2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

# LCM cycle based optimal scheduling in robotic cell with parallel workstations *

Jiafan Zhang, *Member, IEEE*, Xinyu Fang, and Liwei Qi

***Abstract*—Robot-centered cells play a more and more important role in the fields of incorporate automation and repetitive processing in order to increase productivity and to improve quality. In practice, it is always desired to achieve maximum/near-maximum throughput in a robotic cell. Sometimes, even a small improvement in throughput will be one of the highlighted objectives in robot scheduling, for instance, in the 3C industry of communication, computer and consumer electronics. In this paper, the scheduling problem in an *n*-workstation (*n*<=8) *m*-stage robotic cell with parallel workstations is discussed by considering the constraints in real engineering practice, including free/non-free pick-up, allowed time window, and free/non-free process. A new method is proposed for optimal scheduling by means of optimally arranging each blocked cycles in every LCM cycle (Least Common Multiple), and assembling them in a proper order. This method is applied to a practical scheduling problem for a 7-workstation robotic cell. It is shown that the throughput is dramatically enhanced after optimization in the engineering scenario. Moreover, by comparing the resultant increase in revenue with the additional equipment costs, the result diagrams also provide managerial insights into links between throughput and cell layout flexibility.**

## I. INTRODUCTION

Intensive global competition in the 3C industry of communication, computer and consumer electronics has compelled manufacturers to incorporate automation and repetitive processing for improving productivity. In order to reduce cost and improve productivity, robotic cells have been employed in many modern manufacturing systems as analytically demonstrated in the previous work [1][2].

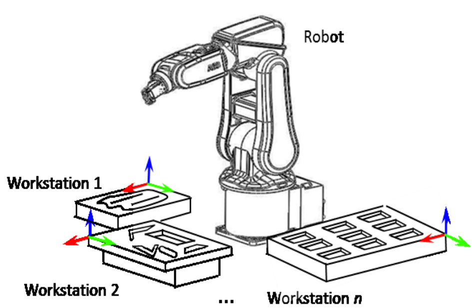


Fig. 1 General diagram of robotic cell

A robotic cell normally consists of a feed-in device, a series of processing workstations, each of which performs a different operation on a workpiece in a pre-determined order, an output device, and one or more robots that transport the parts within the cell. Fig.1 is a pictorial diagram of a general robotic cell. The robotic cell is normally flow-shop type. In a cyclic production, a workpiece, which is transported by the robot, passes through finite numbers of workstations in a pre-determined order. Apparently, maximizing the robotic cell throughput is a vital decision in competitive environments, especially in 3C industry. Because of this, an optimal activity scheduling in robot movement becomes a critical criterion to achieve the maximum productivity, as the robot always follows a pre-programmed moving sequence.

Robot moving sequence is a set of actions which the robot repeats at each cycle. A typical sequence includes loading a workpiece, travelling among workstations, unloading the workpiece, and either fully or partially waiting for the workpiece to complete its process or implementing the process. The scheduling problem has been described in the literature of classical scheduling problems. Sethi *et al.* [3][4] initially developed the necessary framework for this problem, and showed that there are *n*! 1-unit cycles in an *n*-workstation robotic cell. 1-unit cycles are regarded as being attractive since they are practical and easy to understand and control. Hall and Sriskandarajah [5] surveyed scheduling problems with blocking and no-wait conditions and classify their computational complexity. In the work of Chen and Chu [6], the scheduling problems with time window constraints were studied by using branch-and-bound, linear programming, and bi-valued graphs to find optimal multi-unit cycles. Gultekin *et al*. [7] solved two problems which are allocating operations to the machines and robot move sequence, in a two-machine robotic cell. Later they proposed an optimal solution by assuming that each part has a number of different operations to be completed [8][9]. A formulated mixed-integer linear program with common structure was framed by Brucker and Kampmeyer [10], in order to model and solve cyclic scheduling problems. Most of prior studies were implemented based on the constant or additive robot traveling time. Finding the optimal robot move sequence in a robotic cell with general travel time is already proved as an NP-hard problem by Brauner *et al* in [11]. However, the travelling time is always general in the real engineering scenarios, but the number of workstations is usually not over than 8. This makes the optimal problem solvable. Our previous work in [12] implemented the work in scheduling optimization under the consideration of general time of robot travelling, loading, unloading and processing, in real engineering practice, both for single-gripper and dual-gripper robotic cell as well. Additionally, unlike the current studies which focus on the cases where robots are used only for material handling the presented study first highlights the cases where robot itself is used for processing.

The efficient use of robotic cells with parallel workstations has been proved to increase the rate of production and thus

* The authors are with ABB Corporate Research China, Shanghai 201319, China (e-mail: jiafan.zhang@cn.abb.com; xinyu.fang@cn.abb.com; liwei.qi@cn.abb.com ).

provides a competitive advantage in productivity. Geismar et al. [13][15] found an optimal cycle for cells with parallel machines and constant travel-times for a case that is very common in practice. The expression of cycle time by LCM cycles is formulized. Later, they applied it to practice by considering an industrial problem with parallel machines, three robots, and Euclidean travel times [14]. However, each blocked cycle in their work follows the same base permutation. They did not give an indication about how to find the sweet permutation when combining all of the blocked cycle into the LCM cycle. Additionally, the constraints in real-life engineering practice, introduced in [12], were neglected.

The contribution of this study is to propose a new method for the optimal scheduling in an $n$-workstation $m$-stage ($m<n<=8$) single robotic cell with parallel workstations, based on LCM cycles by considering the constraints in real engineering practice. The result diagrams of an engineering scenario provide managerial insights into links between throughput and cell layout flexibility, by comparing the resultant increase in revenue with the additional equipment costs. In next section the problem is formally defined, and the method used in this paper is outlined in Section 3. Section 4 introduces the validation of this method through a practical case of 7-workstation robotic cell with parallel workstation. The areas for the further research and conclusions are followed in Section 5.

## II. Preliminaries and Problem Formulation

A robotic cell is, by its nature, a flow shop with a fixed number of workstations. There are many discussions on the 1-unit cycles in robotic cells with $n$ identical workstations. As shown in Fig.2, a typical 1-unit cycles refers to such a workflow that each station in the workflow $Ws_i$, $i$=1, 2, 3, …, $n$, must have one and only one leader station (i.e. $Ws_{i-1}$) in front, and one and only one follower station (i.e. $Ws_{i+1}$) behind. This type of 1-unit cycles is preferred in both single-gripper and double-gripper scenarios. In particular, dual-gripper solution seems to increase the productivity in a certain sense.

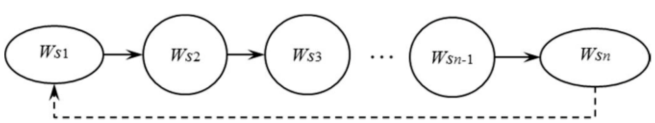

Fig. 2 Typical 1-unit cycle

A robotic cell is, by its nature, a flow shop with a fixed number of workstations. There are many discussions on the 1-unit cycles in robotic cells with $n$ identical workstations. As shown in Fig.2, a typical 1-unit cycles refers to such a workflow that each station in the workflow $Ws_i$, $i$=1, 2, 3, …, $n$, must have one and only one leader station (i.e. $Ws_{i-1}$) in front, and one and only one follower station (i.e. $Ws_{i+1}$) behind. This type of 1-unit cycles is preferred in both single-gripper and double-gripper scenarios. In particular, dual-gripper solution seems to increase the productivity in a certain sense.

Adding an identical workstation parallel to the other workstations in a particular processing stage is especially cost effective if there are a small number of stages whose processing times are significantly larger than those of the other stages. This results in a parallel robotic cell.

Fig. 3 illustrates a parallel robotic cell with $n$ workstations and $m$ stages. Each workstation is assigned as a distinct workstation in corresponding processing stage or being parallel to other workstations in that process. For each processing stage $i$, there are $k_i$ ($k_i \geq 1$) workstations. Obviously,

$$n = \sum_{i=1}^{m} k_i \tag{1}$$

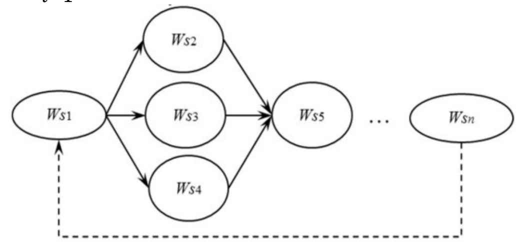

Fig. 3 Parallel workstation robotic cell

Normally several features should be figured out on each workstation, as summarized in Table I.

- If a process needs a robot to collaborate with the workstation, then it is called a 'non-free' process, and it is marked as 'N' in the property of 'RobotFree'.
- If a process has a limitation in time window, within which all operations (e.g. loading, unloading, etc.) are finished, the property 'TimeSlot' would be activated.

TABLE I
Properties of Workstations

| Properties | Value | Comments |
|---|---|---|
| $Ws_i$.Role | 'FI'/'WS'/'FO' | The type of $Ws_i$ |
| $Ws_i$.Parallel | Y/N | A unique or parallel workstation |
| $Ws_i$.ProcessStep | 1, 2, …, $m$ | Stage step of $Ws_i$ |
| $Ws_i$.ParallelTo | 1, 2, …, $m$ | The stage $Ws_i$ is parallel to |
| $Ws_i$.TimeSlot | Number value | Allowed pick-up time window |
| $Ws_i$.RobotFree | Y/N | The robot is occupied during process or not |

Similar as a 1-unit robotic cell, production process for an identified part in a parallel workstation cell also be regarded as the combination of four main sub-tasks: travelling, loading, unloading and processing.

Let $\delta T_{t,ij}$ denote the travelling time between any two workstations $Ws_i$ and $Ws_j$, $1 \leq i, j \leq n$. The general travel time matrix can be expressed as:

$$\mathbf{\Delta T}_t = \begin{bmatrix} 0 & \delta T_{t,12} & \cdots & \delta T_{t,1n} \\ \delta T_{t,12} & 0 & & \delta T_{t,2n} \\ \vdots & & \ddots & \vdots \\ \delta T_{t,n1} & \delta T_{t,n2} & \cdots & 0 \end{bmatrix} \tag{2}$$

Sometimes, if the travel between a certain pair of workstations is not available due to physical obstacles or unavailable robot motion, then $\delta T_{t,ij}=\infty$.

The process time of $Ws_i$ is denoted as $\delta T_{p,i}$, $\forall i$. Therefore, the process time matrix can be expressed as a vector:

$$\Delta \mathbf{T}_p = [\delta T_{p,1} \quad \delta T_{p,2} \quad \cdots \quad \delta T_{p,n}] \tag{3}$$

Similarly, a sophisticated model has different values for loading and unloading time at each workstation $Ws_i$ can be denoted as $\delta T_{l,i}$ and $\delta T_{u,i}$ respectively:

$$\Delta \mathbf{T}_l = [\delta T_{l,1} \quad \delta T_{l,2} \quad \cdots \quad \delta T_{l,n}] \tag{4}$$

$$\Delta \mathbf{T}_u = [\delta T_{u,1} \quad \delta T_{u,2} \quad \cdots \quad \delta T_{u,n}] \tag{5}$$

A natural and widely used measure of productivity is throughput which is defined as the number of finished workpieces produced in a unit time slot. Normally, a robotic cell refers to the production of finished workpieces by repeating a fixed sequence of robot moves, until the required production is complete. For such a cyclic production, each element of the cycle time matrix can be expressed as:

$$\delta T_{ij} = \begin{cases} c_i(\delta T_{l,i} + \delta T_{u,i} + \delta T_{p,i}) & \text{if } i = j \\ \sigma_{ij}\delta T_{t,ij} & \text{if } i \neq j \end{cases} \tag{6}$$

where,

$$\mathbf{c} = [c_1 \quad c_2 \quad \cdots \quad c_n] \tag{7}$$

$$\boldsymbol{\sigma} = \begin{bmatrix} 0 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & 0 & \cdots & \sigma_{2n} \\ \vdots & & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & 0 \end{bmatrix} \tag{8}$$

They represent the times of the processing on $Ws_i$ and travelling between $Ws_i$ and $Ws_j$. For a 1-unit cycle, each process occurs only once during the part whole producing process, namely $c_i$=1.

In this kind of cyclic production, cycle time is referred to the duration during which the sequence of the movements and operations is completed in a normal iteration to produce one workpiece. The straightforward approach for computing the cycle time is to sum all of the times in a normal iteration of the part production. It can be calculated as:

$$\begin{aligned} T &= \sum_{i=1}^{n}\sum_{j=1}^{n}\delta T_{ij} \\ &= \sum_{i=1}^{n} c_i(\delta T_{l,i} + \delta T_{u,i} + \delta T_{p,i}) + \sum_{i=1}^{n}\sum_{j=1}^{n}\sigma_{ij}\delta T_{t,ij} \end{aligned} \tag{9}$$

In general, the shorter cycle time, the larger throughput. Maximizing the throughput is equivalent to pursue the minimum cycle time:

$$\gamma : \min(T) \tag{10}$$

By optimizing the robotic cell layout to minimize $\delta T_{t,ij}$ in Eq.7 is one alternative. Many solutions can be found in relevant literature [16]-[18], and an efficient solution is implemented in our previous work [19].

This work focuses on the cyclic schedules optimization, which is towards optimizing the utilization of the robot and workstations by means of optimal $\sigma_{ij}$ in Eq.7. Perhaps, the "wasteful" robot actions such as unnecessary waiting at a location or moving to a location without performing at least one of the loading or unloading operations would be introduced by the intervening processes. Nevertheless, it is not necessary to be considered since the interest is to maximize the throughput of the cell. In addition, process sequence also relates to the scheduling problem to shorten the processing time $\delta T_{p,i}$ in Eq.7, but there is normally little flexibility in process optimization leading to relatively less influence on the throughput due to the strict nature of process operation.

## III. Preliminaries and Problem Formulation

In an $n$-workstation robotic cell with a 1-unit cycle, a workpiece cannot be moved from its current workstation to the next one if the next workstation is occupied in case of a single gripper application. In this case, the activity consisting of a sequence of actions is a quite popular concept, with notation $A_i$, $1 \leq i \leq n$. It normally involves following actions:

- The robot moves to $Ws_{i-1}$;
- The robot uploads a workpiece from $Ws_{i-1}$;
- The robot travels from $Ws_{i-1}$ to $Ws_i$;
- The robot loads this workpiece onto $Ws_i$;
- The robot is occupied on $Ws_i$ if it is a non-free process;
- The robot cell identifies the status for this activity;
- Waiting for next activity.

Given a cell with parallel workstations and $m$ stages, a multi-unit cycle is normally employed. Unlike the 1-unit cycle cell, the number of distinct cycles of a parallel workstation robotic cell is extremely large, owing to its complexity. Therefore, the field of study is generally narrowed to a particular subset of $k$-unit cycles, called blocked cycle [15]. It highly structure subclass of multi-unit cycles and are natural generalization of 1-unit cycles. To promise the feasibility of the blocked cycle, two necessary conditions are involved:

- Each workstation is loaded once it is unloaded with a specific activity order, called base permutation;
- For each stage $i$, $i$=1, 2, …, $m$, each of its workstations has the same number of activities between its loading and its unloading for each usage.

Thus, each block cycle is a 1-unit cycle with $m$-1 activities. Note that starting with an initial state, the robot performs each of $\lambda$*($m$-1) activities, exactly once, and the final state of the cell is identical to the initial state. Here $\lambda$ is the least common multiple of $k_i$, as $\lambda$=LCM($k_1$, $k_2$, …, $k_m$). LCM cycle is a kind of $\lambda$-unit cycle, which has been proven to greatly increase throughput of cell.

***Example:***

Consider a 6-workstation robotic cell with 4 stages, as shown in Fig.4, here come two possible 2-unit cycles:

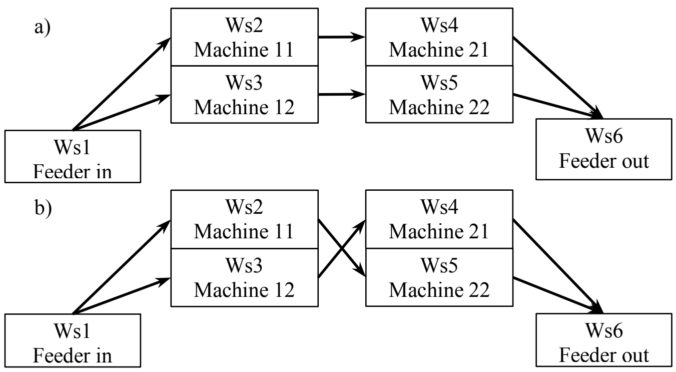


Fig. 4 Two 2-unit cycles in a 4-stage robotic cell with parallel workstations

TABLE II
LCM CYCLES AND BLOCKED CYCLES FOR A ROBOTIC CELL

| LCM cycles $\boldsymbol{S}_{\mathrm{LCM},i}$ ($i$=1, 2, …, $\xi$) |
|---|
| $\boldsymbol{S}_{\mathrm{LCM},1}$=[ $\boldsymbol{S}_{\mathrm{BLK},11}$, $\boldsymbol{S}_{\mathrm{BLK},12}$, …, $\boldsymbol{S}_{\mathrm{BLK},1\lambda}$] |
| $\boldsymbol{S}_{\mathrm{LCM},2}$=[ $\boldsymbol{S}_{\mathrm{BLK},21}$, $\boldsymbol{S}_{\mathrm{BLK},22}$, …, $\boldsymbol{S}_{\mathrm{BLK},2\lambda}$] |
| … |
| $\boldsymbol{S}_{\mathrm{LCM},\xi}$=[ $\boldsymbol{S}_{\mathrm{BLK},\xi 1}$, $\boldsymbol{S}_{\mathrm{BLK},\xi 2}$, …, $\boldsymbol{S}_{\mathrm{BLK},\xi\lambda}$] |

Because the distinct LCM cycle of a robotic cell with parallel workstations sometimes is not unique, let $\boldsymbol{S}_{\mathrm{LCM},i}$ denote the $i^{\mathrm{th}}$ LCM cycle, $1\le i\le\xi$, and $\xi$ is the number of distinct LCM cycles. Therefore, the $j^{\mathrm{th}}$ blocked cycle of $\boldsymbol{S}_{\mathrm{LCM},i}$ can be represented as $\boldsymbol{S}_{\mathrm{BLK},ij}$ ($j$=1, 2, …, $\lambda$), with a certain permutation $p_{ij}$. The following table summarizes an example of LCM cycles for an $n$-workstation, $m$-stage robotic cell.

The cycle time of a LCM cycle $\boldsymbol{S}_{\mathrm{LCM},i}$ can be calculated as:

$$t_{LCM,i}=\sum_{j=1}^{\lambda} t_{BLK,ij} \tag{11}$$

where $t_{\mathrm{BLK},ij}$ is the cycle time of the $j^{\mathrm{th}}$ blocked cycle in the $i^{\mathrm{th}}$ LCM cycle.

For each of 1-unit cycle like blocked cycles, it has an optimal sequence $\boldsymbol{S}_{\mathrm{BLK},ij}$ individually. This process can be referred to the contributions in [12]. Let $\widehat{\boldsymbol{S}}_{\mathrm{BLK},ij}$ denote the corresponding optimal result to each blocked cycle $\boldsymbol{S}_{\mathrm{BLK},ij}$, so that the LCM cycle referring to implement each optimal blocked cycle in serial can be expressed as:

$$\widehat{\boldsymbol{S}}_{\mathrm{LCM},i}=[\widehat{\boldsymbol{S}}_{\mathrm{BLK},i1},\widehat{\boldsymbol{S}}_{\mathrm{BLK},i2},\dots,\widehat{\boldsymbol{S}}_{\mathrm{BLK},i\lambda}]$$

Definitely the cycle time of $\widehat{\boldsymbol{S}}_{\mathrm{LCM},i}$ can be calculated by Eq. 12. Compared to $\boldsymbol{S}_{\mathrm{LCM},i}$, it enhances the cell productivity rate in a certain sense.

$$t_{\widehat{\boldsymbol{S}}_{LCM,i}}=\sum_{j=1}^{\lambda} t_{\widehat{\boldsymbol{S}}_{BLK,ij}}\le\sum_{j=1}^{\lambda} t_{\boldsymbol{S}_{BLK,ij}}=t_{\boldsymbol{S}_{LCM,i}} \tag{12}$$

***Definition:***

Suppose there are two blocked cycles $\boldsymbol{A}$=($A_1$, $A_2$, …, $A_m$)|$p_A$ and $\boldsymbol{B}$=($B_1$, $B_2$, …, $B_n$)|$p_B$. There are a lot of possible ($m$+$n$) activity LCM cycles formed by versatile arrangements of activities $A_{ai}$ and $B_{bj}$ ($1\le ai\le m$, $1\le bj\le n$). If $\boldsymbol{C}$=($C_1$, $C_2$, …, $C_{m+n}$) has the shortest cycle time among them, then it is denoted as:

$$\boldsymbol{C}=\boldsymbol{A}\#\boldsymbol{B} \tag{13}$$

The note should be addressed is that the activities elements from $\boldsymbol{A}$ in any subset of $\boldsymbol{C}$ should follow the cyclic permutation $p_A$. This constraint can be described as:

Suppose $C_{\mathrm{ai}}$ and $C_{aj}$ are any two elements in $\boldsymbol{C}$, and $C_{ai}$=$A_i$, $C_{aj}$=$A_j$,

- if $1\le i<j\le m$, then $ai<aj$, $\forall ai, aj\in[1,m+n]$;
- if $1\le j<i\le m$, then there, between $C_{ai}$ and $C_{aj}$, must be $k$ activities, $k$=($m$-$i$)+($j$-1), mapping onto $A_j$+1, $A_j$+2, …, $A_m$, $A_1$, $A_2$, …, $A_{i-1}$ in series.

Similarly, the elements in $\boldsymbol{C}$ matching to $\boldsymbol{B}$ should also obey above rules.

Applying "#" operation to each of LCM cycles in above case of robotic cell, there, corresponding to $\widehat{\boldsymbol{S}}_{\mathrm{LCM},i}$, must be a $\widehat{\boldsymbol{S}}^{\#}_{\mathrm{LCM},i}=[\widehat{\boldsymbol{S}}_{\mathrm{BLK},i1}\#\,\widehat{\boldsymbol{S}}_{\mathrm{BLK},i2}\#\dots\#\widehat{\boldsymbol{S}}_{\mathrm{BLK},i\lambda}]$, $1\le i\le\xi$, that satisfies $t_{\widehat{\boldsymbol{S}}^{\#}_{LCM,i}}\le t_{\widehat{\boldsymbol{S}}_{LCM,i}}$.

As a result, the optimal production cycle of such a robotic cell can be filtered by translating the problem of throughput maximization into the objective as:

$$\gamma: t_{\widehat{\boldsymbol{S}}^{\#}_{LCM}}=\min(t_{\widehat{\boldsymbol{S}}^{\#}_{LCM,1}},t_{\widehat{\boldsymbol{S}}^{\#}_{LCM,2}},\dots,t_{\widehat{\boldsymbol{S}}^{\#}_{LCM,\xi}}) \tag{14}$$

The algorithm to search the optimal LCM cycle with the shortest cycle time is summarized in Table III.

TABLE III
ALGORITHM FOR SEARCHING THE OPTIMAL LCM CYCLE

**Input:**
Time matrix and allowable time slot;
Workstation properties;
Define the LCM cycles and blocked cycles according to a certain base permutation;

$$\boldsymbol{S}_{LCM}=\begin{bmatrix}\boldsymbol{S}_{LCM,1}\\ \boldsymbol{S}_{LCM,2}\\ \vdots\\ \boldsymbol{S}_{LCM,\xi}\end{bmatrix}=\begin{bmatrix}\boldsymbol{S}_{BLK,11},\boldsymbol{S}_{BLK,12},\dots,\boldsymbol{S}_{BLK,1\lambda}\\ \boldsymbol{S}_{BLK,21},\boldsymbol{S}_{BLK,22},\dots,\boldsymbol{S}_{BLK,2\lambda}\\ \vdots\\ \boldsymbol{S}_{BLK,\xi1},\boldsymbol{S}_{BLK,\xi2},\dots,\boldsymbol{S}_{BLK,\xi\lambda}\end{bmatrix}$$

The number of blocked cycle in each LCM cycle is $\lambda=LCM(k_1,k_2,\dots k_m)$;
The number of LCM cycle is $\xi$;
**Loop1:**
**Loop2:**
**If** $\boldsymbol{S}_{\mathrm{BLK},ij}$ meets feasible criteria and process constraints, **THEN**
By 1-unit cycle scheduling, obtain the optimal activity sequence of $\boldsymbol{S}_{\mathrm{BLK},ij}$, denoted as $\widehat{\boldsymbol{S}}_{BLK,ij}$, and the optimal permutation is denoted as $\hat{p}_{ij}$;
**END**
**END Loop2**
Do # operation and obtain all possible $\boldsymbol{S}^{\#}_{LCM,ij}$;
**IF** $\boldsymbol{S}^{\#}_{LCM,ij}$ meets feasible criteria and process constraints, **THEN**
By 1-unit cycle scheduling, obtain the optimal activity sequence of $\boldsymbol{S}^{\#}_{LCM,ij}$, denoted as $\widehat{\boldsymbol{S}}^{\#}_{LCM,ij}$, and the optimal permutation is denoted as $\hat{p}^{\#}_{ij}$;
**END**
**END Loop1**
**Result:**
Select the optimistic result $\widehat{\boldsymbol{S}}^{\#}_{LCM,ij}$ with the shortest cycle time $t_{\widehat{\boldsymbol{S}}^{\#}_{LCM}}$.

## IV. IMPLEMENTATION SCENARIO

Fig. 5 is a pictorial diagram of a common 6-workstation robotic cell in real semiconductor fabrication. Here only the robot-center operations in the cell are taken into account, by removing the external equipment on the product line and related logistic signals for safety and communication.

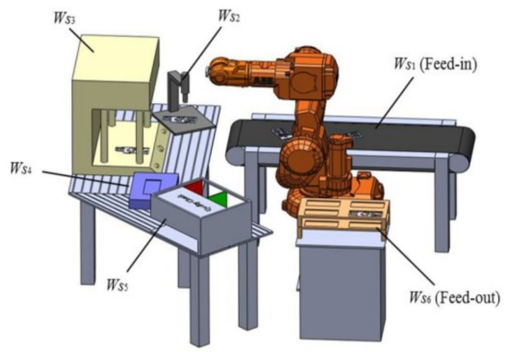


Fig. 5 Pictorial diagram of a 6-workstation robotic cell

TABLE IV
PROPERTIES OF WORKSTATIONS OF A 6-WORKSTATION ROBOTIC CELL

| | Role | Parallel | ParallelTo | Stage | RobotFree | AllowTimeSlot |
|---|---|---|---|---|---|---|
| $Ws_1$ | FI | N | - | 1 | Y | - |
| $Ws_2$ | WS | N | - | 2 | Y | 10 sec |
| $Ws_3$ | WS | N | - | 3 | Y | - |
| $Ws_4$ | WS | N | - | 4 | Y | - |
| $Ws_5$ | WS | N | - | 5 | Y | - |
| $Ws_6$ | FO | N | - | 6 | Y | - |

TABLE V
TIME MATRIX OF PROCESSING, LOADING AND UNLOADING, UNIT: SEC

| | Processing time | Loading time | Unloading time |
|---|---|---|---|
| $Ws_1$ | 0.00 | 0.00 | 0.10 |
| $Ws_2$ | 6.00 | 0.50 | 0.35 |
| $Ws_3$ | 16.00 | 0.70 | 0.12 |
| $Ws_4$ | 5.00 | 0.45 | 0.30 |
| $Ws_5$ | 4.00 | 0.80 | 0.13 |
| $Ws_6$ | 0.00 | 0.30 | 0.00 |

TABLE VI
TIME MATRIX OF TRAVELING, UNIT: SECONDS

| | $Ws_1$ | $Ws_2$ | $Ws_3$ | $Ws_4$ | $Ws_5$ | $Ws_6$ |
|---|---|---|---|---|---|---|
| $Ws_1$ | 0.00 | 1.00 | 1.50 | 0.83 | 1.52 | 0.90 |
| $Ws_2$ | 0.90 | 0.00 | 0.30 | 0.50 | 0.50 | 0.80 |
| $Ws_3$ | 1.50 | 0.30 | 0.00 | 1.50 | 0.80 | 1.10 |
| $Ws_4$ | 1.02 | 0.35 | 0.45 | 0.00 | 0.60 | 2.20 |
| $Ws_5$ | 1.50 | NA | 0.90 | 1.60 | 0.00 | 1.70 |
| $Ws_6$ | 0.80 | 0.85 | 1.10 | 2.30 | 1.90 | 0.00 |

The standard produce flow is strictly followed the base permutation of $\boldsymbol{S}$=($A_2$, $A_3$, $A_4$, $A_5$, $A_6$). The robot takes the part from the conveyor ($Ws_1$) to $Ws_2$ for an adhesive assembly, and then loads it into a fixture on Ws3 for forming with a pressure. After deburring on $Ws_4$, the workpiece is sent to $Ws_5$ for inspection. Finally the finished workpiece is palletized into a tray, namely $Ws_6$. Table IV summarizes the workstation properties. The adhesive processing is implemented on $Ws_2$, and a short time window of 10s is allowed for the part to be loaded to $Ws_3$ before the glue becomes failure. All of the processing would be operated fully by machines, without robot occupation.

Table V and Table VI list the time matrices of processing, travelling, loading and unloading of the cell. Note that the travel from $Ws_5$ to $Ws_2$ is not available due to the physical constraints. Obviously, the cycle time of $T_s$=42.85s to produce a part can be calculated, by following the base permutation.

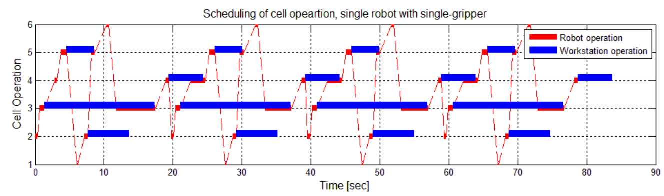


Fig. 6 Gantt chart of robot move with single-gripper

According to the existing work, it is concluded that the best performance with a cycle time of 19.77s can be achieved with a single-gripper. Fig. 6 indicates the optimal scheduling in four elementary cycles for this engineering case. The robot fully utilizes the processing time on $Ws_3$ with intervening activities on $Ws_2$ and $Ws_4$. Once a dual-gripper is selected to replace the single-gripper in this case, the cycle time can be reduced to 16.39s with an assumption of 0.2s for the gripper reposition. Fig. 7 gives an indication from cold start to cyclic production.

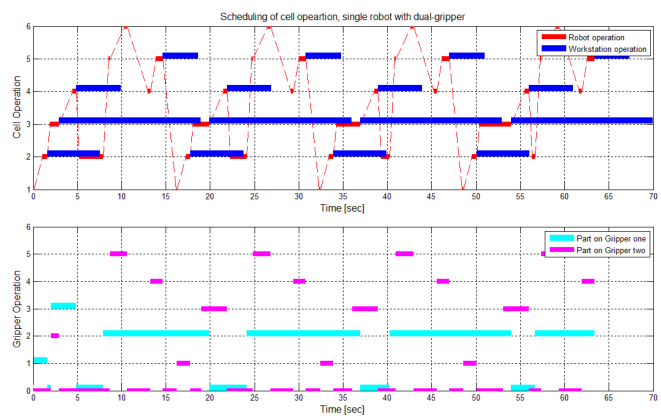


Fig. 7 Gantt chart of robot move with dual-gripper

TABLE VII
PROPERTIES OF WORKSTATIONS OF A 6-WORKSTATION ROBOTIC CELL

| | Role | Parallel | ParallelTo | Stage | RobotFree | AllowTimeSlot |
|---|---|---|---|---|---|---|
| $Ws_1$ | FI | N | - | 1 | Y | - |
| $Ws_2$ | WS | N | - | 2 | Y | 10 sec |
| $Ws_3$ | WS | N | - | 3 | Y | - |
| $Ws_4$ | WS | N | - | 4 | Y | - |
| $Ws_5$ | WS | N | - | 5 | Y | - |
| $Ws_6$ | FO | N | - | 6 | Y | - |
| $Ws_7$ | WS | Y | 3 | - | Y | - |

Apparently, $Ws_3$ blocks the further improvement of productivity, either in single gripper or dual gripper solution, because of its significantly long processing time. Obviously, the reduction in processing time on $Ws_3$ would be the most effective way. Nevertheless, the processing time of a specific process is hard to be changed after the process solution is fixed. Adding an extra workstation $Ws_7$, parallel to $Ws_3$, as shown in Fig. 8, provides an alternative. The new solution has the completely same function and properties as $Ws_3$. Thus the properties of workstations in Table IV, as well as time matrices in Table V and Table VI would be re-defined as following Tables.

TABLE VIII

TIME MATRIX OF PROCESSING, LOADING AND UNLOADING, UNIT: SECONDS

| | Processing time | Loading time | Unloading time |
|---|---|---|---|
| $Ws_1$ | 0.00 | 0.00 | 0.10 |
| $Ws_2$ | 6.00 | 0.50 | 0.35 |
| $Ws_3$ | 16.00 | 0.70 | 0.12 |
| $Ws_4$ | 5.00 | 0.45 | 0.30 |
| $Ws_5$ | 4.00 | 0.80 | 0.13 |
| $Ws_6$ | 0.00 | 0.30 | 0.00 |
| $Ws_7$ | 16.00 | 0.70 | 0.12 |

TABLE IX

TIME MATRIX OF TRAVELING, UNIT: SECONDS

| | $Ws_1$ | $Ws_2$ | $Ws_3$ | $Ws_4$ | $Ws_5$ | $Ws_6$ | $Ws_7$ |
|---|---|---|---|---|---|---|---|
| $Ws_1$ | 0.00 | 1.00 | 1.50 | 0.83 | 1.52 | 0.90 | 1.10 |
| $Ws_2$ | 0.90 | 0.00 | 0.30 | 0.50 | 0.50 | 0.80 | 0.30 |
| $Ws_3$ | 1.50 | 0.30 | 0.00 | 1.50 | 0.80 | 1.10 | 1.90 |
| $Ws_4$ | 1.02 | 0.35 | 0.45 | 0.00 | 0.60 | 2.20 | 1.60 |
| $Ws_5$ | 1.50 | NA | 0.90 | 1.60 | 0.00 | 1.70 | 0.90 |
| $Ws_6$ | 0.80 | 0.85 | 1.10 | 2.30 | 1.90 | 0.00 | 1.10 |
| $Ws_7$ | 1.20 | 0.32 | 0.00 | 1.50 | 0.90 | 1.30 | 1.60 |

According to the previous definition, the unique LCM cycle is composed of two blocked cycles. It can be expressed as:

$$\boldsymbol{S}_{\mathrm{LCM}}=(\boldsymbol{S}_{\mathrm{BLK},1}, \boldsymbol{S}_{\mathrm{BLK},2})$$

where $\boldsymbol{S}_{\mathrm{BLK},1}=(A_2, A_3, A_4, A_5, A_6)$ and $\boldsymbol{S}_{\mathrm{BLK},2}=(B_2, B_7, B_4, B_5, B_6)$. Note that $B_7$ here means the action sequence from moving to $Ws_2$ and pick a part from it and then moving to $Ws_7$.

The optimal sequences of each of them can be obtained:

$$\hat{S}_{LCM} = (\hat{S}_{BLK,1}, \hat{S}_{BLK,2})$$

where,

$$\hat{S}_{BLK,1} = (A_3, A_5, A_2, A_6, A_4)\ , \hat{S}_{BLK,2} = (B_7, B_5, B_2, B_6, B_4).$$

Appling # operation to above $\hat{S}_{LCM}$, it is easy to obtain the sequence $\hat{S}^{\#}_{LCM,ij} = (A_3, A_5, A_2, B_4, A_6, B_7, B_5, B_2, B_6, A_4)$. By means of this improved 2-unit cycle, it manages to produce every two workpiece within 25.4s. As a result, the cycle time of one workpeice production can be regarded as 12.7s. Fig. 8 is a pictorial result of the Gantt chart of this robotic cell. The parallel workstation $Ws_7$ effectively alleviates the dependence on $Ws_3$.

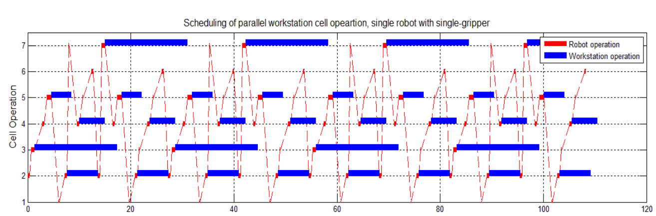


Fig. 8 Gantt chart of robot move with single-gripper in a parallel workstation robotic cell

Compared to the original cell solution with single-gripper, the solution with parallel workstation brings about 36% throughputs. Even compared to the dual-gripper solution, it would also produce more workpieces in one unit time. However, the cost of the whole cell would be increased because of the extra workstation. Since the equipment of a process machine is normally expensive than a dual-gripper, there is a tradeoff between the resultant increase in revenue with the additional equipment costs. The resultant diagram provides a quantitative overview of the scheduling, and actually an insight to the links between throughput and cell flexibility.

TABLE X

COMPARISON OF CYCLE TIME PERFORMANCE OF EACH SOLUTION

| Solutions | Cycle time [s] | Improvement [%] |
|---|---|---|
| Single-gripper with serial workstation | 19.77 | - |
| Dual-gripper with serial workstation | 16.39 | 17.1 |
| Single-gripper with parallel workstation | 12.70 | 36.0 |

## V. CONCLUSIONS

In this paper, the scheduling problem in an $n$-workstation $m$-stage robotic cell is discussed. A new method by means of optimally arranging each of blocked cycles in each LCM cycle, and assembling them in a proper order is proposed. It fully takes the advantages of 1-unit cycle and simplified the process in multi-unit cycle optimization problem. The effect of this proposed method is demonstrated by solving a scheduling problem in an $n$=7 workstation robotic cell. Normally, this could cover a great portion of robot-centered applications in engineering practice, especially in 3C industry. The parallel workstation solution is shown to have more productivity rate than dual-gripper solution, but with a higher cost due to the additional equipment. The resultant diagram provides a quantitative overview of the links between throughput and cell pattern, by comparing the resulting increase in revenue with the additional equipment costs.

In the future work, the scheduling method is planned to expand to the scheduling problem in dual-gripper parallel workstation robotic cells. The problem is not only to find the optimal robot move sequence, but also to jointly maximizing the throughput by deciding on the cell solution, including cell pattern, cell layout and process flexibility.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Hui Zhang from ABB Engineering (Shanghai) for his helpful comments and suggestions.

## REFERENCES

[1] B. T. Han, J. S. Cook, "An efficient heuristic for robot acquisition and cell formation," Annual Operation Research, Vol.77, pp.229-252, 1998

[2] V. Mata, A. Tubaileh, "The machine layout in robot cells," International Journal of Production Research, Vol.36, No.5, pp.1273-1292, 1998

[3] S. P. Sethi, C. Sriskandarajah, G. Sorger, J. Blazewicz, W. Kubiak, "Sequence of parts and robot moves in a robotic cell," International Journal of Flexible Services and Manufacturing, Vol.4, No.3-4, pp.331-358, 1992

[4] M. Dawande, H. N. Geismar, S. P. Sethi, C. Sriskandarajah, "Sequencing and scheduling in robotic cells: Recent developments," Journal of Scheduling, Vol.8, pp.387-426, 2005
[5] N. G. Hall, C. Sriskandarajah, "A survey of machine scheduling problems with blocking and no-wait in process," Operation Research, Vol.44, pp510-525, 1996
[6] H. Chen, C. Chu, J. Proth, "Cyclic scheduling with time window constraints," IEEE Transactions on Robotics and Automation, Vol.14, pp.144-152, 1998
[7] H. Gultekin, M. S. Akturk, O. E. Karasan, "Cyclic scheduling of a 2-machine robotic cell with tooling constraints," European Journal of Operational Research, Vol.174, No.2, pp.777-796, 2006
[8] H. Gultekin, M. S. Akturk, O. E. Karasan, "Scheduling in robotic cells: process flexibility and cell layout," International Journal of Production Research, Vol.46, No.8, pp.2105-2121, 2008
[9] M. S. Akturk, H. Gultekin, O. E. Karasan, "Robotic cell scheduling with operational flexibility," Discrete Applied Mathematics, Vol.145, pp.334-348, 2005
[10] P. Brucker,; T. Kampmeyer, "A general model for cyclic machine scheduling problems," Discrete Applied Mathematics, Vol. 156, No.13, pp.2561-2572, 2008
[11] N. Brauner, G. Finke, W. Kubiak, "Complexity of one-cycle robotic flow shops," Journal of Scheduling, Vol.6, No.4, pp.355-371, 2003
[12] J. F. Zhang, X. Y. Fang, "Robot move scheduling optimization for maximizing cell throughput with constraints in real-life engineering," IEEE International Conference on Robotics and Biomimetics, Shenzhen, China, Dec. 12-14, 2013, pp.221-227
[13] H. N. Geismar, M. W. Dawande, C. Sriskandarajah, "Robotic cells with parallel machines: Throughput maximization in constant travel time cells," Journal of Scheduling, Vol.7, No.5, pop.375-395, 2004
[14] H. N. Geismar, C. Sriskandarajah, N. Ramanan, " Increasing throughput for robotic cells with parallel machines and multiple robots," IEEE Transactions on Automation Science and Engineering, Vol.1, No.1, pp.84-89, 2004
[15] M. W. Dawande, H. N. Geismar, S. P. Sethi, C. Sriskandarajah. Throughput optimization in robotic cells. New York: Springer, 2007
[16] M. F. Aly, A. T. Abbas, S. M. Megahed, "Robot workspace estimation and base placement optimisation techniques for the conversion of conventional work cells into autonomous flexible manufacturing systems," International Journal of Computer Integrated Manufacturing, Vol.23, No.12, pp.1133-1148, 2010
[17] T. C. Lueth, "Automated Planning of Robot Workcell Layouts," Proceedings of IEEE International Conference on Robotics and Automation, pp.1103-1108, Nice, France, May 12-14, 1992
[18] B. Nelson, M. Donath, "Optimizing the location of assembly tasks in manipulator's workspace," Journal of Robotic Systems, Vol.7, No.6, pp.791-811, 1990
[19] J. F. Zhang, X. Y. Fang, "Response surface method based robotic cells layout optimization in small part assembly", Proceedings of The 44th International Symposium on Robotics, Soul, Korea, Oct.25-28, 2013
[20] P. T. Zacharia, N. A. Aspragathos, "Optimal robot task scheduling based on genetic algorithms," Robotics and Computer-Integrated Manufacturing, Vol.21, pp.67-79, 2005