Original article

Institution of MECHANICAL ENGINEERS

Journal of PROCESS MECHANICAL ENGINEERING

Proc IMechE Part E: J Process Mechanical Engineering
0(0) 1–12

DOI: 10.1177/0954408917733850
journals.sagepub.com/home/jpme
SAGE

Scheduling method of robotic cells with machine–robot process and time window constraints

Bing-hai Zhou and Ming Li

Abstract

Robotic cells are widely used in the fields of incorporate automation and repetitive processing. Throughput analysis and scheduling of robotic cells with machine process have been well studied, while research work on robotic cells with machine–robot process in which robots concurrently perform collaborated tasks in addition to part transportation is still at an early stage. After defining the concepts of internal and external time window constraints, we propose a robotic cell scheduling problem with machine process and machine–robot process simultaneously, when processing time window is an essential constraint. According to the constraints in real engineering practice, a mathematic model of single-gripper robotic cells is established to minimize the average manufacturing cycle time. A shifting bottleneck searching algorithm is proposed based on the basis of analyses. In addition, a lower bound for the average manufacturing cycle time is established. Finally, through extensive simulation experiments, the numerical and experimental results demonstrate that the shifting bottleneck searching provides optimal or near optimal solutions.

Keywords

Machine–robot processing, robotic cells, time window constraints, scheduling, algorithm

Date received: 1 March 2017; accepted: 29 August 2017

Introduction

Robotic cell has been regarded as one of the most cost-effective standard tools in terms of repeatability, flexibility, accuracy, and auxiliary for industrial manufacturers. A typical robotic cell as depicted in Figure 1 normally consists of an input buffer M_0 that contains the unprocessed parts, a series of process machines $M_1, M_2, \ldots, M_H$, an output buffer M_{H+1} for completed parts and a robot that is used to transport parts successively from the M_0 through $M_1, M_2, \ldots,$ M_H, and finally into the M_{H+1}. All machines except for input and output buffering devices can process at most one part at any time. Therefore, each part J_n must be either on one of the machines, in the input buffer, output buffer, or transported by the robot.

In a robotic cell system, even a minor improvement in throughput would bring enormous economic benefit for industrial manufacturers. Geismar et al.[1] illustrated that an increase of 7.49% achieved by simply changing the sequence of robot moves could increase revenue by $2,396,800/week in industry. However, the robotic cells scheduling problem (RCSP) concerning how to obtain the optimum sequence of robot activities is a non-deterministic polynomial (NP)-hard problem.[2] The robotic cell is a complex combination of the traditional flow shop and job shop production systems. Yi et al.[3] pointed out the specificity and complexity of the scheduling of robotic cells and distinguished the differences between robotic cells and flow shops and job shops, and provided lower bound values using "pull" and "swap" strategies for single-blade and double-blade robots, respectively. Many scholars and engineers have put forward all kinds of methods for optimal solution in last decades. Leung et al.[4] addressed a multi-hoist scheduling problem for finding the minimum-time cycle with CPLEX by formulating it as an MIP model. Alcaide et al.[5] used a critical path approach to solve the same problem with time window constraints. Ding et al.[6] addressed a decision-moving-done method of event-driven simulation for robotic cells with multiple robots, and proposed an event graph-based pruning search algorithm. Kharbeche and Carlier[7] presented a branch-and-bound algorithm which was the first exact

School of Mechanical Engineering, Tongji University, Shanghai, PR China

Corresponding author:
Bing-hai Zhou, School of Mechanical Engineering, Tongji University, No. 4800 Cao'an Highway, Shanghai 201804, PR China.
Email: bhzhou@tongji.edu.cn

procedure specifically designed with regard to this complex RCSP. However, most of these methods are too complicated to be applied to a large-scale problem.

In terms of simplification, a cyclic scheduling in robotic cells is always attractive since it is easy to understand and control in practice. In the cyclic environment, a cycle is specified by a repeatable sequence of robot moves designed to transfer a set of parts between the machines for their processing. Phillips and Unger[8] built the first mixed-integer programming model to minimize the cycle time for one-unit scheduling with only one hoist. Brauner et al.[9] studied the computational complexity one-cycle robotic flow-shops and proved that the problem is NP-hard in the strong sense when the travel times between the machines of the cell are symmetric and satisfy the triangle inequality. Jungyun et al.[10] provided multi-agent-based scheduling methods for hybrid cellular production lines in semiconductor industry. Xidias et al.[11] studied the cyclic schedule for two robotic manipulators operating in a three-dimensional environment. Yoon[12] proposed a scheduling strategy for deadlock avoidance operation in robotic manufacturing cells. Fathian et al.[13] applied metaheuristic algorithms for two-machine robotic cells output rate analysis in cyclic environment. Crama et al.[14] surveyed cyclic scheduling problems in robotic flowshops, models for such problems, and the complexity of solving these problems. Experts and practitioners[8–14] have extensively researched one-unit cycle schedules. A necessary framework for the cyclic production system was constructed,[15] and it showed that there were $n!$ one-unit cycles in an n-machine robotic cell.[16] Che et al.[17] and Chu[18] have pointed out that multi-unit cycle schedules may achieve better throughput than the optimal 1-unit cycle schedule. The presented paper here addresses a cyclic scheduling problem as well.

In the cyclic scheduling problem, achieving maximum or near maximum throughput of robotic cells is one of the highlighted objectives. The throughput of these systems largely depends on the combination optimization of crucial resources, i.e. machines and robots. According to the situation of the resource occupied when one part is processed, the process for a part can be classified into three types:

1. Machine process (MP). When one part is processed, only the machine is occupied.
2. Robot process (RP). When one part is processed, only the robot is occupied.
3. Machine–robot process (MRP). When one part is processed, both the machine and the robot are occupied.

Most existing scheduling methods, no matter in single-gripper robots[19–23] or in dual-gripper robots,[24–26] focus on either MP, a typical traditional process, where the machine processes one part individually, or RP where robots are used only for material handling. However, few studies pay attention to MRP in which robots concurrently perform collaborated tasks in addition to part transportation. MRP is common in real engineering practice such as chemical industry and fiber-optics fabrication, where the robots concurrently perform collaborated tasks (e.g. accurately readjust location, measure size and add reagent) in addition to part transportation. As an instance of MRP, the application of Grip-Gage-Go (GGG) robot, which performs in-process quality control as its additional task, has been popular in manufacturing cells recently. The auxiliary, equipped at the end of the robot arm, performs accurately diameter measurement while the part is in the machine. The presented paper here focuses on an MRP scheduling problem.

Besides, MRP is seriously influenced by important timing constraints of robotic cells, namely time window constraints, which limit the time interval that a part spends in a process machine. Equivalently, the current processing time of a part on a machine cannot be shorter than the lower bound (LB) or longer than the upper bound of the time window. Time window constraints appear in processes like rapid thermal processing and chemical vapour deposition in semiconductor fabrication, where leaving a part in a process machine longer than its processing time can be detrimental to that part's desired qualities. Time window constraints aggravate the conflict of resources, and thus increasing the complexity of robotic cell scheduling problems. Yoon and Yong[27] first proposed a two-stage algorithm to solve the problem of on-line scheduling of robotic cells with post-processing time window constraints. Alcaide et al.[5] developed a graph model of production processes, making it possible to apply PERT-CPM solution techniques for multi-robot cells scheduling with time-window constraints. Yan et al.[28] studied a robotic cell with processing time windows, and presented a branch and bound algorithm for optimal cyclic schedules to minimize the cycle time. Zhou and Li[29] proposed a try and error-based scheduling algorithm for cluster tools with time window constraints and set minimizing the makespan as the objective. Liu and Zhou[30] presented a time-set constraints-based heuristic algorithm, which successfully solved the scheduling problem of multi-cluster tools with time window constraints.

All of the papers aforementioned only focused on time window constraints without MRP, referred to external time window constraint (ETWC). When it comes to MRP, the authors have to focus on a new type of time window constraints due to waiting for the robot to activate MRP. As the waiting time constraint occurs during the machining process, the authors call it internal time window constraint (ITWC), which imposes a limit on the time that a part waits for the robot to start MRP. If the waiting time is longer than the internal residency constraint time, it would lead to quality problems. As an example of ITWC, parts in PCB (printed circus board) fabrication are processed with photoetching on both sides before developing etch striping (DES). The required photoetching time is 10 s but no more than 15 s on each side. Therefore, the part must be turn over by robot, which is regarded as an MRP, within 5 s after finishing one-side photoetching, otherwise discarded.

The contribution of this paper is to propose a scheduling method for robotic cells with MRP and ITWCs and ETWCs. The remainder of this paper is organized as follows. Problems are more rigorously described and a mathematical model is established in the next section. Then the lower bound (LB) and a shifting bottleneck searching (SBS) algorithm are present. Some simulation experiments are presented in "Computational evaluation" section. Finally, the paper is concluded with "Conclusions and future" section.

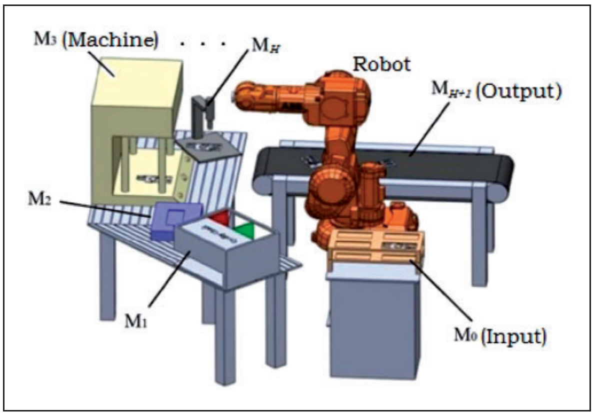

Figure 1. General diagram of robotic cells.

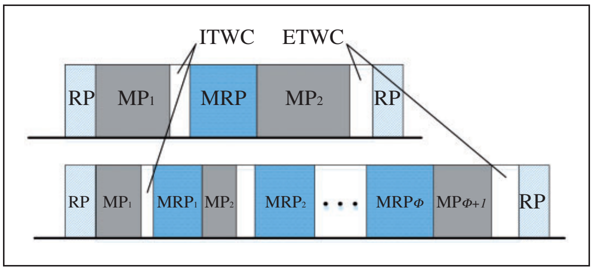

Figure 2. Illustration for ETWC and ITWC.
ETWC: external time window constraint; ITWC: internal time window constraint.

Problem description and formulation

Problem description

As mentioned in the above section, the definitions of ETWC and ITWC are provided for clarity.

Definition 1. ETWCs are the upper bound limit on the time that a part resides in the machine after its process has completed. Correspondingly, the ITWCs impose an upper-bound limit on the time that a part waits for the robot to start MRP stage. The ITWCs and ETWCs are illustrated in Figure 2 (MP_i denotes the i th MP in Machine, so does MRP_i).

Robotic cells with MRP and time window constraints have fixed configurations as Figure 1, all machines share one single-gripper robot, which is in charge of loading the part, travelling among machines, unloading the part, and either fully or partially waiting for the part to complete its process. It is assumed that the time for a robot in transporting, loading and unloading is always deterministic and accurate.

For the cell configuration, robot moves, and processing time, basic assumptions are stated as follows:

(1) A single-gripper robot can pick/place only one part $J_n(\mathrm{n} = 1, 2, \ldots, N)$ at a time. The robot takes the same time to pick and place a part. Transporting time is constant among different machines.
(2) Each machine can process only one part at a time.
(3) A part is permitted to be unloaded from machine only when its process is finished and the next machine is empty.
(4) The processing time and flow patterns (i.e. $\{M_0 \to M_1 \to \cdots \to M_M \to M_{H+1}\}$) of parts are the same, while the processing time is assumed to be longer than transporting time.
(5) There exists MRP during the machining of a single part. In addition, MRP_i would start after MP_i is finished.
(6) There are ITWCs and ETWCs on machines, if the actual residency time of a part exceeds the time window constraints, the part will be unqualified.
(7) Once a part J_n is loaded on the machine M_i, the process will be activated immediately.
(8) After a part J_n is unloaded from the machine M_i, it will be moved to the machine M_{i+1} immediately, in case of quality problems.

Problem formulation

To formulate the scheduling problem, based on the description of Figures 1 and 2, following detailed notations are defined. The notations in Table 1 are based on the descriptions above and all the notations are composed of their main parts and corresponding subscripts.

Our study aims to find an optimal schedule satisfying MRP and ITWCs and ETWCs for robotic cells. In other words, based on the assumptions above, the authors are trying to sequence all the robot activities to minimize the average manufacturing cycle time, or equivalently maximize the throughput, under MRP and time window constraints.

Table 1. Notations based on the descriptions above consist of their main parts and corresponding subscripts.

Notation	Definition
J_n	PART to be scheduled, n is the part number
M_i	The i th machine of the robotic cells
R	Robot of the robotic cells
N	Total number of parts
H	Total number of machines
$P^M_{i,j}$	The j th MP time of J_n in M_i
$\psi(i)$	The times of MP in M_i
P^M_i	The total MP time in M_i, $P^M_i = P^M_{i,1} + P^M_{i,2} + \cdots + P^M_{i,j} + \cdots + P^M_{i,\psi(i)}$
$P^{MR}_{i,j}$	The j th MRP time of J_n in M_i
$\phi(i)$	The times of MRP in M_i
P^{MR}_i	The total MRP time in M_i, $P^{MR}_i = P^{MR}_{i,1} + P^{MR}_{i,2} + \cdots + P^{MR}_{i,j} + \cdots + P^{MR}_{i,\phi(i)}$
$W_{i,j}$	The j th ITWC time of J_n in M_i. It is an upper time limit that the part starts MRP after finishing the j th MP.
W	ITWC time set of J_n in M_i, $W = \{W_{i,1}, W_{i,2}, \cdots, W_{i,j}, \cdots, W_{i,\phi(i)}\}$
U_i	ETWC time of J_n in M_i. It is an upper time limit that the part can reside in M_i after being processed
t_l	Loading time of a part. It is a constant
t_u	Unloading time of a part. It is a constant
t_m	Moving time of the robot between two machines. It is a constant
t_M	Total transporting time of a part, $t_M = t_l + t_u + t_m$
$S_{i,n}$	Time point that J_n enters M_i
$L_{i,n}$	Time point that J_n leaves M_i
$S^M_{i,n,j}$	Time point that J_n starts the j th MP in M_i
$L^M_{i,n,j}$	Time point that J_n finishes the j th MP in M_i
$S^{MR}_{i,n,j}$	Time point that J_n starts the j th MRP in M_i
$L^{MR}_{i,n,j}$	Time point that J_n finishes the j th MRP in M_i
$f_{i,n}$	Actual time that J_n resides in M_i
$d^{in}_{i,n,j}$	The j th actual internal residency time that J_n resides in M_i
$d^{ex}_{i,n}$	Actual external residency time that J_n resides in M_i
TR	Set of the available time intervals of the robot, namely, $TR = \{t_{R,1}, t_{R,2}, \cdots, t_{R,\varphi t(i,n)}\}$, $\varphi t(i,n)$ is the number of time intervals. During the time intervals included in TR, R is available to load, unload, transport and collaborated process for J_n
C_T	Average manufacturing cycle time

On the basis of the assumptions above, the mathematic model is built as follows:

According to the assumption (1), the absolute difference value between the entering time and leaving time of a random part in a random machine must be longer than the time that the robot finishes transporting a part once. Therefore, the authors can obtain relations (1) to (3) as follows

$$\left|S_{i_1,n_1} - S_{i_2,n_2}\right| \geqslant t_M \tag{1}$$

$$\left|L_{i_1,n_1} - L_{i_2,n_2}\right| \geqslant t_M \tag{2}$$

$$\left|S_{i_1,n_1} - L_{i_2,n_2}\right| \geqslant t_M \tag{3}$$

where $i_1 \neq i_2; n_1 \neq n_2,\ i_1, i_2 \in [1, H]; n_1, n_2 \in [1, N]$

As RP should be done in some available time intervals of R for J_n, the authors can obtain the following relation

$$\exists q \in \{1, 2, \ldots, \varphi t(i,n)\} : [S_{i,n} - t_M, S_{i,n}] \subseteq TR \tag{4}$$

$$i = 1, 2, \ldots, M \quad n = 1, 2, \ldots, N$$

where q represents the q th time interval in TR

As mentioned in assumption (2), it is impossible that there is more than one part in a machine simultaneously. The absolute difference value of the entering time and the leaving time between different parts in a random machine must be longer than the processing time. Therefore, the following two relations should be formulated

$$\left|S_{i,n_1} - S_{i,n_2}\right| \geqslant \boldsymbol{P}^{MR}_i + \boldsymbol{P}^M_i \tag{5}$$

$$\left|L_{i,n_1} - L_{i,n_2}\right| \geqslant \boldsymbol{P}^{MR}_i + \boldsymbol{P}^M_i \tag{6}$$

Assumption (5) indicates that there exists MRP for each part, and the MRP should be conducted in some available time intervals of TR. Therefore, the authors can get the following relation

$$\begin{aligned} &\exists k \in \{1, 2, \ldots, \varphi t(i,n)\} : \\ &[S^{MR}_{i,n,j} + d^{in}_{i,n,j}, S^{MR}_{i,n,j} + d^{in}_{i,n,j} + P^M_{i,j}] \subseteq TR \end{aligned} \tag{7}$$

According to the assumption (6), this paper addresses the scheduling problem with ITWCs and ETWCs. The actual internal and external time window time in a machine should be equal to or less than the upper time limit of the time window constraints

$$0 \leqslant d^{in}_{i,n,j} \leqslant W_{i,j} \tag{8}$$

$$0 \leqslant d^{ex}_{i,n} \leqslant U_i \tag{9}$$

It is described in assumption (7) that once a part J_n is loaded on M_i, the machining process will be activated immediately, therefore the following relations are obtained

$$f_{i,n} = P^{MR}_i + P^M_i + d^{ex}_{i,n} + \sum_{j=1}^{\phi(i)} d^{in}_{i,n,j} \tag{10}$$

$$S_{i,n,j}^{MR} = S_{i,n,j}^{M} + P_{i,j}^{M} + d_{i,n,j}^{in} \tag{11}$$

$$S_{i,n,1}^{M} = S_{i,n} \tag{12}$$

In accordance with the assumption (8), actions needed to transport a part are always accurate and successive, so the following relation must be obtained

$$S_{i+1,n} - L_{i,n} = t_M \tag{13}$$

The objective of our study is to minimize the average manufacturing cycle time, or equivalently maximize the throughput.

$$C_T = \lim_{n\to\infty} \frac{S_{H+1,n}}{n} \tag{14}$$

Consequently, the authors have formulated a nonlinear programming problem that takes min C_T as the objective function and relations (1) to (13) as the constraints.

Lower bound

This section establishes an LB for per unit cycle time (equivalently, fundamental period), which is derived from Perkinson et al.[31] In addition, several lemmas, definitions, and theorems are proposed to evaluate the performance of the algorithm in the next section.

Lemma 1. The time point that J_n starts the j th MP in M_i is equal to the time point that J_n finishes the j-1 th MRP in M_i (as shown in Figure 2).

Proof. After a part is loaded on M_i, MP will start first and then MRP. The following is the same. The part needs two resources (robot and machine) in MRP, while one resource in MP. So when MRP transfers to MP, there is no wait time. Namely $S_{i,n,j}^{M} = L_{i,n,j-1}^{MR}$.

Lemma 2. The difference value between the total number of MP and the total number of MRP is equal to H

Proof. While considering MRP and time window constraints simultaneously, a machining process of parts in the machine is divided into $\phi + 1$ MP by ϕ MRP. After a part is loaded on M_i, MP will start first and then MRP (as shown in Figure 2).

$$\text{So } \psi(i) - \phi(i) = 1 \sum_{i=1}^{H} \psi(i) - \sum_{i=1}^{H} \phi(i) = H \quad \square$$

The fundamental period FP_0 of robotic cells without considering MRP is given by

$$FP_0 = \max\{(H+1)t_M, \quad \max_{i\in[1,H]}\{P_i^M\} + 2t_M\} \tag{15}$$

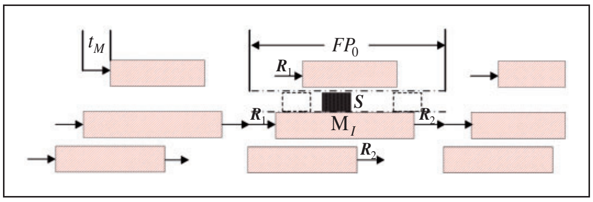

Figure 3. An illustrative chart for fundamental period with MRP.
MRP: machine–robot process.

The bottleneck machine is designated by M_I.

Proof. See Perkinson et al. $\square$

Definition 2. Let d_f denote the shortest delay time in front process of M_I due to MRP, then

$$d_f = \min\{\mathrm{L}'_{\mathrm{I}-2,\,n+1} | [\mathrm{L}'_{\mathrm{I}-2,\,n+1}, \mathrm{L}'_{\mathrm{I}-2,\,n+1} + t_M] \in \eth_{TR} S\} - \mathrm{L}_{\mathrm{I}-2,\,n+\bar{1}} \tag{16}$$

where $\mathrm{L}_{\mathrm{I}-2,\,n+1}$ is the time point that J_{n+1} leaves M_{I-2} without MRP; $\mathrm{L}'_{\mathrm{I}-2,\,n+1}$ is the time point that J_{n+1} leaves M_{I-2} with MRP. S is the set of time intervals of MRP in FP_0 (as shown in Figure 3).

Definition 3. Let d_s denote the shortest delay time in subsequent process of M_I due to MRP,

$$\text{then } d_s = \min\{S'_{I+2,n-1} | [S'_{\mathrm{I}+2,\,n-1} - t_M, S'_{\mathrm{I}+2,\,n-1}] \in \eth_{TR} S\} - \mathrm{S}_{\mathrm{I}+2,\,n-1} \tag{17}$$

where $S_{I+2,\,n-1}$ is the time point that J_{n+1} enters M_{I+2} without MRP; $\mathrm{S}'_{I+2,\,n-1}$ is the time point that J_{n+1} enters M_{I+2} with MRP.

Theorem 1. The LB of robotic cells with MRP is given by

$$LB = \max\{(H+1)t_M + \max_{i\in[1,H]}\{P_i^{MR}\}, \max_{i\in[1,H]}\{P_i^{MR} + P_i^M\} + 2t_M, \max\{FP_A, FP_B\}\} \tag{18}$$

where $FP_A = \max_{i\in[1,I]}\{P_i^{MR} + P_i^M\} + 2t_M + d_f$, $FP_B = \max_{i\in[I,H]}\{P_i^{MR} + P_i^M\} + 2t_M + d_s$

Proof. As shown in Figure 3, $1R$ is the set of time intervals of the robot transport in fundamental period, which includes $H+1$ RP, $\sum_{i=1}^{H}\phi(i)$ MRP and $\sum_{i=1}^{H}\psi(i)$ MP. If $S \cap R = \emptyset$, it indicates that there is no overlap between the MRP and the robot transport, therefore MRP makes no difference in the original fundamental period. So $LB = \max$

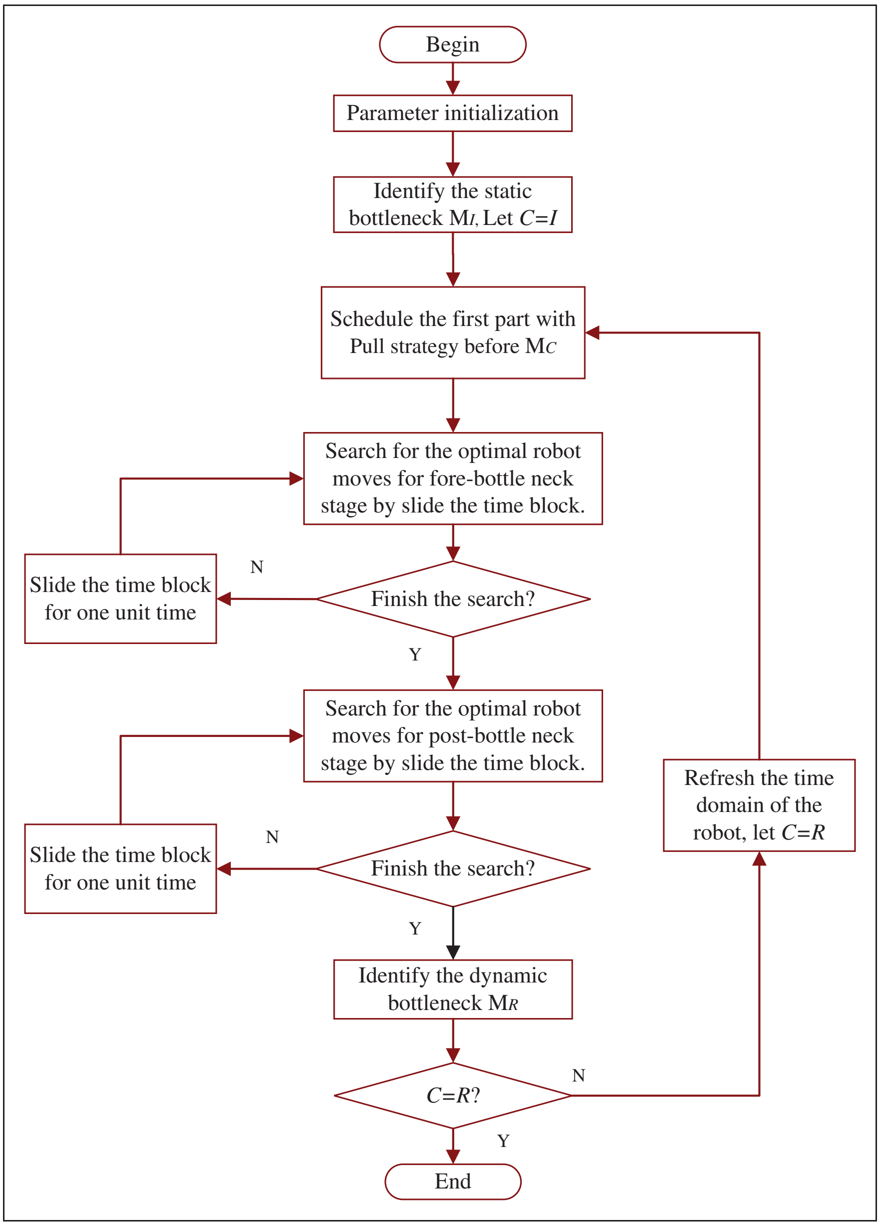

Figure 4. Outline flow diagram of the proposed shifting bottleneck searching algorithm.

$\{(H+1)t_M + \max\limits_{i\in[1,H]}\{P_i^{MR}\}, \quad \max\limits_{i\in[1,H]}\{P_i^{MR}+P_i^M\}+2t_M\}$, when considering MRP. The machine with $\max\limits_{i\in[I,H]}\{P_i^{MR}+P_i^M\}$ is designated by M_I. R_1 is the set of time intervals of the robot transport before a part enters M_I, meanwhile R_2 is the set of time intervals of the robot transport after a part leaves $M_I(R_1, R_2 \subseteq R)$. $LB = FP_A$ in case $S \cap R_1 \neq \emptyset$ and $S \cap R_2 = \emptyset$. In such a situation, MRP has an effect on the process in M_I, which acts as the bottleneck. The starting time of the robot transport that is affected by S will be sure to be postponed. If $S \cap R_1 = \emptyset$ and $S \cap R_2 \neq \emptyset$, it indicates that the process on the second bottleneck after M_I will lag behind for MRP. As a result, $LB = FP_B$. When $S \cap R_1 \neq \emptyset$ and $S \cap R_2 \neq \emptyset$, it is supposed that MRP has effect on both front and subsequent process of M_I, and then $LB = \max\{FP_A, FP_B\}$. □

SBS algorithm

According to the mathematical model formulated in 'Problem description and formulation' section, the SBS algorithm based on time constraint sets is presented for solving the scheduling problem of robotic cells with MRP and time window constraints. An outline flow diagram of the proposed algorithm is given in Figure 4.

Definition 4. According to processing time and actual residency time, the authors define two bottleneck types:

(1) Static bottleneck is M_I which has the longest required processing time, i.e. M_I subjects to $P_I^{MR}+P_I^M = \max\limits_{i\in[1,H]}\{P_i^{MR}+P_i^M\}$

(2) Dynamic bottleneck is M_R which has the longest actual processing time, i.e. M_R subjects to

$$P_R^{MR} + P_R^M + \sum_{j=1}^{\phi(R)} d_{R,n,j}^{in} + \sum_{j=1}^{\phi(R)} d_{R,n}^{ex} = \max_{i\in[1,H]} \left\{ P_i^{MR} + P_i^M + \sum_{j=1}^{\phi(i)} d_{i,n,j}^{in} + \sum_{j=1}^{\phi(i)} d_{i,n}^{ex} \right\}$$

Static bottleneck is easy to be distinguished based on the systematic analysis of manufacturing process data. However, when a certain scheduling solution is applied, the actual time that a part resides in a machine is always not equal to the required processing time due to the complex dynamic characteristics of the robotic cell. It is not straightforward to localize the dynamic bottleneck compared with the static one. Therefore, the dynamic bottleneck is designated as the machine with the maximum required processing time and actual residency time.

At the initial state, the static bottleneck M_I is identified firstly, and then let M_I acts as the current bottleneck M_C, equivalently set $C = I$. Therefore, the cell is divided into two groups by M_I: the fore-bottleneck and the post-bottleneck stages. Two different strategies are undertaken for these two stages. For the fore-bottleneck stages, a pull strategy is adopted, thus the time points of robot loading, unloading at each stage are calculated by backtracking policy. Parts are scheduled one after another according to the flow patterns (i.e. $\{M_0 \to M_1 \to \cdots \to M_M \to M_{H+1}\}$). For the post bottleneck stages, a push strategy is applied. Time-block-sliding method is used as an important way to reduce the residency time, as well as to minimize C_T. After all parts are scheduled, the machine with $\max_{i\in[1,H]}\{P_i^{MR} + P_i^M + \sum_{j=1}^{\phi(i)} d_{i,n,j}^{in} + \sum_{j=1}^{\phi(i)} d_{i,n}^{ex}\}$ is designated as the dynamic bottleneck M_R. If M_C and M_R are the same machine, it indicates that the bottleneck does not change and the scheduling is valid and feasible. Otherwise, let M_R takes the place of the old current bottleneck, and starts a new circulation until bottleneck shifting stops.

The details of steps are as follows

- Given: $P_{i,j}^M, P_{i,j}^{MR}, W_{i,j}, U_i, \quad i = 1, \ldots, H$, $TR = \{[0, +\infty]\}$

Denote M_C as the current bottleneck when scheduling.

Step 1 Identify the static bottleneck M_I, and let $C = I$.

Step 2 Schedule the first part with the pull strategy before M_C.

Set $L_{in} = 0$, $S_{1,1} = L_{in} + t_M$

Calculate the entering time and leaving time of first part in $M_1, \ldots, M_C$

For $i = 1, \ldots, C$ **do**

$S_{i,1,1}^M = S_{i,1}$, $d_{i,1}^{ex} = 0$

For $j = 1, \ldots, \psi(i)$ **do**

$L_{i,1,j}^M = S_{i,1,j}^M + P_{i,j}^M$, $S_{i,1,j}^{MR} = L_{i,1,j}^M$, $L_{i,1,j}^{MR} = S_{i,1,j}^{MR} + P_{i,j}^{MR}$

set $d_{i,1,j}^{in} = 0$

$L_{i,1} = L_{i,1,\psi(i)}^M$, $S_{i+1,1} = L_{i,1} + t_M$

Step 3 Refresh the available time domain of the robot.

For $i = 1, \ldots, C$ **do**

For $j = 1, \ldots, \psi(i)$ **do**

$TR \leftarrow \eth_{TR}[S_{i,1} - t_M, S_{i,1}]$,

$TR \leftarrow \eth_{TR}[S_{i,1,j}^{MR}, S_{i,1,j}^{MR} + P_{i,j}^{MR}]$

($\eth_{TR}[a, b]$ is the complementary set of $[a, b]$ in universal set TR)

Step 4 Scheduling the M_C with no-wait strategy. Calculate the entering time and leaving time of $J_2, \ldots, J_N$ in M_C

For $n = 2, \ldots, N$ **do**

$S_{C,n} = L_{C,n-1} + 2t_M$

For $j = 1, \ldots, \psi(C)$ **do**

$S_{C,n,j}^M = S_{C,n}$, $L_{C,n,j}^M = S_{C,n,j}^M + P_{C,j}^M$, $S_{C,n,j}^{MR} = L_{C,n,j}^M$, $d_{C,n,j}^{in} = 0$, $L_{C,n,j}^{MR} = S_{C,n,j}^{MR} + P_{C,j}^{MR}$,

$L_{C,n} = L_{C,n,\psi(C)}^M$, $d_{C,n}^{ex} = 0$

Step 5 Refresh the available time domain of the robot.

For $n = 2, \ldots, N$ **do**

For $j = 1, \ldots, \psi(C)$ **do**

$TR \leftarrow \eth_{TR}[S_{C,n} - t_M, S_{C,n}]$,

$TR \leftarrow \eth_{TR}[S_{C,n,j}^{MR}, S_{C,n,j}^{MR} + P_{C,j}^{MR}]$

Step 6 Forward searching check external and internal time window time.

For $n = 2, \ldots, N$ **do**

For $i = C-1, C-2, \ldots, 0$ **do**

$L_{i,n} = S_{i+1,n} - t_M$, $L_{i,n,\psi(i)}^M = L_{i,n}$, $d_{i,n}^{ex} = 0$

If $[L_{i,n,\psi(i)}^M - P_{i,\psi(i)}^M - P_{i,\psi(i)}^{MR}, L_{i,n,\psi(i)}^M - P_{i,\psi(i)}^M] \not\subset TR$ **Then**

Slide the time block for one unit time

$L_{i,n,\psi(i)}^M \leftarrow L_{i,n,\psi(i)}^M - 1$, $d_{i,n}^{ex} \leftarrow d_{i,n}^{ex} + 1$

If $d_{i,n}^{ex} > U_i$ **Then**

$S_{C,n} \leftarrow S_{C,n} + 1$ **go to step 4**

For $j = \psi(i), \psi(i) - 1, \ldots, 1$ **do**

$S_{i,n,j}^M = L_{i,n,j}^M - P_{i,j}^M$, $L_{i,n,j-1}^{MR} = S_{i,n,j}^M$, $S_{i,n,j-1}^{MR} = L_{i,n,j-1}^{MR} - P_{i,j-1}^{MR}$, $L_{i,n,j-1}^M = S_{i,n,j-1}^{MR}$, $d_{i,n,j-1}^{in} = 0$

If $[S_{i,n,j-1}^{MR}, L_{i,n,j-1}^{MR}] \not\subset TR$ **Then**

Slide the time block for one unit time

$L_{i,n,j}^M \leftarrow L_{i,n,j}^M - 1$, $d_{i,n,j}^{in} \leftarrow d_{i,n,j}^{in} + 1$

If $d_{i,n,j}^{in} > W_{i,j}$ **Then**

$S_{C,n} \leftarrow S_{C,n} + 1$ **go to step 4**

$S_{i,n} = S_{i,n,1}^M$, $TR \leftarrow \eth_{TR}[S_{i,n} - t_M, S_{i,n}]$, $TR \leftarrow \eth_{TR}[S_{i,n,j}^{MR}, S_{i,n,j}^{MR} + P_{i,j}^{MR}]$

Step 7 Backward searching check external and internal time window time
For $n = 1, \ldots, N$ **do**
For $i = C+1, C+2, \ldots, H$ **do**
$S_{i,n} = L_{i-1,n} + t_M$, $S^M_{i,n,1} = S_{i,n}$
For $j = 1, \ldots, \psi(i) - 1$ **do**

$$L^M_{i,n,j} = S^M_{i,n,j} + P^M_{i,j},\ S^{MR}_{i,n,j} = L^M_{i,n,j},\ d^{in}_{i,n,j} = 0$$

If $[S^{MR}_{i,n,j}, S^{MR}_{i,n,j} + P^{MR}_{i,j}] \not\subset TR$ **Then**
Slide the time block for one unit time
$S^{MR}_{i,n,j} \leftarrow S^{MR}_{i,n,j} + 1$, $d^{in}_{i,n,j} \leftarrow d^{in}_{i,n,j} + 1$

If $d^{in}_{i,n,j} > W_{i,j}$ **Then**
$S_{C,n} \leftarrow S_{C,n} + 1$ **go to step 4**

$$L^{MR}_{i,n,j} = S^{MR}_{i,n,j} + P^{MR}_{i,j},\ S^M_{i,n,j+1} = L^{MR}_{i,n,j}$$

$$L^M_{i,n,\psi(i)} = S^M_{i,n,\psi(i)} + P^M_{i,\psi(i)},\ L_{i,n} = L^M_{i,n,\psi(i)},\ d^{ex}_{i,n} = 0$$

If $[L_{i,n}, L_{i,n} + t_M] \not\subset TR$ **Then**
Slide the time block for one unit time

$L_{i,n} \leftarrow L_{i,n} + 1$, $d^{ex}_{i,n} \leftarrow d^{ex}_{i,n} + 1$
If $d^{ex}_{i,n} > U_i$ **Then**
$S_{C,n} \leftarrow S_{C,n} + 1$ **go to step 4**
$TR \leftarrow \eth_{TR}[S_{i,n} - t_M, S_{i,n}]$,
$TR \leftarrow \eth_{TR}[S^{MR}_{i,n,j}, S^{MR}_{i,n,j} + P^{MR}_{i,j}]$

Step 8 Identify the dynamic bottleneck M_R
If $R \neq C$ **Then** refresh the time domain of the robot, let $C = R$ **go to step 2**
Else, the algorithm ends.

On the basis of detailed steps, the time complexity of the SBS algorithm is $O(H^2N^2\psi(\mathrm{I})^2)$

Computational evaluation

This section describes the problem instances and the experimental details. To evaluate the SBS algorithm, the LB proposed in 'Lower bound' section is used as the basic benchmark. Then the authors compare the performance of the solution approaches experimentally and present insights into the algorithm's performance through analyzing the results comprehensively.

The elongation rate (R) that the SBS algorithm relative to the lower-bound is defined as $R = \frac{C_T - LB}{LB} \times 100\%$. The lower R is, the better the SBS algorithm is. In addition, the improvement rate (R_S) that the SBS algorithm relative to the structural FP_S that the authors propose in this paper below is defined as $R_S = \frac{FP_S - C_T}{FP_S} \times 100\%$. The greater R_S is, the better the algorithm is. Obviously, R and R_S are negatively correlated.

$FP_D = \max\{|(P^M_{i+1} + P^{MR}_{i+1}) - (P^M_i + P^{MR}_i)|\}$ denotes the absolute difference value between the processing time of the adjacent machines; FP_S denotes the structural fundamental period and $FP_S = LB + FP_D$.

Table 2. The original data of the first experiment.

Parameters	Data
Number of machines	8
Transporting time of robots (s)	2
Mean of MP times (s)	40
Standard deviation of MP times (s)	10
Mean of MRP times (s)	10
Standard deviation of MRP times (s)	5
Mean of ETWC times (s)	20
Standard deviation of ETWC times (s)	5
Mean of ITWC times (s)	10
Standard deviation of ITWC times (s)	2
Number of parts	10,20,30,...,200

MRP: machine–robot process; MP machine process; ETWC: external time window constraint; ITWC: internal time window constraint.

The cell factor is calculated as $CF = \frac{max\{P^{MR}_i\} + 2t_M}{max\{P^{MR}_i + P^M_i\} + 2t_M}$ to characterize the robotic cells, which reflects the relationship between the maximum processing time of J_n in a machine and the occupied time of the robot. The smaller CF is, the more vacant the robot is; alternatively, the robot is busier.

The whole simulations consist of four experiments. The authors programmed the algorithm in C++ using Microsoft Visual Studio 2008. The authors did the simulation experiments on an DELL personal computer with 1TB hard disk, 8 GB DDR3 memory and 2.4 GHz Intel Core TM i7 processor 330 M. For each case in the following experiments, the data of instances and experiments are generated according to the guidelines offered by Hall and Posner,[32] which can closely represent real engineering practice, and the result is the mean of 10 instances.

General running time of SBS algorithm

The purpose of this verification experiment is to verify the possibility of applying the SBS algorithm into practice. Take an eight-machine robotic cell as an example, and the related data are in Table 2.

Figure 5 shows that the running time of the SBS algorithm increases along with the number of parts. Obviously, the running time is close to zero when the number of parts is within 40. The running time is less than 80 ms when the number of parts is 200. As far as the result, the algorithm is very suitable for on-line scheduling.

Impact of CF and configuration flexibility

This experiment compares the average manufacturing cycle time of the SBS algorithm with the LB under the impact of CF and configuration flexibility. Data from this experiment are in Table 3. R and R_S were

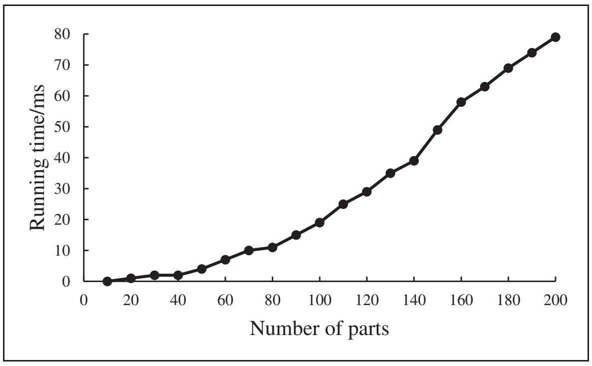

Figure 5. Running time of algorithm.

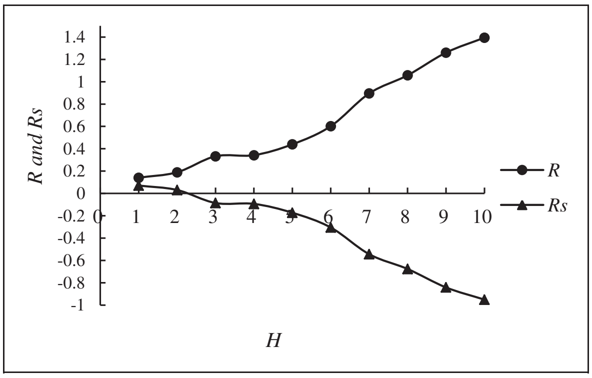

Figure 7. Relationship of R and R_S with H.

Table 3. The original data of the second experiment.

Parameters	Impact of *CF*	Impact of *H*
Number of machines	8	1,2,3,. . .,10
Transporting time of robots (s)	2	2
Mean of MP times (s)	40	40
Standard deviation of MP times (s)	10	10
Mean of MRP times (s)	10	20
Standard deviation of MRP times (s)	5	5
Mean of ETWC times (s)	20	20
Standard deviation of ETWC times (s)	5	5
Mean of ITWC times (s)	10	10
Standard deviation of ITWC times (s)	2	2
Number of parts	100	100
Value of *CF*	1,2,3,. . .,8	

MRP: machine–robot process; MP machine process; : ETWC: external time window constraint; ITWC: internal time window constraint.

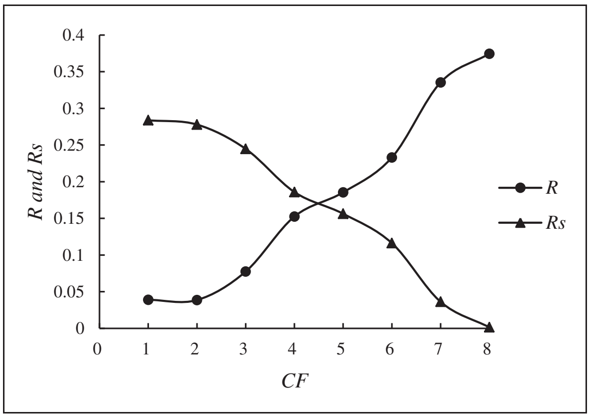

Figure 6. Relationship of R and R_S with *CF*.

Table 4. The original data of the third experiment.

Parameters	Impact of MRP
Number of machines	8
Transporting time of robots (s)	2
Mean of MP times (s)	40
Standard deviation of MP times (s)	10
Mean of MRP times (s)	0,10,20,30
Standard deviation of MRP times (s)	5
Mean of ETWC times (s)	20
Standard deviation of ETWC times (s)	5
Mean of ITWC times (s)	10
Standard deviation of ITWC times (s)	2
Number of parts	100

MRP: machine–robot process; MP machine process; : ETWC: external time window constraint; ITWC: internal time window constraint.

calculated under a variation of *CF* from unity to eight as well as the number of machines from 1 to 10. The corresponding results are shown in Figures 6 and 7.

Figure 6 shows that the impact of *CF* on R and R_S. R generally increases with the increase of *CF*, while R_S generally decreases with the increase of *CF*. It infers that C_T approaches to the LB with the decrease of *CF*. When $CF < 4$, R is less than 0.2, which means the result is pretty good. In practice, the value of *CF* is generally small because the transporting time is small relative to the maximal processing time, so the authors can expect good scheduling results in practice.

Figure 7 shows the impact of configuration flexibility on R and R_S. R generally increases with the increase of H. While R_S presents an opposite result. Both R and R_S suggest that the performance of the algorithm declines along with the increase of the number of machines. Generally, *CF* increases with the increase of the number of machines. Consequently, this experiment verifies the results of the last experiment under the impact of *CF*.

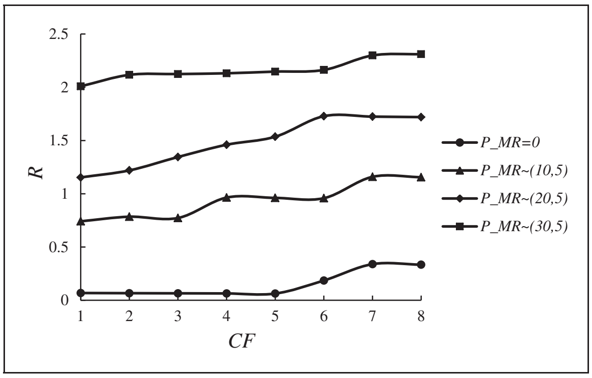

Figure 8. Effect of MRP on scheduling.
MRP: machine–robot process.

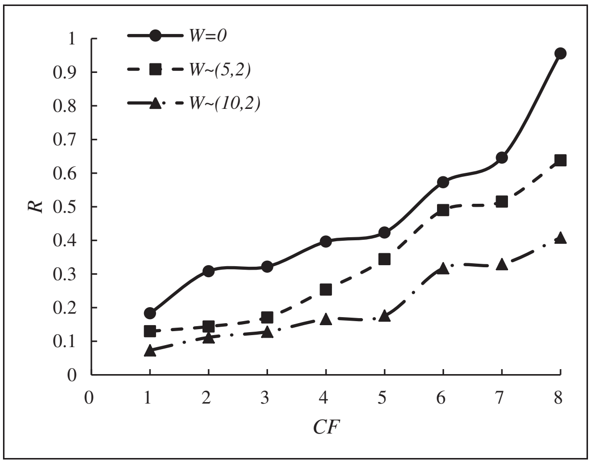

Figure 9. Effect of internal time window constraints on scheduling.

Table 5. The original data of the fourth experiment.

Parameters	Impact of ITWC	Impact of ETWC
Number of machines	8	8
Transporting time of robots (s)	2	2
Mean of MP times (s)	40	40
Standard deviation of MP times (s)	10	10
Mean of MRP times (s)	10	10
Standard deviation of MRP times (s)	5	5
Mean of ETWC times (s)	20	10,20,30
Standard deviation of ETWC times (s)	2	5
Mean of ITWC times (s)	0,5,10	10
Standard deviation of ITWC times (s)	5	5
Number of parts	100	100
Value of *CF*	1,2,3,. . .,8	1,2,3,. . .,8

MRP: machine–robot process; MP machine process; : ETWC: external time window constraint; ITWC: internal time window constraint.

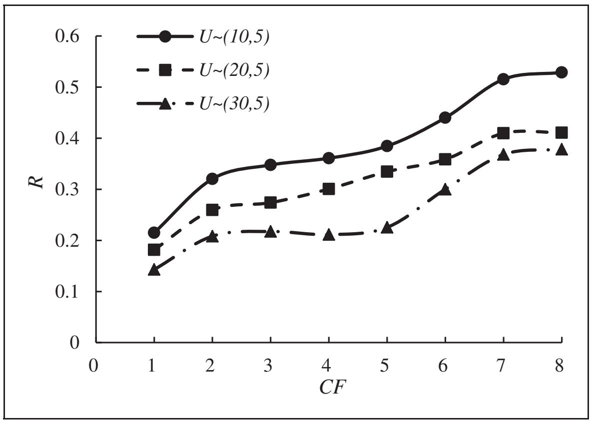

Figure 10. Effect of external time window constraints on scheduling.

Impact of MRP

This experiment compares the average manufacturing cycle time of the SBS algorithm with the LB under the impact of MRP. The authors study the cases of different means of MRP times and different number of machines with MRP. Data from this experiment are shown in Table 4. The variation tendency of *R* with the variation of *CF* is displayed in Figure 8.

Figure 8 shows the relationship of *R* with *CF* under the impact of the dispersion of MRP time. In spite of the different means of processing time, the general variation tendency of the curves is uniform. The results suggest that the algorithm is very adaptive to different means of MRP time. *R* is very close to zero when $P^{MR} = 0$. *R* increases along with the increase of mean of MRP times, while *R* is relatively stable when the value of *CF* is changed. Generally, owing to the process technic and physical structure of robotic cells, the ratios of *MRP time*/*MP time* are low in practice. Therefore, the proposed SBS algorithm is feasible.

Impact of ITWCs and ETWCs

This experiment compares the average manufacturing cycle time with the LB under the impact of time window constraints, namely, the flexibility of the ITWC and ETWC time. Data from this experiment are shown in Table 5. The variation tendency of *R* with the change of *CF* is shown in Figures 9 and 10.

Figure 9 shows the relationship of *R* with *CF* under the variation of the ITWC time. *R* generally decreases with the increase of *CF*. *R* is less than 0.5 when *CF* is smaller than five, which indicates that the algorithm achieves an excellent scheduling result. The general variation tendency of the different curves is uniform in spite of different ITWC time. The results indicate

Table 6. The ANOVA experiments.

Object	Source of variation	Sum of squares	Degrees of freedom	Mean sum of squares	F-ratio	Sig.
ANOVA of Running time	Inter group	156.253	19	8.2238	56.4308	0.00
	Intra group	26.232	180	0.1457		
	Total	182.485	199			
ANOVA of *R*	Inter group	4522.451	27	167.4982	65.9645	0.00
	Intra group	639.883	252	2.5392		
	Total	5162.334	279			

that the algorithm is very adaptive to different ITWC times.

Figure 10 shows the relationship of R with CF under the impact of dispersion of the ETWC time. Though the different curves of the different means of ETWC time are not identical, the general variation tendency of them is uniform. R tends to be steady when CF is greater than six. Obviously, the three curves follow the principle that the more ETWC time the process require, the better the scheduling result the algorithm can achieve.

Statistical analysis

Since one cannot correctly claim that the average of gained outcomes is significantly different with the LB or not, the ANOVA experiments are performed in this phase referred to Teymourian et al.[33] and Kayvanfar et al.,[34] aiming at evaluating whether our simulation results are statistically significant or not. The results of experiments are shown in Table 6.

As it could be observed in Table 6, both of the p-values are zero in ANOVA of running time and R, it indicates our simulation results are statistically significant. As a result, existing differences in the observed values can assure the superiority of the results.

Conclusions and future works

This paper develops the SBS algorithm as a new method for scheduling robotic cells with MRP and internal and ETWCs. MRP is common in real engineering practice such as in chemical industry and in fiber-optics fabrication, whereas most previous researches addressed the cases where the robot cells only have MP or RP. What's more, the current research provides a guidance for issues considering MRP and ITWCs and ETWCs in robotic cells.

The simulation experiments indicate that the algorithm is valid and feasible for the real-time scheduling of robotic cells with MRP and ITWCs and ETWCs. Firstly, the running time of the algorithm is still very short even when 200 parts are scheduled at the same time. Secondly, the algorithm is able to achieve optimal or near optimal scheduling solutions in most cases. R is relatively stable with the increase of CF under the impact of the dispersion of MRP time, which means that the algorithm is adaptive to the variation of CF. The general variation tendency of the different curves is uniform in spite of different ITWC time. The results indicate that the algorithm is very adaptive to different ITWC time. Thirdly, the comparisons with the benchmark under the impact of MRP and internal and external time window time validate the steady and excellent performance of the algorithm. Lastly, an ANOVA analysis is presented to identify the relationships between parameters and objective.

The formulation and heuristic algorithm are mainly focused in this research, and future works will be strived to apply our method in other classes of problems and develop new algorithms for more complicated problem like the scheduling of the multi-robot cells with MRP and machine failure.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by National Natural Science Foundation of China under Grant No. [71471135].

References

1. Geismar HN, Pinedo M and Sriskandarajah C. Robotic cells with parallel machines and multiple dual gripper robots: a comparative overview. *IIE Trans* 2008; 40: 1211–1227.
2. Hall NG, Kamoun H and Sriskandarajah C. Scheduling in robotic cells: complexity and steady state analysis. *Eur J Oper Res* 1998; 109: 43–65.
3. Yi J, Ding S, Song D, et al. Steady-state throughput and scheduling analysis of multicluster tools: a decomposition approach. *IEEE Trans Autom Sci Eng* 2008; 5: 321–336.
4. Leung JMY, Zhang G, Yang X, et al. Optimal cyclic multi-hoist scheduling: a mixed integer programming approach. *Oper Res* 2004; 52: 965–976.

5. Alcaide D, Chu C, Kats V, et al. Cyclic multiple-robot scheduling with time-window constraints using a critical path approach. *Eur J Oper Res* 2007; 177: 147–162.
6. Ding S, Yi J and Zhang MT. Multi-cluster tools scheduling: an integrated event graph and network model approach. *IEEE Trans Semicond Manuf* 2006; 19: 339–351.
7. Kharbeche M and Carlier J. Exact methods for the robotic cell problem. *Flex Serv Manuf J* 2011; 23: 242–261.
8. Phillips LW and Unger PS. Mathematical programming solution of a hoist scheduling program. *IIE Trans* 1976; 8: 219–225.
9. Brauner N, Finke G and Kubiak W. Complexity of one-cycle robotic flow-shops. *J Schedul* 2003; 6: 355–371.
10. Jungyun K, Seong YC and Hyun JY. Multi-agent-based scheduling methods for hybrid cellular production lines in semiconductor industry. *Proc IMechE Part B: J Engineering Manufacture* 2014; 288: 1701–1712.
11. Xidias EK, Zacharia PTh and Aspragathos NA. Time-optimal task scheduling for two robotic manipulators operating in a three-dimensional environment. *Proc IMechE Part I: J Systems and Control Engineering* 2010; 224: 845–855.
12. Yoon HJ. Scheduling for deadlock avoidance operation in robotic manufacturing cells. *Proc IMechE Part B: J Engineering Manufacture* 2010; 224: 329–340.
13. Fathian M, Nakhai I, Heydari M, et al. Applying meta-heuristic algorithms for output rate analysis in two-machine robotic manufacturing cells. *Int J Adv Robotic Syst* 2013; 10: 169–179.
14. Crama Y, Kats V, van de Klundert J, et al. Cyclic scheduling in robotic flowshops. *Ann Oper Res* 2000; 96: 97–124.
15. Sethi SP, Sriskandarajah C, Sorger G, et al. Sequencing of parts and robot moves in a robotic cell. *Int J Flex Manuf Syst* 1992; 4: 331–358.
16. Dawande M, Geismar HN, Sethi SP, et al. Sequencing and scheduling in robotic cells: recent developments. *J Schedul* 2005; 8: 387–426.
17. Che A, Chu C and Chu F. Multicyclic hoist scheduling with constant processing times. *IEEE Trans Rob Autom* 2002; 18: 69–80.
18. Chu C. A faster polynomial algorithm for 2-cyclic robotic scheduling. *J Schedul* 2006; 9: 453–468.
19. Bakar RA and Watada J. Biological computation of optimal task arrangement for a flexible machining cell. *Proc IMechE Part I: J Systems and Control Engineering* 2008; 222: 605–618.
20. Kumar S, Ramanan N and Sriskandarajah C. Minimizing cycle time in large robotic cells. *IIE Trans* 2005; 37: 123–136.
21. Liu MX and Zhou BH. Modelling and scheduling analysis of multi-cluster tools with residency constraints based on time constraint sets. *Int J Prod Res* 2013; 51: 4835–4852.
22. Wang Z and Zhou BH. Bottleneck-based scheduling method of multi-robot cells with residency constraints. *Int J Comput Integr Manuf* 2015; 28: 1237–1251.
23. Zhang J, Fang X and Qi L. LCM cycle based optimal scheduling in robotic cell with parallel workstations. In: *IEEE international conference on robotics and automation*, Piscataway, NJ, USA, 22 September 2014, pp.1367–1373. IEEE.
24. Drobouchevitch IG, Sethi SP and Sriskandarajah C. Scheduling dual gripper robotic cell: one-unit cycles. *Eur J Oper Res* 2006; 171: 598–631.
25. Sethi SP, Sidney JB and Sriskandarajah C. Scheduling in dual gripper robotic cells for productivity gains. *IEEE Trans Robot Autom* 2001; 17: 324–341.
26. Sriskandarajah C and Chandrasekaran R. Scheduling multiple parts in a robotic cell served by a dual-gripper robot. *Oper Res* 2004; 52: 65–82.
27. Yoon HJ and Yong LD. On-line scheduling of robotic cells with post-processing residency constraints. In: *IEEE international conference on systems, man and cybernetics*, Piscataway, NJ, USA, 5–8 October 2003, pp.2785–2790. IEEE.
28. Yan PY, Chu CB, Yang ND, et al. A branch and bound algorithm for optimal cyclic scheduling in a robotic cell with processing time windows. *Int J Prod Res* 2010; 48: 6461–6480.
29. Zhou BH and Li X. Try and error-based scheduling algorithm for cluster tools of wafer fabrications with residency time constraints. *J Cent South Univ* 2012; 19: 187–192.
30. Liu MX and Zhou BH. Scheduling algorithm of multi-cluster tools based on time constraint sets. *Acta Autom Sin* 2012; 38: 479–484.
31. Perkinson TL, Mclarty PK, Gyurcsik RS, et al. Single-wafer cluster tool performance: An analysis of throughput. *IEEE Transactions on Semiconductor Manufacturing* 1994; 7: 369–373.
32. Hall NG and Posner ME. Generating experimental data for computational testing with machine scheduling applications. *Oper Res* 2001; 49: 854–865.
33. Teymourian E, Kayvanfar V, Komaki GM, et al. Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem. *Inform Sci* 2016; 334–335: 354–378.
34. Kayvanfar V and Teymourian E. Hybrid intelligent water drops algorithm to unrelated parallel machines scheduling problem: a just-in-time approach. *Int J Prod Res* 2014; 2: 5857–5879.