International Journal of Production Research
Vol. 00, No. 00, DD Month 200x, 1–17

Timed event graph-based cyclic reconfigurable flow shop modeling and optimization

S. C. Ren, D. Xu, F. Wang, and M. Tan

(2nd revised manuscript prepared using the LaTeX 2ε template available via the journal homepage)

The manufacturing process of a part involves sequential steps and each step could be viewed as the part being manufactured by a process module with some specific function. The module must be placed on a machine and connected with the machine via standard interfaces. The machine considered here is a carrier or general platform that can hold one or several different modules simultaneously. Based on the idea that modules are independent of machines and different combinations of modules and machines result in different configurations, the cyclic reconfigurable flow shop is proposed for the new manufacturing paradigm-Reconfigurable Manufacturing System (RMS). The cyclic reconfigurable flow shop can be modeled as a timed event graph. Different cases of cyclic reconfigurable flow shops are discussed respectively and the optimal configuration can be obtained by solving the corresponding mixed-integer program derived from the timed event graph model.

1 Introduction

In order to stay competitive in current manufacturing environment, companies must respond to the drastic changes of market demands quickly and cost-effectively. Koren *et al.* (1999) have proposed a new manufacturing paradigm that a reconfigurable manufacturing system (RMS) is designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in response to sudden changes in market or in regulatory requirements. Compared with conventional manufacturing systems such as dedicated manufacturing system (DMS) or flexible manufacturing system (FMS), RMS is a dynamic, evolving system and it has been generally acknowledged to be one of the future manufacturing paradigms.

RMS has been extensively studied by many scholars from different aspects. Zhao *et al.* (2000a,b, 2001a,b) have proposed that products required by customers can be classified into several product families, each of which is a set of similar products and each family corresponds to one configuration of the RMS. The products belonging to the same family will be produced by the RMS under the corresponding configuration and three issues related to RMS (i.e. the optimal configurations in the design, the optimal selection policy in the utilization and the performance measure in the improvement) are further discussed. Yigit *et al.* (2002, 2003) has addressed the problem of optimizing modular products in RMS. The problem is posed as a generalized subset selection problem where the best subsets of modules instances of unknown sizes are determined by minimizing an object function that represents a trade-off between the quality loss due to modularization and the cost of reconfiguration while satisfying the problem constrains. Abdi and Labib have employed analytical hierarchical process (AHP) or fuzzy analytical hierarchical process (FAHP) in the design strategy (2003), grouping and selecting products (2004a) and feasibility study of the tactical design justification (2004b) for RMS. Bruccoleri *et al.* (2003) have discussed the issue of exception handling in RMS. Others focus on the aspects of RMS such as reconfigurable machining tools (Landers *et al.* 2001), logic controllers (Park *et al.* 1999, 2001), etc.

In this paper, a kind of cyclic reconfigurable flow shops are proposed for the RMS. This paper is organized as follows. The basic idea that modules are independent of machines and different combinations of modules and machines result in different configurations is explained in section 2. Based on the above idea, a kind of cyclic reconfigurable flow shops are proposed in section 3 and modeled as timed event graphs in section 4. In section 5, mixed-integer programs are derived from the timed event graph models to obtain the optimal configurations. An example is illustrated in section 6 and future work is discussed in section 7.

2 Basic idea

The manufacturing process of a part involves sequential steps and each step could be viewed as the part being manufactured by a *process module* with some specific function. The module must be placed on a *machine* and connected with the machine via standard interfaces. The machine considered here is a carrier or general platform that can hold one or several different modules simultaneously. The machine plays the role of supplying power, communicating, coordinating and controlling different modules, and etc. For example, a robot (machine) can perform the operations of cutting and drilling if it is equipped with the cutting and drilling tools (modules). Other examples are the dedicated machine in DMS and computerized numerically controlled (CNC) machine in FMS. The dedicated machine could be considered as a machine with only one module to perform single functionality, while the CNC machine is a machine with multiple modules to achieve functional flexibility. However, the modules are fixed on the dedicated or CNC machine traditionally. In order to quickly adjust system capacity and functionality to meet market changes, one possible solution would be to make modules independent of machines, that is, modules can be removed from one machine and added to another machine freely.

Different combinations of modules and machines result in different *configurations*. The process of the system changing from one configuration to another is called *reconfigure*. Generally speaking, the system performance differs under different configurations. One goal of the reconfigurable manufacturing systems is to find a reasonable configuration method (i.e. to distribute modules over machines) to achieve the desired system performance.

It's worth noting that to some degree the above module concept are similar to that proposed by Rogers and Bottaci (1997). Rogers and Bottaci divide the modules into four categories: process machine primitives, motion units, modular fixturing and configurable control systems. The modular production systems (MPS) are built upon the appropriate selection of modules from these categories.

3 Cyclic reconfigurable flow shop descriptions

The flow shop is one traditional way of organizing manufacturing systems. In a flow shop, all jobs (parts) have the same route through serial machines while the sequences of the jobs on each machine may be different. Permutation flow shops are a special class of flow shops where the sequences of the jobs on each machine are identical. Garey *et al.* (1976) prove that the problem of scheduling for the permutation flow shop with more than two machines and makespan minimization as the objective is NP-complete. Extensive literature has been focused on developing heuristic procedures to find sub-optimal solutions and Framinan *et al.* (2004) give a good review and classification of heuristics for this problem. In manufacturing environment, cyclic scheduling policy is widely adopted to repetitively produce the so-called minimal part set (MPS), or product mix, where the MPS is the smallest possible set of product type quantities in which the numbers of assembled products of the various types are in the desired ratios. The manufacturing system produces one MPS each cycle and the throughput is represented as the inverse of the cycle time. Much effort has been devoted to the study of cyclic manufacturing systems (Crama and van de Klundert 1997, Kamoun and Sriskandarajah 1993).

Based on the idea in section 2, a kind of cyclic reconfigurable flow shops are proposed. For the convenience of descriptions, the following notations are made at first.

(1) $J = \{J_k | k = 1, 2, ..., |J|\}$ is a finite set of jobs.
(2) $M = \{M_j | j = 1, 2, ..., |M|\}$ is a finite set of machines.
(3) $m = \{m_i | i = 1, 2, ..., |m|\}$ is a finite set of modules.
(4) $m(J_k)$ denotes the set of modules required to manufacture J_k such that $m(J_k) \subseteq m$ and $m = \bigcup_{k=1}^{|J|} m(J_k)$.

(5) $\sigma(J_k) = \{(m_{i_1}, m_{i_2}) | m_{i_1}, m_{i_2} \in m(J_k)$ and m_{i_1} precedes $m_{i_2}\}$ denotes the set of module precedences required to manufacture J_k. The ordered pair (m_{i_1}, m_{i_2}) denotes that module m_{i_1} precedes module m_{i_2} when manufacturing J_k.
(6) $\sigma = \bigcup_{k=1}^{|J|} \sigma(J_k)$ denotes the set of module precedences.

Carried by each own pallet (cart or automatic guided vehicle (AGV)), jobs $J_1, J_2, ..., J_{|J|}$ access machines $M_1, M_2, ..., M_{|M|}$ sequentially (the job will not skip the machine even if there is no operation on the machine). After being processed by all the machines, the job is unloaded from the pallet and the pallet returns immediately to pick up the next job. The combination of modules and machines can be represented as a *configuration matrix* $\mathbf{Y}$ and each entry of $\mathbf{Y}$ is defined as

$$y_{i,j} = \begin{cases} 1 & \text{if } m_i \text{ is placed on } M_j \text{ under one configuration} \\ 0 & \text{otherwise} \end{cases}$$

$\mathbf{Y} \in \mathbb{B}^{|m| \times |M|}, \mathbb{B} = \{0, 1\}$. Because one module must be placed on only one machine under any configuration, $\mathbf{Y}$ should satisfy

$$\sum_{j=1}^{|M|} y_{i,j} = 1 \quad i = 1, 2, ..., |m|. \tag{1}$$

Let C denote the *configuration set* (i.e. the set of the configuration matrices) and $|C| = |M|^{|m|}$. The index of the machine that m_i is placed on can be represented as

$$MI(m_i) = \sum_{j=1}^{|M|} j y_{i,j} \quad i = 1, 2, ..., |m|.$$

According to the characteristics of flow shops (i.e. all jobs visit the machines in the increasing order of the machine indexes and every machine is visited once by each job), for any ordered pair (m_{i_1}, m_{i_2}), we have

$$MI(m_{i_1}) \leq MI(m_{i_2}) \quad (m_{i_1}, m_{i_2}) \in \sigma$$

or

$$\sum_{j=1}^{|M|} j y_{i_1,j} \leq \sum_{j=1}^{|M|} j y_{i_2,j} \quad (m_{i_1}, m_{i_2}) \in \sigma \tag{2}$$

where the equality holds if and only if m_{i_1}, m_{i_2} are placed on the same machine. Let $\mathbf{Y_f}$ denote the *feasible configuration matrix* satisfying constraints (1),(2) and C_f the *feasible configuration set* (i.e. the set of feasible configuration matrices). C_f is a subset of C, that is, $C_f \subseteq C$.

To simplify the analysis, the following assumptions are made

(1) The number of pallets carrying J_k is one and the buffers between machines are infinite.
(2) The processing time of J_k on m_i, denoted by $z_{k,i}$, is deterministic (positive constant) if $m_i \in m(J_k)$ or 0 otherwise.
(3) Compared with the processing time, the transportation time between machines, the setup time of machines and the changeover time between modules on the same machine can be neglected. The processing time of J_k on M_j, denoted by $w_{k,j}$, equals the sum of the processing time of J_k on all the needed modules placed on M_j, i.e.

$$w_{k,j} = \sum_{m_i \in m(J_k) \wedge MI(m_i)=j} z_{k,i}$$

further rewritten as

$$w_{k,j} = \sum_{i=1}^{|m|} z_{k,i} y_{i,j} \tag{3}$$

or

$$\mathbf{W} = \mathbf{ZY} \tag{4}$$

where $\mathbf{W} = (w)_{k,j} (\mathbf{Z} = (z)_{k,i})$ denotes the processing time matrix of jobs on machines (resp. on modules).
(4) The operations of jobs on machines are non-preemptive.

4 Timed event graph-based cyclic reconfigurable flow shop modeling

An ordinary Petri net (Murata 1989) is a 4-tuple, $PN = (P, T, F, K_0)$ where

$P = \{p_1, p_2, ..., p_{|P|}\}$ is a finite set of places.
$T = \{t_1, t_2, ..., t_{|T|}\}$ is a finite set of transitions.
$F \subseteq (P \times T) \bigcup (T \times P)$ is a set of directed arcs.
$K_0 : P \rightarrow \{0, 1, 2, ...\}$ is the initial marking.
$P \bigcap T = \emptyset$ and $P \bigcup T \neq \emptyset$.

An event graph (marked graph or decision-free net) is an ordinary Petri net such that each place p has exactly one input transition and exactly one output transition, i.e.

$$|{}^{\bullet}p| = |p^{\bullet}| = 1 \quad \forall p \in P$$

where ${}^{\bullet}p(p^{\bullet})$ denotes the set of input (resp. output) transitions of p. A timed event graph is an event graph with certain timing policy. The timing policy adopted in this paper is that the transitions are timed while the places are not. As soon as being enabled, transition t starts firing (i.e. consumes one token from each of its input places) and ends firing (i.e. generates one token to each of its output places) after some amount of time (release delay). The following results are well known for the timed event graphs (Commoner *et al.* 1971, Ramamoorthy and Ho 1980).

(1) An event graph is live if and only if each circuit contains at least one token in the initial marking.
(2) The number of tokens in each circuit remains constant in all the reachable markings.
(3) Let x_t^l be the time at which transition t starts firing for the l-th time. The cycle time of transition t is defined as

$$\lim_{l \to \infty} \frac{x_t^l}{l}.$$

In a live and strongly connected event graph, all transitions have the same cycle time λ and λ is given by

$$\lambda = \max_{\gamma \in \Gamma} \frac{\mu(\gamma)}{\kappa(\gamma)}$$

where γ denotes one circuit; Γ denotes the set of circuits; $\mu(\gamma)$ denotes the sum of release delay of the transitions in circuit γ and $\kappa(\gamma)$ denotes the number of tokens that circuit γ contains in the initial marking K_0.

Karp algorithm (Karp 1978), Howard algorithm (Cochet-Terrasson *et al.* 1998), linear programming method (Campos *et al.* 1992, Magott 1984, Yamada and Kataoka 1994) and etc. are available for evaluating λ.

The cyclic reconfigurable flow shop in which the sequences of jobs on each machine are identical and fixed, i.e. $J_1 J_2 ... J_{|J|}$, can be modeled as a timed event graph, $TEG_1 = (P, T, F, K_0)$ where

$P = P^b \bigcup P^r \bigcup P^{c_1} \bigcup P^{c_2}$ is the set of places where

$P^b = \{p^b_{k,j} | k = 1, 2, ..., |J|; j = 1, 2, ..., |M| - 1\}$ is the set of buffer places.
$P^r = \{p^r_{k,|M|} | k = 1, 2, ..., |J|\}$ is the set of resource places.
$P^{c_1} = \{p^{c_1}_{k,j} | j = 1, 2, .., |M|; k = 1, 2, ..., |J| - 1\}$ is the set of initially unmarked command places.
$P^{c_2} = \{p^{c_2}_{|J|,j} | j = 1, 2, ..., |M|\}$ is the set of initially marked command places.

$T = \{t_{k,j} | k = 1, 2, ..., |J|; j = 1, 2, ..., |M|\}$ is the set of transitions. Transition $t_{k,j}$ denotes the operation of job J_k on machine M_j and the release delay of $t_{k,j}$ is $w_{k,j}$.

$F = F^b \bigcup F^r \bigcup F^{c_1} \bigcup F^{c_2}$ is the set of directed arcs where
$F^b = \{(t_{k,j}, p^b_{k,j}), (p^b_{k,j}, t_{k,j+1}) | k = 1, 2, ..., |J|; j = 1, 2, ..., |M| - 1\}$ is the set of directed arcs that go from or to the buffer places.
$F^r = \{(t_{k,|M|}, p^r_{k,|M|}), (p^r_{k,|M|}, t_{k,1}) | k = 1, 2, ..., |J|\}$ is the set of directed arcs that go from or to the resource places.
$F^{c_1} = \{(t_{k,j}, p^{c_1}_{k,j}), (p^{c_1}_{k,j}, t_{k+1,j}) | j = 1, 2, .., |M|; k = 1, 2, ..., |J|-1\}$ is the set of directed arcs that go from or to the initially unmarked command places.
$F^{c_2} = \{(t_{|J|,j}, p^{c_2}_{|J|,j}), (p^{c_2}_{|J|,j}, t_{1,j}) | j = 1, 2, ..., |M|\}$ is the set of directed arcs that go from or to the initially marked command places.

$K_0 : P \rightarrow \{0, 1, 2, ...\}$ is the initial marking where

$$K_0(p) = \begin{cases} 0 & p \in P^b \\ 1 & p \in P^r \\ 0 & p \in P^{c_1} \\ 1 & p \in P^{c_2} \end{cases}.$$

The *buffer, resource and command places* are classified according to the method used in (Hillion and Proth 1989). TEG_1 is shown in figure 1 where bars represent transitions, circles represent places and dots represent tokens.

The sequence of jobs on M_j can be represented by a permutation π_j (i.e. a bijection from the set J to itself) and $\pi_j^{-1}(k)$ denotes the position that job J_k is in the permutation. For the cases that the sequences of jobs on each machine may be different, the cyclic reconfigurable flow shop can be modeled as a timed event graph extended with undirected arcs, $TEG_2 = (P, T, F, K_0)$ where

$P = P^b \bigcup P^r \bigcup P^{c_1} \bigcup P^{c_2}$ is the set of places where
$P^b = \{p^b_{k,j} | k = 1, 2, ..., |J|; j = 1, 2, ..., |M| - 1\}$ is the set of buffer places.
$P^r = \{p^r_{k,|M|} | k = 1, 2, ..., |J|\}$ is the set of resource places.
$P^{c_1} = \{p^{c_1}_{k_1 k_2,j} | j = 1, 2, .., |M|; k_1, k_2 = 1, 2, ..., |J|, k_1 < k_2\}$ is the set of

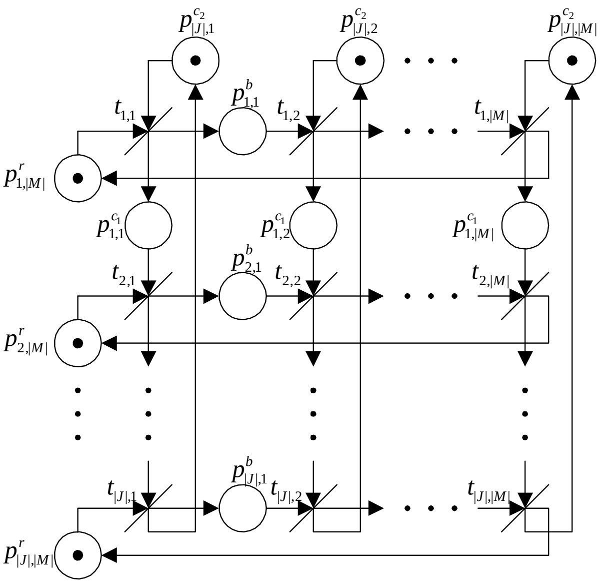

Figure 1. Timed event graph model TEG_1

initially unmarked command places.

$P^{c_2} = \{p^{c_2}_{k_1k_2,j} | j = 1, 2, ..., |M|; k_1, k_2 = 1, 2, ..., |J|, k_1 < k_2\}$ is the set of possibly initially marked command places.

$T = \{t_{k,j} | k = 1, 2, ..., |J|; j = 1, 2, ..., |M|\}$ is the set of transitions. Transition $t_{k,j}$ denotes the operation of job J_k on machine M_j and the release delay of $t_{k,j}$ is $w_{k,j}$.

$F = F^b \bigcup F^r \bigcup F^{c_1} \bigcup F^{c_2}$ is the set of directed and undirected arcs where

$F^b = \{(t_{k,j}, p^b_{k,j}), (p^b_{k,j}, t_{k,j+1}) | k = 1, 2, ..., |J|; j = 1, 2, ..., |M| - 1\}$ is the set of directed arcs that go from or to the buffer places.

$F^r = \{(t_{k,|M|}, p^r_{k,|M|}), (p^r_{k,|M|}, t_{k,1}) | k = 1, 2, ..., |J|\}$ is the set of directed arcs that go from or to the resource places.

$F^{c_1} = \{(t_{k_1,j}, p^{c_1}_{k_1k_2,j}), (p^{c_1}_{k_1k_2,j}, t_{k_2,j}) | j = 1, 2, ..., |M|; k_1, k_2 = 1, 2, ..., |J|, k_1 < k_2\}$ is the set of undirected arcs that connect the initially unmarked command places.

$F^{c_2} = \{(t_{k_1,j}, p^{c_2}_{k_1k_2,j}), (p^{c_2}_{k_1k_2,j}, t_{k_2,j}) | j = 1, 2, ..., |M|; k_1, k_2 = 1, 2, ..., |J|, k_1 < k_2\}$ is the set of undirected arcs that connect the possibly initially marked command places.

$p^{c_1}_{k_1k_2,j}, p^{c_2}_{k_1k_2,j}$ and the related undirected arcs are shown in figure 2.

$K_0 : P \rightarrow \{0, 1, 2, ...\}$ is the initial marking where

$\forall p^b_{k,j} \in P^b, K_0(p^b_{k,j}) = 0.$

$\forall p^r_{k,|M|} \in P^r, K_0(p^r_{k,|M|}) = 1.$

$\forall p^{c_1}_{k_1k_2,j} \in P^{c_1}, K_0(p^{c_1}_{k_1k_2,j}) = 0.$

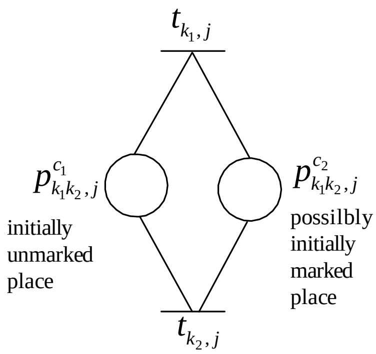

Figure 2. $p^{c1}_{k_1k_2,j}, p^{c2}_{k_1k_2,j}$ and the related undirected arcs

$\forall p^{c_2}_{k_1k_2,j} \in P^{c_2}$,

$$K_0(p^{c_2}_{k_1k_2,j}) = \begin{cases} 1 & \text{if } \pi_j^{-1}(k_1) = 1 \text{ and } \pi_j^{-1}(k_2) = |J| \\ 1 & \text{if } \pi_j^{-1}(k_1) = |J| \text{ and } \pi_j^{-1}(k_2) = 1 \\ 0 & \text{otherwise} \end{cases} .$$

If π_j is known, the undirected arcs can be converted into the directed arcs under some conditions. In table 1, column 1, 2 and 3 corresponds to the undirected arcs, directed arcs and the conditions respectively.

Table 1. Directed arcs converted from undirected arcs under some conditions when π_j is known

Undirected arcs	Directed arcs	Conditions
$(t_{k_1,j}, p^{c1}_{k_1k_2,j}), (p^{c1}_{k_1k_2,j}, t_{k_2,j})$	$(t_{k_1,j}, p^{c1}_{k_1k_2,j}), (p^{c1}_{k_1k_2,j}, t_{k_2,j})$	$\pi_j^{-1}(k_1) < \pi_j^{-1}(k_2)$
	$(t_{k_2,j}, p^{c1}_{k_1k_2,j}), (p^{c1}_{k_1k_2,j}, t_{k_1,j})$	$\pi_j^{-1}(k_1) > \pi_j^{-1}(k_2)$
$(t_{k_1,j}, p^{c2}_{k_1k_2,j}), (p^{c2}_{k_1k_2,j}, t_{k_2,j})$	$(t_{k_2,j}, p^{c2}_{k_1k_2,j}), (p^{c2}_{k_1k_2,j}, t_{k_1,j})$	$\pi_j^{-1}(k_1) = 1$ and $\pi_j^{-1}(k_2) = \|J\|$
	$(t_{k_1,j}, p^{c2}_{k_1k_2,j}), (p^{c2}_{k_1k_2,j}, t_{k_2,j})$	$\pi_j^{-1}(k_1) = \|J\|$ and $\pi_j^{-1}(k_2) = 1$
	$(t_{k_1,j}, p^{c2}_{k_1k_2,j}), (p^{c2}_{k_1k_2,j}, t_{k_2,j})$	$\pi_j^{-1}(k_1) < \pi_j^{-1}(k_2)$ and $\pi_j^{-1}(k_1) - 1 + \|J\| - \pi_j^{-1}(k_2) \neq 0$
	$(t_{k_2,j}, p^{c2}_{k_1k_2,j}), (p^{c2}_{k_1k_2,j}, t_{k_1,j})$	$\pi_j^{-1}(k_1) > \pi_j^{-1}(k_2)$ and $\|J\| - \pi_j^{-1}(k_1) + \pi_j^{-1}(k_2) - 1 \neq 0$

5 Timed event graph-based cyclic reconfigurable flow shop optimization

The throughput can be represented as the inverse of the cycle time (i.e. $1/\lambda$). Obviously, the smaller the cycle time is, the greater the throughput is. As described in section 2, different configurations would result in different system performance. If cycle time is selected as the system performance to be optimized, the *optimal configuration* can be defined as the one under which the reconfigurable flow shop functions with the minimal cycle time. The optimal configuration can be obtained by solving the mixed-integer programs derived from the timed event graph models.

Based on TEG_1, the corresponding mixed-integer program $MIP1$ can be constructed as follows. For every $p \in P$ and its related directed arcs, there exists one inequality constraint

$$x_{p^\bullet} - x_{^\bullet p} + K_0(p)\lambda \geq w_{^\bullet p}$$

where $p^\bullet$($^\bullet p$) denotes the output (resp. input) transition of p and $w_{^\bullet p}$ denotes the release delay of $^\bullet p$. By combining with constraints (1), (2) and replacing $w_{k,j}$ with equation (3), $MIP1$ can be formulated in the following standard form.

$$\begin{array}{ll}
\min \quad \lambda & \\
\text{subject to} & \\
x_{k,j+1} - x_{k,j} - \sum_{i=1}^{|m|} z_{k,i} y_{i,j} \geq 0 & k = 1,2,...,|J|; j = 1,2,...,|M|-1 \\
x_{k,1} - x_{k,|M|} + \lambda - \sum_{i=1}^{|m|} z_{k,i} y_{i,|M|} \geq 0 & k = 1,2,...,|J| \\
x_{k+1,j} - x_{k,j} - \sum_{i=1}^{|m|} z_{k,i} y_{i,j} \geq 0 & j = 1,2,...,|M|; k = 1,2,...,|J|-1 \\
x_{1,j} - x_{|J|,j} + \lambda - \sum_{i=1}^{|m|} z_{|J|,i} y_{i,j} \geq 0 & j = 1,2,...,|M| \\
\sum_{j=1}^{|M|} y_{i,j} = 1 & i = 1,2,...,|m| \\
\sum_{j=1}^{|M|} j y_{i_2,j} - \sum_{j=1}^{|M|} j y_{i_1,j} \geq 0 & (m_{i_1}, m_{i_2}) \in \sigma \\
\lambda, x_{k,j} \geq 0 \quad \text{and} \quad y_{i,j} \in \{0,1\} &
\end{array} \tag{5}$$

For the case that the sequences of jobs on each machine may be different, new binary variables η_{j,k_1,k_2} are introduced and defined as

$$\eta_{j,k_1,k_2} = \begin{cases} 1 & \text{if } J_{k_1} \text{ precedes } J_{k_2} \text{ in permutation } \pi_j, \text{ i.e. } \pi_j^{-1}(k_1) < \pi_j^{-1}(k_2) \\ 0 & \text{otherwise} \end{cases}.$$

$\pi_j^{-1}(k)$ can be represented as

$$\pi_j^{-1}(k) = 1 + \sum_{k_s<k} \eta_{j,k_s,k} + \sum_{k_l>k} (1 - \eta_{j,k,k_l}) \tag{6}$$

where $\sum_{k_s<k} \eta_{j,k_s,k} (\sum_{k_l>k}(1 - \eta_{j,k,k_l}))$ is the number of jobs preceding J_k whose index is smaller (resp. larger) than k in permutation π_j.

Similar to $MIP1$, the mixed-integer program $MIP2$ can be derived from

$TEG2$ in the following way.

(i) For every $p \in P^b \bigcup P^r$ and its related directed arcs, there exists one inequality constraint

$$x_{p\bullet} - x_{\bullet p} + K_0(p)\lambda \geq w_{\bullet p}.$$

(ii) For every $p^{c_1}_{k_1 k_2, j} \in P^{c_1}$ and its related undirected arcs, there exist a pair of inequality constraints

$$\begin{array}{l} x_{k_2,j} - x_{k_1,j} + R(1 - \eta_{j,k_1,k_2}) \geq w_{k_1,j} \\ x_{k_1,j} - x_{k_2,j} + R\eta_{j,k_1,k_2} \geq w_{k_2,j} \end{array}$$

where R is a sufficiently large positive constant.

(iii) For every $p^{c_2}_{k_1 k_2, j} \in P^{c_2}$ and its related undirected arcs, there exist a pair of inequality constraints

$$\begin{array}{l} x_{k_1,j} - x_{k_2,j} + \lambda + R[\pi_j^{-1}(k_1) - 1 + |J| - \pi_j^{-1}(k_2)] \geq w_{k_2,j} \\ x_{k_2,j} - x_{k_1,j} + \lambda + R[\pi_j^{-1}(k_2) - 1 + |J| - \pi_j^{-1}(k_1)] \geq w_{k_1,j} \end{array}.$$

Replaced with equation (6), the above constraints can be further rewritten as

$$\begin{array}{l} x_{k_1,j} - x_{k_2,j} + \lambda + R[1 + \sum_{k_s<k_1} \eta_{j,k_s,k_1} + \sum_{k_l>k_1} (1 - \eta_{j,k_1,k_l}) - 1 \\ +|J| - 1 - \sum_{k_s<k_2} \eta_{j,k_s,k_2} - \sum_{k_l>k_2} (1 - \eta_{j,k_2,k_l})] \geq w_{k_2,j} \\ x_{k_2,j} - x_{k_1,j} + \lambda + R[1 + \sum_{k_s<k_2} \eta_{j,k_s,k_2} + \sum_{k_l>k_2} (1 - \eta_{j,k_2,k_l}) - 1 \\ +|J| - 1 - \sum_{k_s<k_1} \eta_{j,k_s,k_1} - \sum_{k_l>k_1} (1 - \eta_{j,k_1,k_l})] \geq w_{k_1,j} \end{array}.$$

$MIP2$ is formulated in the standard form as follows.

$$
\begin{array}{ll}
\min \quad \lambda & \\
\text{subject to} & \\
x_{k,j+1} - x_{k,j} - \sum_{i=1}^{|m|} z_{k,i} y_{i,j} \geq 0 & k = 1,2,...,|J|; j = 1,2,...,|M|-1 \\
x_{k,1} - x_{k,|M|} + \lambda - \sum_{i=1}^{|m|} z_{k,i} y_{i,|M|} \geq 0 & k = 1,2,...,|J| \\
x_{k_2,j} - x_{k_1,j} + R(1-\eta_{j,k_1,k_2}) - \sum_{i=1}^{|m|} z_{k_1,i} y_{i,j} \geq 0 & j = 1,2,...,|M|; \\
 & k_1,k_2 = 1,2,...,|J|, k_1 < k_2 \\
x_{k_1,j} - x_{k_2,j} + R\eta_{j,k_1,k_2} - \sum_{i=1}^{|m|} z_{k_2,i} y_{i,j} \geq 0 & j = 1,2,...,|M|; \\
 & k_1,k_2 = 1,2,...,|J|, k_1 < k_2 \\
x_{k_1,j} - x_{k_2,j} + \lambda + R[1 + \sum_{k_s<k_1} \eta_{j,k_s,k_1} + \sum_{k_l>k_1}(1-\eta_{j,k_1,k_l}) - 1 & \\
+|J| - 1 - \sum_{k_s<k_2} \eta_{j,k_s,k_2} - \sum_{k_l>k_2}(1-\eta_{j,k_2,k_l})] - \sum_{i=1}^{|m|} z_{k_2,i} y_{i,j} \geq 0 & j = 1,2,...,|M|; \\
 & k_1,k_2 = 1,2,...,|J|, k_1 < k_2 \\
x_{k_2,j} - x_{k_1,j} + \lambda + R[1 + \sum_{k_s<k_2} \eta_{j,k_s,k_2} + \sum_{k_l>k_2}(1-\eta_{j,k_2,k_l}) - 1 & \\
+|J| - 1 - \sum_{k_s<k_1} \eta_{j,k_s,k_1} - \sum_{k_l>k_1}(1-\eta_{j,k_1,k_l})] - \sum_{i=1}^{|m|} z_{k_1,i} y_{i,j} \geq 0 & j = 1,2,...,|M|; \\
 & k_1,k_2 = 1,2,...,|J|, k_1 < k_2 \\
\sum_{j=1}^{|M|} y_{i,j} = 1 & i = 1,2,...,|m| \\
\sum_{j=1}^{|M|} j y_{i_2,j} - \sum_{j=1}^{|M|} j y_{i_1,j} \geq 0 & (m_{i_1}, m_{i_2}) \in \sigma \\
\lambda, x_{k,j} \geq 0, y_{i,j} \in \{0,1\} \quad \text{and} \quad \eta_{j,k_1,k_2} \in \{0,1\} \text{ for } k_1 < k_2 &
\end{array}
\tag{7}
$$

where R is a sufficiently large positive constant.

Cyclic permutation reconfigurable flow shops, in which the sequences of jobs on each machine are identical, are a special class of reconfigurable flow shop. The sequences of jobs on each machine can be represented by the same permutation π. A new variable β_{k_1,k_2} is introduced and defined as

$$
\beta_{k_1,k_2} = \begin{cases} 1 & \text{if } J_{k_1} \text{ precedes } J_{k_2} \text{ in permutation } \pi, \text{ i.e. } \pi^{-1}(k_1) < \pi^{-1}(k_2) \\ 0 & \text{otherwise} \end{cases} .
$$

The following equality holds

$$
\eta_{j,k_1,k_2} = \beta_{k_1,k_2} \quad j = 1,2,...,|M| \tag{8}
$$

Replace η_{j,k_1,k_2} with equation (8) and we get the corresponding mixed-integer program $MIP3$ for the cycle permutation reconfigurable flow shop.

$$
\begin{array}{ll}
\min \quad \lambda & \\
\text{subject to} & \\
x_{k,j+1} - x_{k,j} - \sum_{i=1}^{|m|} z_{k,i} y_{i,j} \geq 0 & k = 1,2,...,|J|; j = 1,2,...,|M|-1 \\
x_{k,1} - x_{k,|M|} + \lambda - \sum_{i=1}^{|m|} z_{k,i} y_{i,|M|} \geq 0 & k = 1,2,...,|J| \\
x_{k_2,j} - x_{k_1,j} + R(1-\beta_{k_1,k_2}) - \sum_{i=1}^{|m|} z_{k_1,i} y_{i,j} \geq 0 & j = 1,2,...,|M|; \\
 & k_1, k_2 = 1,2,...,|J|, k_1 < k_2 \\
x_{k_1,j} - x_{k_2,j} + R\beta_{k_1,k_2} - \sum_{i=1}^{|m|} z_{k_2,i} y_{i,j} \geq 0 & j = 1,2,...,|M|; \\
 & k_1, k_2 = 1,2,...,|J|, k_1 < k_2 \\
x_{k_1,j} - x_{k_2,j} + \lambda + R[1 + \sum_{k_s<k_1} \beta_{k_s,k_1} + \sum_{k_l>k_1}(1-\beta_{k_1,k_l}) - 1 & \\
+|J| - 1 - \sum_{k_s<k_2} \beta_{k_s,k_2} - \sum_{k_l>k_2}(1-\beta_{k_2,k_l})] - \sum_{i=1}^{|m|} z_{k_2,i} y_{i,j} \geq 0 & j = 1,2,...,|M|; \\
 & k_1, k_2 = 1,2,...,|J|, k_1 < k_2 \\
x_{k_2,j} - x_{k_1,j} + \lambda + R[1 + \sum_{k_s<k_2} \beta_{k_s,k_2} + \sum_{k_l>k_2}(1-\beta_{k_2,k_l}) - 1 & \\
+|J| - 1 - \sum_{k_s<k_1} \beta_{k_s,k_1} - \sum_{k_l>k_1}(1-\beta_{k_1,k_l})] - \sum_{i=1}^{|m|} z_{k_1,i} y_{i,j} \geq 0 & j = 1,2,...,|M|; \\
 & k_1, k_2 = 1,2,...,|J|, k_1 < k_2 \\
\sum_{j=1}^{|M|} y_{i,j} = 1 & i = 1,2,...,|m| \\
\sum_{j=1}^{|M|} j y_{i_2,j} - \sum_{j=1}^{|M|} j y_{i_1,j} \geq 0 & (m_{i_1}, m_{i_2}) \in \sigma \\
\lambda, x_{k,j} \geq 0, y_{i,j} \in \{0,1\} \quad \text{and} \quad \beta_{k_1,k_2} \in \{0,1\} \text{ for } k_1 < k_2 &
\end{array}
\tag{9}
$$

where R is a sufficiently large positive constant.

The number of decision variables and constrains in $MIP1, MIP2$ and $MIP3$ are shown in table 2 respectively.

Table 2. The number of decision variables and constraints in $MIP1, MIP2$ and $MIP3$

Mixed-integer program	The number of decision variables	The number of constraints
$MIP1$	$1 + \lvert J\rvert\lvert M\rvert + \lvert m\rvert\lvert M\rvert$	$2\lvert J\rvert\lvert M\rvert + \lvert m\rvert + \lvert\sigma\rvert$
$MIP2$	$\frac{\lvert J\rvert\lvert M\rvert(\lvert J\rvert+1)}{2} + 1 + \lvert m\rvert\lvert M\rvert$	$\lvert J\rvert\lvert M\rvert(2\lvert J\rvert - 1) + \lvert m\rvert + \lvert\sigma\rvert$
$MIP3$	$\frac{\lvert J\rvert(\lvert J\rvert-1)}{2} + 1 + \lvert J\rvert\lvert M\rvert + \lvert m\rvert\lvert M\rvert$	$\lvert J\rvert\lvert M\rvert(2\lvert J\rvert - 1) + \lvert m\rvert + \lvert\sigma\rvert$

6 Case study

In this example, the cyclic reconfigurable flow shop consists of 3 jobs, 3 machines and 4 modules. J_1 is to be processed on modules in the sequence of m_3m_4, J_2 in the sequence of m_1m_4 and J_3 in the sequence of $m_1m_2m_3$. The processing time matrix of jobs on modules

$$\mathbf{Z} = \begin{pmatrix} 0 & 0 & 45 & 30 \\ 23 & 0 & 0 & 61 \\ 60 & 76 & 5 & 0 \end{pmatrix}.$$

We have

$$\begin{aligned}
&J = \{J_1, J_2, J_3\} \\
&M = \{M_1, M_2, M_3\} \\
&m(J_1) = \{m_3, m_4\} \\
&m(J_2) = \{m_1, m_4\} \\
&m(J_3) = \{m_1, m_2, m_3\} \\
&m = \bigcup_{k=1}^{3} m(J_k) = \{m_1, m_2, m_3, m_4\} \\
&\sigma(J_1) = \{(m_3, m_4)\} \\
&\sigma(J_2) = \{(m_1, m_4)\} \\
&\sigma(J_3) = \{(m_1, m_2), (m_2, m_3)\} \\
&\sigma = \bigcup_{k=1}^{3} \sigma(J_k) = \{(m_1, m_2), (m_2, m_3), (m_3, m_4), (m_1, m_4)\}.
\end{aligned}$$

Let $R = 10,000$ and the optimal configuration, the minimal cycle time (λ^*) are shown in table 3 after solving $MIP1, MIP2$ and $MIP3$.

Table 3. The optimal solutions to $MIP1, MIP2$ and $MIP3$

Mixed-integer program	The sequence of jobs on M_j			The modules placed on M_j			λ^*
	M_1	M_2	M_3	M_1	M_2	M_3	
$MIP1$	$J_1J_2J_3$	$J_1J_2J_3$	$J_1J_2J_3$	m_1	m_2	m_3m_4	150
$MIP2$	$J_3J_2J_1$	$J_3J_1J_2$	$J_3J_2J_1$	m_1	m_2	m_3m_4	141
$MIP3$	$J_3J_2J_1$	$J_3J_2J_1$	$J_3J_2J_1$	m_1	m_2m_3	m_4	141

7 Conclusions and future work

Based on the idea that modules are independent of machines and different combinations of modules and machines result in different configurations, the cyclic reconfigurable flow shop is proposed in this paper. The cyclic reconfigurable flow shop can be modeled as a timed event graph. Different cases of cyclic reconfigurable flow shops are discussed respectively and the optimal configuration can be obtained by solving the corresponding mixed-integer program derived from the timed event graph model.

Future work should be concentrated on the following aspects:

(1) The optimal configuration is defined as the one with the minimal cycle time. More generally, other factors such as the reconfigure cost should be considered in evaluating the overall system performance.
(2) For the case that $z_{k,i}$ is random, stochastic models should be established to evaluate the system performance.
(3) Practically, the buffers between machines are limited. The cyclic reconfigurable flow shops with limited buffers should be studied in the future.
(4) The idea in section 2 can be applied to other types of manufacturing systems such as job shops.

Acknowledgement

The research work is supported by National Key Fundamental Research and Development Project of China (973, No.2002CB312200).

REFERENCES

Abdi, M. R. and Labib, A. W., A design strategy for reconfigurable manufacturing systems (RMSs) using analytical hierarchical process (AHP): a case study. *International Journal of Production Research,* 2003, **41**(10), 2273–2299.

Abdi, M. R. and Labib, A. W., Grouping and selecting products: the design key of reconfigurable manufacturing systems (RMSs). *International Journal of Production Research,* 2004a, **42**(3), 521–546.

Abdi, M. R. and Labib, A. W., Feasibility study of the tactical design justification for reconfigurable manufacturing systems using the fuzzy analytical hierarchical process. *International Journal of Production Research,* 2004b, **42**(15), 3055–3076.

Bruccoleri, M., Amico, M. and Perrone, G., Distributed intelligent control of exceptions in reconfigurable manufacturing systems. *International Journal of Production Research,* 2003, **41**(7), 1393–1412.

Campos, J., Chiola, G., Colom, J. M., and Silva, M., Properties and performance bounds for timed marked graphs. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* 1992, **39**(5), 386–401.

Cochet-Terrasson, J., Cohen, G., Gaubert, S., Gettrick, M. M. and Quadrat, J.-P., Numerical computation of spectral elements in max-plus algebra. In: *Proceedings of the IFAC Conference on System Structure and Control,* Nantes, France, July, 1998.

Commoner, F., Holt, A. W., Even, S., Pnueli, A., Marked directed graphs. *Journal of Computer and System Sciences,* 1971, **5**(5), 511–523.

Crama, Y. and van de Klundert, J., Cyclic scheduling of identical parts in a robotic cell. *Operations Research,* 1997, **45**(6), 952–965.

Framinan, J. M., Gupta, J. N. D. and Leisten, R., A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. *Journal of the Operational Research Society,* 2004, **55**(12), 1243–1255.

Garey, M. R., Johnson, D. S. and Sethi, R., The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research,* 1976, **1**(2), 117–129.

Hillion, H. and Proth, J.-M., Performance evaluation of job-shop systems using timed event-graphs. *IEEE Transactions on Automatic Control,* 1989, **34**(1), 3–9.

Kamoun, H. and Sriskandarajah, C., The complexity of scheduling jobs in repetitive manufacturing systems. *European Journal of Operational Research,* 1993, **70**(3), 350–364.

Karp, R. M., A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics,* 1978, **23**(3), 309–311.

Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. and Van Brussel, H., Reconfigurable manufacturing systems. *Annals of the CIRP,* 1999, **48**(2), 527–540.

Landers, R. G., Min, B. K. and Koren, Y., Reconfigurable machine tools. *Annals of the CIRP,* 2001, **50**(1), 269–274.

Magott, J., Performance evaluation of concurrent systems using Petri nets. *Information Processing Letters,* 1984, **18**(1), 7–13.

Murata, T., Petri nets: properties, analysis and applications. *Proceedings of the IEEE,* 1989, **77**(4), 541–580.

Park, E., Tilbury, D.M. and Khargonekar, P.P., Modular logic controllers for machining systems: Formal representation and performance analysis using Petri nets. *IEEE Transactions on Robotics and Automation,* 1999, **15**(6), 1046–1061.

Park, E., Tilbury, D.M. and Khargonekar, P.P., A modelling and analysis methodology for modular logic controllers of machining systems using Petri net formalism. *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews,* 2001, **31**(2), 168–188.

Ramamoorthy, C. V. and Ho, G. S., Performance evaluation of asynchronous concurrnt systems using Petri nets. *IEEE Transactions on Software Engineering,* 1980, **6**(5), 440–449.

Rogers, G. G. and Bottaci, L., Modular production systems: a new manufacturing paradigm. *Journal of Intelligent Manufacturing,* 1997, **8**(2), 147–156.

Yamada, T. and Kataoka, S., On some LP problems for performance evaluation of timed marked graphs. *IEEE Transactions on Automatic Control,* 1994, **39**(3), 696–698.

Yigit, A. S., Ulsoy, A. G. and Allahverdi, A., Optimizing modular product design for reconfigurable manufacturing. *Journal of Intelligent Manufacturing,* 2002, **13**(4), 309–316.

Yigit, A. S. and Allahverdi, A., Optimal selection of module instances for modular products in reconfigurable manufacturing systems. *International Journal of Production Research,* 2003, **41**(17), 4063–4074.

Zhao, X. B., Wang, J. C. and Luo, Z. B., A stochastic model of a reconfigurable manufacturing system—Part 1: A framework. *International Journal of Production Research,* 2000a, **38**(10), 2273–2285.

Zhao, X. B., Wang, J. C. and Luo, Z. B., A stochastic model of a reconfigurable manufacturing system—Part 2: Optimal configurations. *International Journal of Production Research,* 2000b, **38**(12), 2829–2842.

Zhao, X. B., Wang, J. C. and Luo, Z. B., A stochastic model of a reconfigurable manufacturing system—Part 3: Optimal selection policy. *International Journal of Production Research,* 2001a, **39**(4), 747–758.

Zhao, X. B., Wang, J. C. and Luo, Z. B., A stochastic model of a reconfigurable manufacturing system—Part 4: Performance measure. *International Journal of Production Research,* 2001b, **39**(6), 1113–1126.