time is uniformly distributed between (1:100) unit time. It can be concluded from the figure that the maximum deviation in the cycle time calculated by the genetic algorithm is 3.9%. The results prove the consistency of the present algorithm for relatively large problems. No further analysis was possible to evaluate the accuracy of the results obtained by the algorithm since enumeration procedure failed to reach the solution beside the lack of bench mark problems in literature.

3.2. Relationship between (t_p/t_t) and the optimum robot moves cycle

The aim of this section is to determine the relation between t_p/t_t (processing time/robot travel time) and robot cycle that yields the optimum cycle time. The results of processing five parts and eight parts are given in Figs. 4 and 5 respectively. The optimum robot moves cycle has the same pattern in both cases. At low ratio of (t_p/t_t) (<0.71) cycle #1 is superior to other robot cycles as it yields the optimum cycle time. This is due to the fact that cycle #1 is robot dominant and meanwhile has the smallest number of moves (10 moves).

At ratios around 1 ($0.95 \leqslant (t_p/t_t) \leqslant 1.02$) cycles #1, 11, 13, 19 and 21 give the optimum cycle times. Although cycle #1 has the minimum number of moves, however, cycles #11, 13 and 19 have the next lowest number of moves with small difference compared to cycle #1, and consequently give the optimum CT for the same reasons. Starting from $t_p/t_t = 1.02$ to less than 5, cycle #21 yields the optimum cycle time and is superior to all other cycles. It is noticed that cycle #21 is mainly robot dominant in most of the cases and turns to be process dominant in few other cases. In case where Cycle #21 is robot dominant; it yields the optimum cycle time due to the least number of moves it contains. When the process is dominant, it remains the optimum, because after adding the robot waiting time to the action time, yet the total move time remains considerably small. Therefore, the difference in the number of moves in cycle #21 compared to other robot dominant cycles (which are cycles #6, 9 and 24) keep the superiority of cycle #21.

When ($4.92 \leqslant (t_p/t_t) \leqslant 5$), cycles #9, 21 and 24 yield the optimum CT. This range may be considered as transitional range of (t_p/t_t) where more than one cycle have the same CT knowing that cycles #9 and 24 are robot dominant. For ($5 \leqslant (t_p/t_t) \leqslant 7.14$) cycles #9 and 24 remain robot dominant and optimum since they have the same number of moves (14 moves). Yet for some ratios of (t_p/t_t) associated with small travel time (0.5 unit times), cycle #9 become process dominant

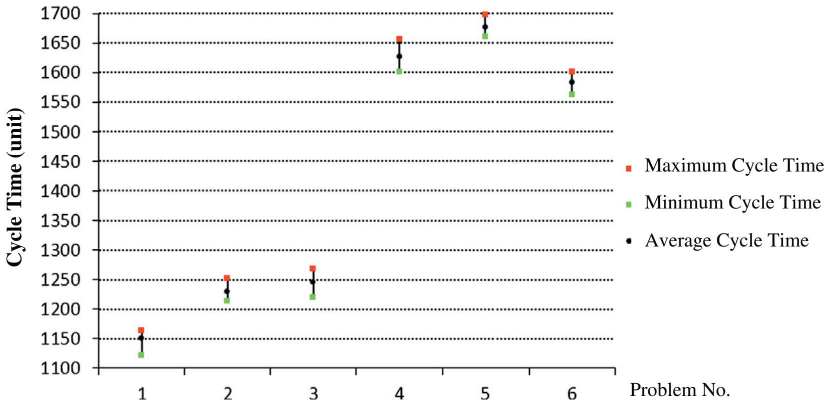

Figure 3 The maximum, minimum, and average cycle time in 15 and 20 parts problems.

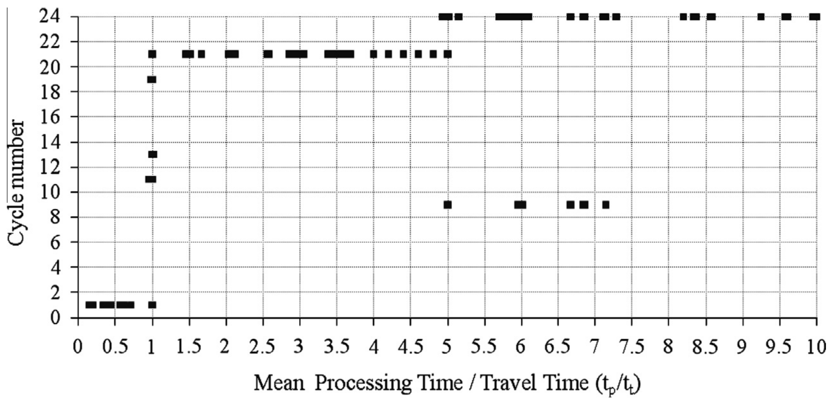

Figure 4 The relation between (t_p/t_t) and the optimum cycle (five parts).

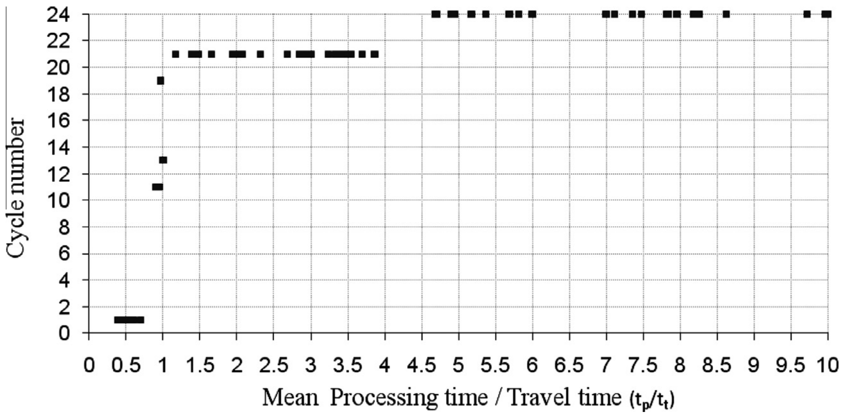

Figure 5 The relation between (t_p/t_t) and the optimum cycle (eight parts).

and hence is no more optimum and therefore, leaving cycle #24 to be the sole optimum cycle. For ratios higher than 7.14 all cycles are process dominant and again cycle #24 is the only optimum cycle.

The results shown in Fig. 5 for eight parts problems are quite similar to the results in Fig. 4 for five parts problems except that cycle #9 is no more optimum in the range $(5 \leqslant (t_p/t_t) \leqslant 7.14)$. It can be concluded that, the number of parts under processing does not affect the robot cycles that yield the optimum cycle time.

3.3. Robot utilization and robot moves cycles

The utilization of different cycles is calculated based on the following equation.

$$\text{Utilization} = \frac{\text{Total action time}}{\text{Cycle time}} \tag{8}$$

The results shown in Fig. 6 show that the robot utilization is 100% in nine robot cycles. These cycles are all robot dominant as mentioned before. Cycle #1 has the lowest robot utilization among all cycles (92.59%).

Fig. 7 represents the utilization of the robot within the processing time is equal to the travel time. Cycles 6, 9, 10, 12, 14, 16, 20, 21, and 24 are robot dominant cycles. In this case there are five optimum cycles which are cycles #1, 11, 13, 19, and 21. Although these cycles have the same cycle time (350 unit time), they have different robot utilizations. The only exception is cycle #21 which has 100% robot utilization as shown in Fig. 8. The differences in robot utilization are related to the number of travels made by the robot. Cycle #1 has 10 move distances, while cycles #11, 13, and 19 have 12 move distances and cycle #21 has 14 move distances. Although the robot travels larger distances in cycle #21, it waits for equal times in other cycles, which yields the same cycle time.

A number of experiments were conducted to study the behavior of a number of selected cycles which have yielded the optimum cycle time at different levels of the value of (t_p/t_t) while scheduling five different types of products on four machine robotic cell. The results are given in Table 5 which includes various ratios of (t_p/t_t) with the other related resulted which are the CT, the number of robot move distances, the action time, the robot waiting time and the robot utilization. The change in (t_p/t_t) is made by once varying t_p while keeping t_t constant and by keeping t_p constant while varying the values of t_t.

NRD = number of robot moved distances

It is evident from the results that, when $(t_p/t_t) \leqslant 1.0$, there are many cycles that yield the optimum CT with robot utilization. The higher the number of robot moves the higher the robot utilization. Knowing that these cycles are robot dominant and the processing time is small compared to travel time, the

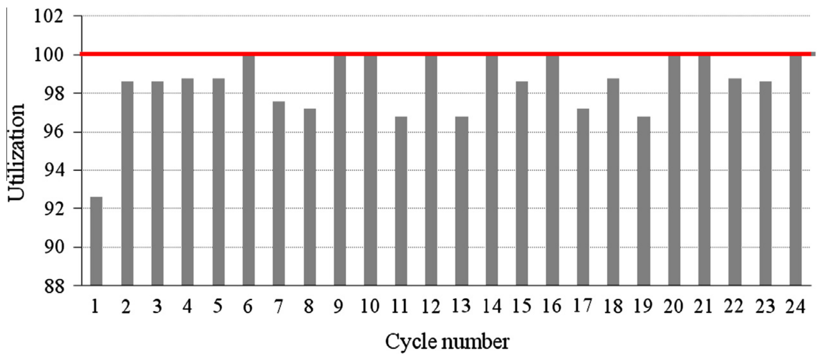

Figure 6 The robot utilization for $(t_p/t_t) = 1/5$.

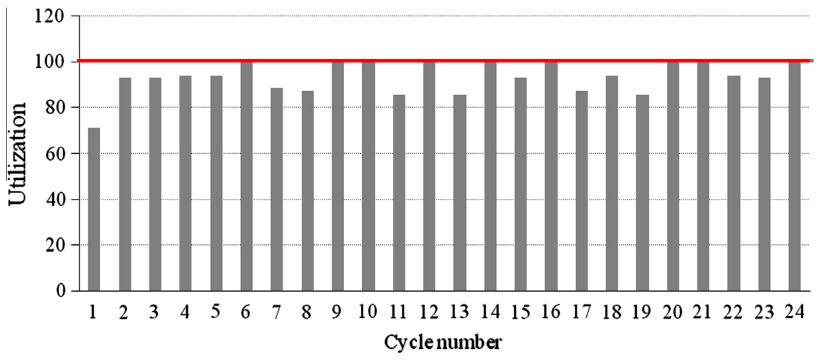

Figure 7 The robot utilization for $(t_p/t_t) = 5/5$.

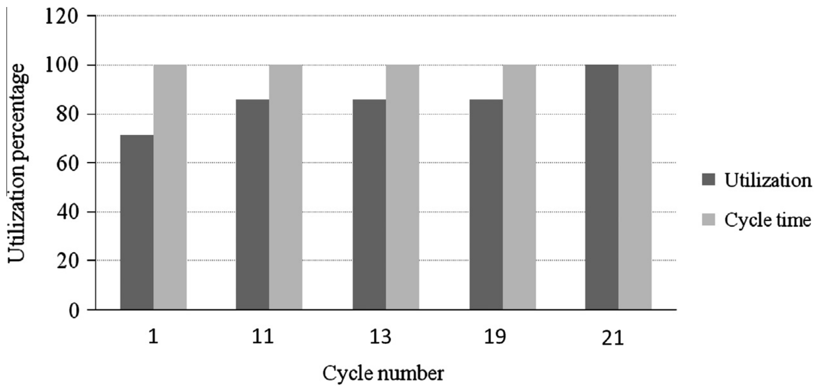

Figure 8 The robot utilization percentage for optimum cycles for $(t_p/t_t) = 5/5$.

cycle time is the same while utilization differs. Generally, the only decisive factor in determining the optimum cycles is the value of (t_p/t_t) regardless of the individual values of t_p and t_t. An important remark can be observed from Table 5 which is, by increasing the processing time or (t_p/t_t) the cycles become process dominant and the number of cycles yielding optimum CT decreases. Cycle #24 proved to perform well at high values of (t_p/t_t). The robot utilization continually decreases at values of $(t_p/t_t) > 10$ reaching considerably small values at high values of (t_p/t_t).

Based on the degradation phenomena of robot utilization at high values of (t_p/t_t), it can be concluded that the use of robot is economically justified in flow-shop manufacturing cells when (t_p/t_t) is small. The use robots become less economically justifiable at high values of (t_p/t_t) and other types of material handling equipment may be recommended.

4. Conclusions and recommendations

A genetic algorithm was developed and used to solve the problem of scheduling robot moves and jobs sequencing in four-machines blocking robotic cells producing identical or different jobs to minimize the cycle time. The developed GA was tested against the full enumeration procedure and showed high performance in solving the problem.

Table 5 Detailed results of different Cycles yielding optimum CT.

t_p/t_t	Cycle	Cycle time	NRD	Action time	Waiting time	Utilization %	t_p/t_t	Cycle	Cycle time	NRD	Action time	Waiting time	Utilization %	t_p/t_t	Cycle	Cycle time	NRD	Action time	Waiting time	Utilization %	t_p/t_t	Cycle	Cycle time	NRD	Action time	Waiting time	Utilization %
1/1= 1.0	1	70	10	50	20	71.4	1/3 = 0.33	1	170	10	150	20	88.2	1/5 = 0.2	1	270	10	250	20	92.6	1/7 = 0.14	1	370	10	350	20	94.6
	9	80	16	80	0	100		9	240	16	240	0	100		9	400	16	400	0	100		9	560	16	560	0	100
	11	70	12	60	10	85.7		11	190	12	180	10	94.7		11	310	12	300	10	96.8		11	430	12	420	10	97.7
	13	70	12	60	10	85.7		13	190	12	180	10	94.7		13	310	12	300	10	96.8		13	430	12	420	10	97.7
	19	70	12	60	10	85.7		19	190	12	180	10	94.7		19	310	12	300	10	96.8		19	430	12	420	10	97.7
	21	70	14	70	0	100		21	210	14	210	0	100		21	350	14	350	0	100		21	490	14	490	0	100
	24	80	16	80	0	100		24	240	16	240	0	100		24	400	16	400	0	100		24	560	16	560	0	100
5/1 = 5.0	1	150	10	50	100	33.3	5/3 = 1.67	1	250	10	150	100	60	5/5= 1.0	1	350	10	250	100	71.4	5/7 = 0.72	1	450	10	350	100	77.8
	9	80	16	80	0	100		9	240	16	240	0	100		9	400	16	400	0	100		9	560	16	560	0	100
	11	115	12	60	55	52.2		11	230	12	180	50	78.3		11	350	12	300	50	85.7		11	470	12	420	50	89.4
	13	110	12	60	50	54.5		13	230	12	180	50	78.3		13	350	12	300	50	85.7		13	470	12	420	50	89.4
	19	115	12	60	55	52.2		19	230	12	180	50	78.3		19	350	12	300	50	85.7		19	470	12	420	50	89.4
	21	80	14	70	10	87.5		21	210	14	210	0	100		21	350	14	350	0	100		21	490	14	490	0	100
	24	80	16	80	0	100		24	240	16	240	0	100		24	400	16	400	0	100		24	560	16	560	0	100
20/1= 20.0	1	450	10	50	400	11.1	20/3 = 6.7	1	550	10	150	400	27.3	20/5 = 4.0	1	650	10	250	400	38.5	20/7 = 2.86	1	750	10	350	400	46.7
	9	170	16	80	90	47.1		9	240	16	240	0	100		9	400	16	400	0	100		9	560	16	560	0	100
	11	340	12	60	280	17.6		11	420	12	180	240	42.9		11	500	12	300	200	60		11	620	12	420	200	67.7
	13	260	12	60	200	23.1		13	380	12	180	200	47.4		13	500	12	300	200	60		13	620	12	420	200	67.7
	19	340	12	60	280	17.6		19	420	12	180	240	42.9		19	500	12	300	200	60		19	620	12	420	200	67.7
	21	230	14	70	160	30.4		21	290	14	210	80	72.4		21	350	14	350	0	100		21	490	14	490	0	100
	24	120	16	80	40	66.7		24	240	16	240	0	100		24	400	16	400	0	100		24	560	16	560	0	100
50/1 = 50.0	1	1050	10	50	1000	4.8	50/3 = 16.7	1	1150	10	150	1000	13	50/5 = 10.0	1	1250	10	250	1000	20	50/7 = 7.14	1	1350	10	350	1000	25.9
	9	395	16	80	315	20.3		9	435	16	240	195	52.2		9	475	16	400	75	83.2		9	560	16	560	0	100
	11	790	12	60	730	7.6		11	870	12	180	690	20.7		11	950	12	300	650	31.6		11	1030	12	420	610	40.8
	13	560	12	60	500	10.7		13	680	12	180	500	26.5		13	800	12	300	500	37.5		13	920	12	420	500	45.7
	19	790	12	60	730	7.6		19	870	12	180	690	20.7		19	950	12	300	650	31.6		19	1030	12	420	610	40.8
	21	530	14	70	460	13.2		21	590	14	210	380	35.6		21	650	14	350	300	53.8		21	710	14	490	220	69
	24	270	16	80	190	29.6		24	310	16	240	70	77.4		24	400	16	400	0	100		24	560	16	560	0	100
100/1 = 100.0	1	2050	10	50	2000	2.4	100/3 = 33.3	1	2150	10	150	2000	7	100/5= 20.0	1	2250	10	250	2000	11.1	100/7 = 14.29	1	2350	10	350	2000	14.9
	9	770	16	80	690	10.4		9	810	16	240	570	24.9		9	850	16	400	450	43.5		9	890	16	560	330	60.4
	11	1540	12	60	1480	3.9		11	1620	12	180	1440	11.1		11	1700	12	300	1400	17.6		11	1780	12	420	1360	23.6
	13	1060	12	60	1000	5.7		13	1180	12	180	1000	15.3		13	1300	12	300	1000	23.1		13	1420	12	420	1000	29.6
	19	1540	12	60	1480	3.9		19	1620	12	180	1440	11.1		19	1700	12	300	1400	17.6		19	1780	12	420	1360	23.6
	21	1030	14	70	960	6.8		21	1090	14	210	880	19.3		21	1150	14	350	800	30.4		21	1210	14	490	720	40.5
	24	520	16	80	440	15.4		24	560	16	240	320	42.9		24	600	16	400	200	66.7		24	640	16	560	80	87.5

Enumeration was efficient only in scheduling small size problems incorporating a limited number of jobs in a four machine robotic cell. As the number of jobs increases, enumeration fails to reach a solution. On the other hand, GA proved to be efficient in solving problems with large number of parts in considerably short time, whether the problems are unconstrained or constrained (MPS) and with identical or non-identical parts in four machines robotic cells.

The ratio between the mean processing time and robot travel time plays major role in determining which robot cycle has the minimum cycle time. In general, cycle #24 gives the minimum cycle time for values of (t_p/t_t) higher than 5 which agree with previous research in the field. Although cycle #9 yielded optimal cycle time in the same range of (t_p/t_t), cycle #24 is most dominant. For lower values of (t_p/t_t), other cycles show better cycle times. Among these cycles are cycles number 11, 19, 13, and 21. It was shown that cycle #21 is superior to other cycles for (t_p/t_t) in the range between 1.02 and 5. Cycle #1 is the optimum robot cycle and superior to other cycles for low ratios of (t_p/t_t) lower than 0.95.

The cycle/s that yields the optimum robot cycle time are independent of the number of jobs being processed. Although more than one cycle may give the minimum robot cycle time, the best cycle is determined based on the robot utilization, the smaller the utilization the better the cycle. Also robot utilization determines the optimum robot cycle(s) when all cycles are process dominant.

For future work, more efforts may be directed to solve the *k-unit* cycles, where the number of robot cycles and corresponding NP-hard cycles are considerably higher than that of the present case. It is also worthy solving problems that consider skipping one or more machine for some parts and that which consider picking up, dropping and loading/unloading operations.

References

[1] Sethi SP, Sriskandarajah C, Sorger G, Blazewicz J, Kubiak W. Sequencing of parts and robot moves in a robotic cell. Int J Flex Manuf Syst 1992;4:331–58.

[2] Carlier Jacques, Haouari Mohamed, Kharbeche Mohamed, Moukrima Aziz. An optimization-based heuristic for the robotic cell problem. Eur J Oper Res 2010;202:636–45.

[3] Soukhal A, Martineau P. Resolution of a scheduling problem in a flowshop robotic cell. Eur J Oper Res 2005;161:62–72.

[4] Hall Nicholas G, Kamoun Hichem, Sriskandaraja Chelliah. Scheduling in robotic cells: classification, two and three machine cells. Oper Res 1997;45(3):421–39.

[5] Brauner N, Finke G. Cycles and permutations in robotic cells. Math Comput Modell 2001;34:565–91.

[6] Crama Yves, Klundert Joris Van De. Cyclic scheduling of identical parts in a robotic cell. Oper Res 1997;45(6):952–65.

[7] Neil Geismar H, Dawande Milind, Sriskandarajah Chelliah. Robotic cells with parallel machines: throughput maximization in constant travel time cells. J Sched 2004;7:375–95.

[8] Alcaide David, Chu Chengbin, Kats Vladimir, Levner Eugene, Sierksma Gerard. Cyclic multiple-robot scheduling with time-window constraints using a critical path approach. Eur J Oper Res 2007;177:147–62.

[9] Sadek Yomna, El-Kharbotly Amin, Afia Nahid. Production scheduling in robotic cells. In: Proceedings of the 37th international conference on computers and industrial engineering, Alexandria, Egypt, October 20–23; 2007.

[10] Kamoun Hichem, Hall Nicholas G, Sriskandarajah Chelliah. Scheduling in robotic cells: heuristics and cell design. Oper Res 1999;47(6):821–35.

[11] Sriskandaraja Chelliah, Drobouchevitch Inna, Sethi Suresh R, Chandrasekaran Ramaswamy. Scheduling multiple parts in a robotic cell served by a dual-gripper robot. Oper Res 2004;52(1):65–82.

[12] Dawande Milind, Neil Geismar H, Sethi Suresh P, Sriskandarajah Chelliah. Sequencing and scheduling in robotic cells: recent developments. J Sched 2005;8:387–426.

[13] Brauner Nadia. Identical part production in cyclic robotic cells: concepts, overview and open questions. Discrete Appl Math 2008;156:2480–92.

M.M.S Abdulkader is a teaching and research assistant and Ph.D. student at the Design and Production Engineering Dept., Faculty of Engineering, Ain Shams University, Cairo, Egypt. He has a degree in Mechanical Engineering (2006) and M.Sc. in Production planning (2011) from Ain Shams University.

Mohamed M. El-Beheiry is an assistant professor at Ain Shams University, Faculty of Engineering, Cairo, Egypt. He has a degree in Mechanical Engineering (1994) and an MSc in Production Planning and Scheduling (1999) from the Ain Shams University. He also has a PhD degree in Supply Chain Management (2004). He works as a His research areas include production scheduling, inventory control, supply chain dynamics, and simulation.

Nahid H. Afia is an associate professor at the Design and Production Engineering Dept., Faculty of Engineering, Ain Shams University,Cairo, Egypt since 2008. She has received her Ph.D. in 1992 from Ain Shams University.

Amin K. El-Kharbotly is a professor at the Design and Production Engineering Dept., Faculty of Engineering, Ain Shams University, Cairo, Egypt since 1993. He has received his Ph.D. in 1977 from Ain Shams University.