

A dynamic heuristic-based algorithm to part input sequencing in flexible manufacturing systems for mass customization capability

Yumin He · Milton L. Smith

Published online: 22 April 2008
© Springer Science+Business Media, LLC 2008

Abstract In the increasingly competitive global markets, enterprises face challenges in responding to customer orders quickly, as well as producing customized products cost-effectively. This paper proposes a dynamic heuristic-based algorithm for the part input sequencing problem of flexible manufacturing systems (FMSs) in a mass customization (MC) environment. The FMS manufactures a variety of parts, and customer orders arrive dynamically with order size as small as one. Segmental set functions are established in the proposed algorithm to apply the strategy of dynamic workload balancing, and the shortest processing time (SPT) scheduling rule. Theoretical analysis is performed and the effectiveness of the algorithm in dynamic workload balancing under the complex and dynamic environment is proven. The application of the algorithm is illustrated by an example. The potential of its practical applications to the FMSs in make-to-order (MTO) supply chains is also discussed. Further research is provided.

Keywords Part input sequencing · Flexible manufacturing systems (FMSs) · Mass customization (MC) · Set function · Dynamic workload balancing

1 Introduction

Enterprises are confronted by the challenges of global markets of the 21st century. Customer requirements are becoming increasingly important. They demand

Y. He (✉)
Department of Information System, Beihang University, Beijing 100083, P.R. China
e-mail: heyumin@buaa.edu.cn

M. L. Smith
Department of Industrial Engineering, Texas Tech University, Lubbock, TX 79409, USA
e-mail: Milton.Smith@ttu.edu

products and expect their orders to be handled quickly, accurately, and cost-effectively (Shapiro et al. 2004). On the other hand, customer orders become more frequent, and order sizes may be as small as one item. Pine et al. (1993) have described a factory manufacturing customized pagers in lot sizes as small as one within hours of an order arriving from a customer. Therefore, enterprises have to meet the challenges of quickly responding to customer orders as well as producing customized products cost-effectively to survive in competitive global markets.

Mass customization (MC) is adopted by enterprises as a manufacturing strategy because it calls for flexibility and quick responsiveness. In an ever-changing environment, people, processes, units, and technology reconfigure to give customers exactly what they want with low cost, high quality, and variety (Pine et al. 1993). In order to realize MC, manufacturers need to adopt highly reactive strategies. Reactive strategies are also adopted in supply chains such as the emergence of make-to-order (MTO) supply chains because supply chains are now in new and complex environments. Suppliers are facing complex manufacturing environments. They need to respond to demand fluctuations and to produce requirements at a low price. They cannot afford to make on-time deliveries by carrying a large amount of inventory.

A flexible manufacturing system (FMS) is a computer-controlled manufacturing system that produces a variety of part types simultaneously for different demand volumes. The flexible manufacturing technology is an important enabler of mass customization. Pine et al. (1993) have pointed out that an FMS could be adopted to automate a module for MC because it can choose instantly any product component within its wide envelope of variety.

The FMS part input sequencing problem is the determination of a sequence for inputting parts into a flexible manufacturing system. It falls in the domain of FMS scheduling. Researchers have investigated the problem, and both static and dynamic approaches have been developed. This paper considers the FMS part input sequencing problem in an MC environment. Customer orders arrive dynamically with order size as small as one; part processing information may be available in advance. We propose a dynamic heuristic-based algorithm for the problem, which balances workload dynamically and utilizes segmental set functions to apply strategies as well as their priorities.

The remainder of the paper is organized as follows. Section 2 briefly summarizes relevant literature in FMS part input sequencing. Section 3 presents a solution procedure, which is a dynamic heuristic-based algorithm. Theoretical analysis of the proposed algorithm is presented in Sect. 4. The application of the proposed algorithm is illustrated by an example in Sect. 5. Conclusions, managerial implications, and further research are provided in Sect. 6.

2 Part input sequencing in FMSs

Relevant research in FMS part input sequencing from prior literature is summarized in this section. Stecke (1983) defined five production planning problems for FMS operations: the part type selection problem, the machine grouping problem, the

production ratio problem, the resource allocation problem, and the loading problem. The part type selection problem is defined as the determination of a subset for immediate and simultaneous processing among a set of part types that have production requirements. The production ratio problem is the determination of the part mix ratios at which a set of part types selected should be produced over the next production period.

Stecke and Kim (1988) studied FMS part type selection approaches and presented a flexible approach to select part types and simultaneously determine their mix ratios to balance aggregate machine workload in a flexible flow system (FFS). The comparative results indicate that the flexible approach enables the FFS to be more highly utilized. Stecke and Kim (1991) detailed the flexible approach. They illustrated how existing procedures that determine the relative mix ratios balancing workload of ordered part types can contribute also to part type selection. The analyses of fixtures, carts, and travel times were conducted and it was found that the flexible approach required fewer dedicated fixtures than batching. Stecke and Toczyłowski (1992) considered FMS dynamic part type selection. Mathematical programming models were developed. Their approach allows a certain production flexibility to improve the utilization of the system even at the possible expense of the completion of some parts earlier. Stecke (1992) made in-depth study on the part mix ratios and developed procedures to determine part mix ratios for independent demands in FMSs. Part input sequencing and machine breakdown situations were considered. It pointed out the usefulness of the part mix ratios in solving other planning and operating problems such as providing guidelines for determining an appropriate part input sequence.

Early research in FMS part input sequencing includes its consideration in combination with FMS production planning problems. Even though early research was not mainly focused on the part input sequencing problem, the researchers developed different approaches for the problem. Stecke and Kim (1991) applied an algorithm, a modified version of Johnson's algorithm, to determine a part input sequence in FFSs. They demonstrated these same ratios can be useful in determining a part input sequence to balance workload over time and suggested further research in determining a good part input sequence. Stecke (1992) also addressed part input sequencing in FMSs and mentioned several approaches, although the primary focus of the research was not on FMS part input sequencing. Further research on developing a more precise algorithm to find a part input sequence was suggested.

O'Keefe and Rao (1992) compared six part input sequencing approaches for an FFS: three static approaches and three dynamic approaches. Two new dynamic methods, look-ahead simulation and a fuzzy heuristic rule base, were considered for inputting a part from available part types into an FFS at a decision point. They concluded that the static determination of the best input sequence is appropriate for a stable FFS but a rapidly changing FFS may benefit from a dynamic part input method.

Stecke and Smith (1996) studied part input sequencing in FFSs. They integrated part input sequences determination with part mix ratio determination and revealed several significant conclusions regarding part mix ratios, part input sequences, and

look-ahead capability. These include the determination of the part mix ratios proved to be more significant in performance improvement than the determination of part input sequences, the robustness of the IP formulations, and equally high overall machine utilizations by look-ahead capability at lower levels of work-in-process (WIP).

Kim et al. (2001) performed a study on the FMS part input sequencing problem. They decomposed the problem into the part input grouping problem and the sequencing problem. Heuristic algorithms were developed for the part input grouping problem and a number of scheduling rules were used for the sequencing problem. However, their results indicate that dispatch rules for the sequencing problem do not have much influence on the system performance of FMSs.

In summary, researchers have studied the FMS production planning and operation management problems for decades. FMS part input sequencing has also been an interesting research topic. From the brief summary of the typical research in FMS part input sequencing from the literature it can be seen that a variety of approaches have been developed, including batches of MPS, the ordered MPS, the permutation of an MPS, the modified Johnson's algorithm, look-ahead simulation, fuzzy rule base, as well as other heuristics.

3 Proposed solution procedure

The objective of this research is to develop an effective approach for the part input sequencing problem of FMSs in a mass customization environment. The FMS produces a variety of parts. Customer orders arrive dynamically with order size as small as one and therefore production requirements cannot be determined completely before a production cycle begins. We propose a dynamic heuristic-based algorithm for the FMS part input sequencing problem.

Workload balance has been recognized in the literature to be able to eliminate bottlenecks and to increase productivity in FMSs. Stecke and Morin (1985) showed that balancing workload maximizes expected production for production planning in certain types of FMSs. Literature has illustrated that the shortest processing time (SPT) scheduling rule can increase productivity in FMS scheduling. Choi and Malstrom (1988) examined several scheduling rules in an FMS by a physical simulator and concluded that SPT results in high output, low throughput time, and low WIP inventory in the FMS. Because the objective of the proposed algorithm is to increase productivity, strategies that can increase productivity are applied in the development of the algorithm. The strategy of dynamic workload balancing is applied. Then, SPT is applied. Also, first-in first-out (FIFO) is applied when more parts have the same priority as the results of dynamic workload balancing and SPT.

Segmental set functions are generated to include the strategy of dynamic workload balancing, and the SPT scheduling rule. Instead of an operation processing time, total time of operations of a part is used in SPT for part input sequencing. The dynamic workload of a process station is defined as total time of remaining operations of parts inputted into the FMS but not finished on the corresponding process station. It is

dependent on the current status of an operating FMS. It can be obtained using a shop-floor monitoring system. Contemporary technology can be applied in developing a monitoring system (Kimura and Kanda 2005).

The proposed algorithm selects and inputs a part dynamically at each input decision point that occurs when an inputted part completes its operations and leaves. The algorithm is state dependent and is composed of two phases: the initialization phase and the iteration phase. The initialization phase of the algorithm is used to input parts quickly to process stations initially for simultaneous processing. The quantity of initially inputted parts should not be large and should be decided according to the capacity of an FMS to ensure no blocking or deadlocking of the FMS occurs. The iteration phase of the algorithm, which involves processing a part to completion and introduction of a new part, is utilized repeatedly to dynamically select and input parts.

The algorithm is depicted by the use of the concepts and approaches of discrete mathematics (Johnsonbaugh 2004). Notation utilized in the algorithm is summarized in Table 1; x , y , z , yy , yz , zy , or zz is utilized to represent an individual set. The algorithm is described in an outline form as follows:

Phase I. Initialization

Step 1. Let $\Omega = 0$ and $\theta = 1$. Check if parts are available in the preprocess area and formulate part sets at the initial time t_0 . Let

$$g_{q,p}(t_0) = \begin{cases} 1, & \text{part } p \text{ is in set } q \text{ at time } t_0, q = 1, 2, \dots, M; \\ 0, & \text{else.} \end{cases} \quad (1)$$

$$E_q(t_0) = \{p | g_{q,p}(t_0) = 1 \wedge O_{p,1} = q\}, q = 1, 2, \dots, M.$$

Step 2. Check if $\theta \geq \eta$. If it is, then go to Step 3. Else

$$\hat{p}(q) = \left\{ p | p \in E_q(t_0) \wedge a_p = \min_{p \in E_q(t_0)} \{a_p\} \right\}, q = 1, 2, \dots, M. \quad (2)$$

Input $\hat{p}(q)$. Let $\delta_{\hat{p}(q)} = 0$. Let $g_{q,\hat{p}(q)}(t_0) = 0$ and update $E_q(t_0)$, $q = 1, 2, \dots, M$, $\theta = \theta + 1$. Repeat this step.

Step 3. Let $\theta = 1$ and $t = t_0$. Check if parts are available in the preprocess area and formulate the part set x at time t . Let

$$g_{x,p}(t) = \begin{cases} 1, & \text{part } p \text{ is in set } x \text{ at time } t; \\ 0, & \text{else.} \end{cases}$$

$$E_x(t) = \{p | g_{x,p}(t) = 1\}.$$

Phase II. Iteration

Step 1. Check if $t \geq T$. If it is, then stop; else check if $E_x(t) = \phi$. If it is, then stop; else check until $\exists p, \delta_p = 1$. Let $t = c_p$.

Step 2. Collect the current workload of process stations at time t , $\gamma_j(t)$, $j \in J$ and obtain

Table 1 Notation utilized in the algorithm

Symbol	Definition
p	Part index
j	Process station index
k	Operation index
I	Integer set, $I = \{\dots, -1, 0, 1, \dots\}$
M	Total process stations
J	Process station set, $J = \{1, 2, \dots, M\}$
T	Production cycle
t	Current time
t_0	Initial time
η	Average number of parts at each process station
a_p	Arrival time of part p
c_p	Completion time of part p
θ	Iteration index
q	Set indicator
$E_q(t)$	Set q at time t
$O_{p,k}$	Process station of operation k of part p
σ_p	Total time of operations of part p
S	Size of the preprocess area, $S > 1$
ξ	Weight
$\gamma_j(t)$	Workload of process station j at time t
\hat{j}	Process station having the least $\gamma_j(t)$
\tilde{j}	Process station having the second least $\gamma_j(t)$
P	Domain
λ_x	Simple function
λ_y	Simple function
λ'	Segmented function
$\bar{\lambda}$	Transformed function
$\tilde{\lambda}$	Weighted function
$\hat{\lambda}$	Overall function
$\lambda_x(p)$	Elements in range of λ_x
$\lambda_y(p)$	Elements in range of λ_y
$\lambda_q(p)$	Elements in range of λ'
$\bar{\lambda}_q(p)$	Elements in range of $\bar{\lambda}$
$\tilde{\lambda}_q(p)$	Elements in range of $\tilde{\lambda}$
$\hat{\lambda}_q(p)$	Elements in range of $\hat{\lambda}$
$\hat{p}(q)$	Part having the earliest a_p in set q
$\tilde{p}(q)$	Part having the minimal σ_p in set q
$p^*(q)$	Part corresponding to the minimal element of $\hat{\lambda}$ and having the earliest a_p in set q
$g_{q,p}(t)$	Status of part p in set q at time t , such that $g_{q,p}(t) = \begin{cases} 1, & \text{part } p \text{ is in set } q \text{ at time } t; \\ 0, & \text{else.} \end{cases}$

Table 1 continued

Symbol	Definition
δ_p	Status of part processing, such that $\delta_p = \begin{cases} 1, & \text{part } p \text{ finishes all operations;} \\ 0, & \text{else.} \end{cases}$
Ω	Status of obtaining elements in ranges of set functions, such that $\Omega = \begin{cases} 1, & \text{elements are obtained;} \\ 0, & \text{else.} \end{cases}$

$$\begin{aligned} \hat{j} &= \left\{ j \mid j \in J \wedge \gamma_j(t) = \min_{j \in J} \{ \gamma_j(t) \} \right\}, \\ \tilde{j} &= \left\{ j \mid j \in J \wedge \gamma_j(t) = \min_{j \in J, j \neq \hat{j}} \{ \gamma_j(t) \} \right\}. \end{aligned} \tag{3}$$

Step 3. Check if parts are available in the preprocess area and formulate the part set x at time t . Let

$$\begin{aligned} g_{x,p}(t) &= \begin{cases} 1, & \text{part } p \text{ is in set } x \text{ at time } t; \\ 0, & \text{else.} \end{cases} \\ E_x(t) &= \{ p \mid g_{x,p}(t) = 1 \}. \end{aligned} \tag{4}$$

Step 4. Classify the parts in $E_x(t)$ into subsets and obtain $E_y(t)$ and $E_z(t)$ such that $E_y(t) \cup E_z(t) = E_x(t)$. Let

$$\begin{aligned} g_{q,p}(t) &= \begin{cases} 1, & \text{part } p \text{ is in set } q \text{ at time } t, q = y, z; \\ 0, & \text{else.} \end{cases} \\ E_y(t) &= \{ p \mid p \in E_x(t) \wedge g_{y,p}(t) = 1 \wedge O_{p,1} = \hat{j} \}, \\ E_z(t) &= \{ p \mid p \in E_x(t) \wedge g_{z,p}(t) = 1 \wedge O_{p,1} \neq \hat{j} \}. \end{aligned} \tag{5}$$

Step 5. Classify the parts in $E_y(t)$ into subsets and obtain $E_{yy}(t)$ and $E_{yz}(t)$ such that $E_{yy}(t) \cup E_{yz}(t) = E_y(t)$. Classify the parts in $E_z(t)$ into subsets and obtain $E_{zy}(t)$ and $E_{zz}(t)$ such that $E_{zy}(t) \cup E_{zz}(t) = E_z(t)$. Let

$$\begin{aligned} g_{q,p}(t) &= \begin{cases} 1, & \text{part } p \text{ is in set } q \text{ at time } t, q = yy, yz, zy, zz; \\ 0, & \text{else.} \end{cases} \\ E_{yy}(t) &= \{ p \mid p \in E_y(t) \wedge g_{yy,p}(t) = 1 \wedge O_{p,1} = \hat{j} \wedge O_{p,2} = \tilde{j} \}, \\ E_{yz}(t) &= \{ p \mid p \in E_y(t) \wedge g_{yz,p}(t) = 1 \wedge O_{p,1} = \hat{j} \wedge O_{p,2} \neq \tilde{j} \}, \\ E_{zy}(t) &= \{ p \mid p \in E_z(t) \wedge g_{zy,p}(t) = 1 \wedge O_{p,1} \neq \hat{j} \wedge O_{p,1} = \tilde{j} \}, \\ E_{zz}(t) &= \{ p \mid p \in E_z(t) \wedge g_{zz,p}(t) = 1 \wedge O_{p,1} \neq \hat{j} \wedge O_{p,1} \neq \tilde{j} \}. \end{aligned} \tag{6}$$

Step 6. Check if $\Omega = 1$. If it is, then let $\theta = 1$ and go to Step 7; else check if $E_q(t) = \phi, q = x, y, yy, yz, zy, zz$. If it is, then let $\Omega = 1$ and go to Step 3 of this phase. Else

$$\tilde{p}(q) = \left\{ p \mid p \in E_q(t) \wedge \sigma_p = \min_{p \in E_q(t)} \{ \sigma_p \} \right\}, \tag{7}$$

$\lambda_q(\tilde{p}(q)) = \theta, \theta = \theta + 1$. Let $g_{q,\tilde{p}(q)}(t) = 0$ and update $E_q(t)$. Repeat this step.

Step 7. Obtain set functions as follows, which are called as the simple functions

$$\begin{aligned} \lambda_x: E_x(t) &\rightarrow I, \lambda_x = \{ (p, \lambda_x(p)) \mid p \in E_x(t) \}; \\ \lambda_y: E_y(t) &\rightarrow I, \lambda_y = \{ (p, \lambda_y(p)) \mid p \in E_y(t) \}. \end{aligned} \tag{8}$$

Obtain the segmented function as follows, which is also a set function

$$P = \{ p \mid p \in E_q(t), q = yy, yz, zy, zz \}, \tag{9}$$

$$\lambda': P \rightarrow I, \lambda' = \{ (p, \lambda_q(p)) \mid p \in P \}. \tag{10}$$

Step 8. Obtain the transformed function

$$\bar{\lambda}: P \rightarrow I, \bar{\lambda} = \{ (p, \bar{\lambda}_q(p)) \mid p \in P \}. \tag{11}$$

$$\bar{\lambda}_q(p) = \begin{cases} \bar{\lambda}_{yy}(p) = \lambda_y(p) + \lambda_{yy}(p), & p \in E_{yy}(t); \\ \bar{\lambda}_{yz}(p) = \lambda_y(p) + \lambda_{yz}(p), & p \in E_{yz}(t); \\ \bar{\lambda}_{zy}(p) = \lambda_z(p) + \lambda_{zy}(p), & p \in E_{zy}(t); \\ \bar{\lambda}_{zz}(p) = \lambda_z(p) + \lambda_{zz}(p), & p \in E_{zz}(t). \end{cases}$$

Let $\lambda_z(p) = 0, p \in E_z(t); \lambda_{yz}(p) = 0, p \in E_{yz}(t); \lambda_{zz}(p) = 0, p \in E_{zz}(t)$. Then

$$\bar{\lambda}_q(p) = \begin{cases} \lambda_y(p) + \lambda_{yy}(p), & p \in E_{yy}(t); \\ \lambda_y(p), & p \in E_{yz}(t); \\ \lambda_{zy}(p), & p \in E_{zy}(t); \\ 0, & p \in E_{zz}(t). \end{cases} \tag{12}$$

Step 9. Add the weight ξ to the transformed function to obtain the weighted function

$$\tilde{\lambda}: P \rightarrow I, \tilde{\lambda} = \left\{ (p, \tilde{\lambda}_q(p)) \mid p \in P \right\}, \tilde{\lambda}_q(p) = \bar{\lambda}_q(p) + \xi S, \tag{13}$$

$$\text{where } \xi = \begin{cases} -5, & q = yy; \\ -3, & q = yz; \\ -1, & q = zy; \\ 0, & q = zz. \end{cases} \tag{14}$$

Step 10. Obtain the overall function

$$\hat{\lambda}: P \rightarrow I, \hat{\lambda} = \left\{ (p, \hat{\lambda}_q(p)) \mid p \in P \right\}, \hat{\lambda}_q(p) = \lambda_x(p) + \tilde{\lambda}_q(p). \tag{15}$$

Step 11. Obtain

$$p^*(x) = \left\{ p \mid p \in E_x(t) \wedge a_p = \min_{p \in E_x(t)} \left\{ a_p \mid \hat{\lambda}_q(p) = \min_{p \in E_x(t)} \{ \hat{\lambda}_q(p) \} \right\} \right\}. \tag{16}$$

Input part $p^*(x)$. Let $\delta_{p^*(x)} = 0$ and $\Omega = 0$. Let $g_{x,p^*(x)}(t) = 0$ and update $E_x(t)$. Go to Step 1 of this phase.

Theoretical analysis

To analyze the effectiveness of the proposed algorithm, the concepts and approaches of discrete mathematics are applied and the properties of the set functions are discussed. Here, $m, n, s,$ or e is utilized to represent an individual part. First, $\lambda_x(p)$ and $\lambda_y(p)$ are the elements in the ranges of the simple functions λ_x and λ_y and have the property of an ascending order of σ_p . Proposition 1 describes this.

Proposition 1 $\forall m, n \in E_x(t) \neq \phi, \sigma_m \leq \sigma_n \Leftrightarrow \lambda_x(m) \leq \lambda_x(n); \forall m, n \in E_y(t) \neq \phi, \sigma_m \leq \sigma_n \Leftrightarrow \lambda_y(m) \leq \lambda_y(n).$

Proof The elements $\lambda_x(p)$ are obtained by the recursive applications of formula (7) and the assignments of integers started with 1 and increased by 1 to each $\lambda_x(\tilde{p}(x))$. Hence an ascending order of σ_p is obtained. Therefore, $\forall m, n \in E_x(t) \neq \phi,$ if $\sigma_m \leq \sigma_n,$ then $\lambda_x(m) \leq \lambda_x(n).$ Vice versa, if $\lambda_x(m) \leq \lambda_x(n),$ then $\sigma_m \leq \sigma_n.$ Therefore, $\sigma_m \leq \sigma_n \Leftrightarrow \lambda_x(m) \leq \lambda_x(n), \forall m, n \in E_x(t) \neq \phi.$ Similarly, it can be obtained that $\sigma_m \leq \sigma_n \Leftrightarrow \lambda_y(m) \leq \lambda_y(n), \forall m, n \in E_y(t) \neq \phi.$ \square

$\lambda_q(p), q = yy, yz, zy, zz$ are the elements in the range of the segmented function λ' and have the property of an ascending order of σ_p in each subset. Proposition 2 depicts this.

Proposition 2 $\forall m, n \in E_q(t) \neq \phi, q = yy, yz, zy, zz, \sigma_m \leq \sigma_n \Leftrightarrow \lambda_q(m) \leq \lambda_q(n).$

Proof The elements $\lambda_q(p)$ are obtained by the recursive applications of formula (7) and the assignments of integers started with 1 and increased by 1 to each $\lambda_q(\tilde{p}(q))$. Hence an ascending order of σ_p is obtained. Therefore, $\forall m, n \in E_q(t) \neq \phi, q = yy, yz, zy, zz.$ If $\sigma_m \leq \sigma_n$ then $\lambda_q(m) \leq \lambda_q(n).$ Vice versa, if $\lambda_q(m) \leq \lambda_q(n),$ then $\sigma_m \leq \sigma_n.$ Therefore, $\sigma_m \leq \sigma_n \Leftrightarrow \lambda_q(m) \leq \lambda_q(n), \forall m, n \in E_q(t) \neq \phi, q = yy, yz, zy, zz.$ \square

Theorem 1 *Formula (15) is valid in obtaining the overall function $\hat{\lambda}.$*

Proof $E_y(t) \cup E_z(t) = E_x(t)$ according to Step 4. Similarly according to Step 5, $E_{yy}(t) \cup E_{yz}(t) = E_y(t)$ and $E_{zy}(t) \cup E_{zz}(t) = E_z(t).$ $\therefore E_{yy}(t) \cup E_{yz}(t) \cup E_{zy}(t) \cup E_{zz}(t) = E_x(t).$ According to formula (9),

$$\begin{aligned} P &= \{p|p \in E_q(t), q = yy, yz, zy, zz\} \\ &= \{p|p \in E_{yy}(t) \vee p \in E_{yz}(t) \vee p \in E_{zy}(t) \vee p \in E_{zz}(t)\} \\ &= \{p|p \in E_{yy}(t)\} \cup \{p|p \in E_{yz}(t)\} \cup \{p|p \in E_{zy}(t)\} \cup \{p|p \in E_{zz}(t)\} \\ &= E_{yy}(t) \cup E_{yz}(t) \cup E_{zy}(t) \cup E_{zz}(t). \end{aligned}$$

$\therefore E_x(t) = P.$ According to formulae (13) and (15), $\hat{\lambda}: P \rightarrow I, \hat{\lambda} = \left\{ \left(p, \hat{\lambda}_q(p) \right) | p \in P \right\}, \hat{\lambda}_q(p) = \lambda_x(p) + \tilde{\lambda}_q(p) = \lambda_x(p) + \bar{\lambda}_q(p) + \xi_S. \lambda_x(p)$ are the elements in the range of $\lambda_x.$ $\bar{\lambda}_q(p)$ are the elements in the range of $\bar{\lambda}.$ S and ξ are constants with values of integers. The domain of $\bar{\lambda}$ is $P.$ Because the domains of λ_x and $\bar{\lambda}$ are the same, formula (15) is valid in obtaining the overall function $\hat{\lambda}.$ \square

$\bar{\lambda}_q(p), q = yy, yz, zy, zz$ are the elements in the range of the transformed function $\bar{\lambda}$ and have the property of an ascending order of σ_p in each subset. Theorem 2 describes this.

Theorem 2 $\forall m, n \in E_q(t) \neq \phi, q = yy, yz, zy, zz, \sigma_m \leq \sigma_n \Leftrightarrow \bar{\lambda}_q(m) \leq \bar{\lambda}_q(n)$.

Proof According to formula (12), $\bar{\lambda}_{yy}(p) = \lambda_y(p) + \lambda_{yy}(p), p \in E_{yy}(t); \bar{\lambda}_{yz}(p) = \lambda_y(p), p \in E_{yz}(t); \bar{\lambda}_{zy}(p) = \lambda_{zy}(p), p \in E_{zy}(t); \bar{\lambda}_{zz}(p) = 0, p \in E_{zz}(t)$. According to Proposition 1, $\forall m, n \in E_y(t) \neq \phi, \sigma_m \leq \sigma_n \Leftrightarrow \lambda_y(m) \leq \lambda_y(n)$. According to Proposition 2, $\forall m, n \in E_q(t) \neq \phi, q = yy, yz, zy, zz, \sigma_m \leq \sigma_n \Leftrightarrow \lambda_q(m) \leq \lambda_q(n)$. If $\sigma_m \leq \sigma_n, \bar{\lambda}_{yy}(m) - \bar{\lambda}_{yy}(n) = \lambda_y(m) - \lambda_y(n) + \lambda_{yy}(m) - \lambda_{yy}(n) \leq 0, m, n \in E_{yy}(t)$. On the other hand, if $\bar{\lambda}_{yy}(m) \leq \bar{\lambda}_{yy}(n), m, n \in E_{yy}(t)$, then $\lambda_y(m) - \lambda_y(n) + \lambda_{yy}(m) - \lambda_{yy}(n) \leq 0, m, n \in E_{yy}(t)$. Different conditions make the above inequality hold:

- (1) $\lambda_y(m) - \lambda_y(n) \leq 0$, and $|\lambda_{yy}(m) - \lambda_{yy}(n)| \leq |\lambda_y(m) - \lambda_y(n)|$;
- (2) $\lambda_{yy}(m) - \lambda_{yy}(n) \leq 0$, and $|\lambda_y(m) - \lambda_y(n)| \leq |\lambda_{yy}(m) - \lambda_{yy}(n)|$;
- (3) $\lambda_y(m) - \lambda_y(n) \leq 0$, and $\lambda_{yy}(m) - \lambda_{yy}(n) \leq 0$.

Therefore, $\sigma_m \leq \sigma_n$ according to Proposition 1 for condition (1); $\sigma_m \leq \sigma_n$ according to Proposition 2 for condition (2); $\sigma_m \leq \sigma_n$ according to Propositions 1 and 2 for condition (3). $\therefore \sigma_m \leq \sigma_n \Leftrightarrow \bar{\lambda}_{yy}(m) \leq \bar{\lambda}_{yy}(n), m, n \in E_{yy}(t)$. Similarly, $\sigma_m \leq \sigma_n \Leftrightarrow \bar{\lambda}_{yz}(m) \leq \bar{\lambda}_{yz}(n), m, n \in E_{yz}(t); \sigma_m \leq \sigma_n \Leftrightarrow \bar{\lambda}_{zy}(m) \leq \bar{\lambda}_{zy}(n), m, n \in E_{zy}(t); \sigma_m \leq \sigma_n \Leftrightarrow \bar{\lambda}_{zz}(m) \leq \bar{\lambda}_{zz}(n), m, n \in E_{zz}(t)$. $\therefore \forall m, n \in E_q(t) \neq \phi, q = yy, yz, zy, zz, \sigma_m \leq \sigma_n \Leftrightarrow \bar{\lambda}_q(m) \leq \bar{\lambda}_q(n)$. □

$\hat{\lambda}_q(p), q = yy, yz, zy, zz$ are the elements in the range of the overall function $\hat{\lambda}$ and have the property of an ascending order of σ_p in each subset. Theorem 3 depicts this.

Theorem 3 $\forall m, n \in E_q(t) \neq \phi, q = yy, yz, zy, zz, \sigma_m \leq \sigma_n \Leftrightarrow \hat{\lambda}_q(m) \leq \hat{\lambda}_q(n)$.

Proof According to formulae (13) and (15),

$$\begin{aligned} \hat{\lambda}_{yy}(p) &= \lambda_x(p) + \bar{\lambda}_{yy}(p) + \xi S, \quad p \in E_{yy}(t); \quad \hat{\lambda}_{yz}(p) = \lambda_x(p) + \bar{\lambda}_{yz}(p) + \xi S, \quad p \in E_{yz}(t); \\ \hat{\lambda}_{zy}(p) &= \lambda_x(p) + \bar{\lambda}_{zy}(p) + \xi S, \quad p \in E_{zy}(t); \quad \hat{\lambda}_{zz}(p) = \lambda_x(p) + \bar{\lambda}_{zz}(p) + \xi S, \quad p \in E_{zz}(t). \end{aligned}$$

According to Proposition 1, $\forall m, n \in E_x(t) \neq \phi, \sigma_m \leq \sigma_n \Leftrightarrow \lambda_x(m) \leq \lambda_x(n)$. According to Theorem 2, $\forall m, n \in E_q(t) \neq \phi, q = yy, yz, zy, zz, \sigma_m \leq \sigma_n \Leftrightarrow \bar{\lambda}_q(m) \leq \bar{\lambda}_q(n)$. Therefore, if $\sigma_m \leq \sigma_n, \hat{\lambda}_{yy}(m) - \hat{\lambda}_{yy}(n) = \lambda_x(m) - \lambda_x(n) + \bar{\lambda}_{yy}(m) - \bar{\lambda}_{yy}(n) \leq 0, m, n \in E_{yy}(t)$. On the other hand, if $\hat{\lambda}_{yy}(m) \leq \hat{\lambda}_{yy}(n), m, n \in E_{yy}(t)$, then $\lambda_x(m) - \lambda_x(n) + \bar{\lambda}_{yy}(m) - \bar{\lambda}_{yy}(n) \leq 0, m, n \in E_{yy}(t)$. Different conditions make the above inequality hold:

- (1) $\lambda_x(m) - \lambda_x(n) \leq 0$, and $|\bar{\lambda}_{yy}(m) - \bar{\lambda}_{yy}(n)| \leq |\lambda_x(m) - \lambda_x(n)|$;
- (2) $\bar{\lambda}_{yy}(m) - \bar{\lambda}_{yy}(n) \leq 0$, and $|\lambda_x(m) - \lambda_x(n)| \leq |\bar{\lambda}_{yy}(m) - \bar{\lambda}_{yy}(n)|$;
- (3) $\lambda_x(m) - \lambda_x(n) \leq 0$, and $\bar{\lambda}_{yy}(m) - \bar{\lambda}_{yy}(n) \leq 0$.

Therefore, $\sigma_m \leq \sigma_n$ according to Proposition 1 for condition (1); $\sigma_m \leq \sigma_n$ according to Theorem 2 for condition (2); and $\sigma_m \leq \sigma_n$ according to Proposition 1 and Theorem 2 for condition (3). $\therefore \sigma_m \leq \sigma_n \Leftrightarrow \hat{\lambda}_{yy}(m) \leq \hat{\lambda}_{yy}(n), m, n \in E_{yy}(t)$. Similarly, $\sigma_m \leq \sigma_n \Leftrightarrow \hat{\lambda}_{yz}(m) \leq \hat{\lambda}_{yz}(n), m, n \in E_{yz}(t); \sigma_m \leq \sigma_n \Leftrightarrow \hat{\lambda}_{zy}(m) \leq \hat{\lambda}_{zy}(n), m, n \in E_{zy}(t); \sigma_m \leq \sigma_n \Leftrightarrow \hat{\lambda}_{zz}(m) \leq \hat{\lambda}_{zz}(n), m, n \in E_{zz}(t)$. $\forall m, n \in E_q(t) \neq \phi, q = yy, yz, zy, zz, \sigma_m \leq \sigma_n \Leftrightarrow \hat{\lambda}_q(m) \leq \hat{\lambda}_q(n)$. □

A minimal $\hat{\lambda}_q(p)$ corresponding to the parts in subset $E_{yy}(t)$ or $E_{yz}(t)$ is less than that corresponding to the parts in subset $E_{zy}(t)$ or $E_{zz}(t)$. Theorem 4 explains this.

Theorem 4 If $m \in E_{yy}(t), n \in E_{yz}(t), s \in E_{zy}(t), e \in E_{zz}(t)$,

$$\begin{aligned} \hat{\lambda}_{yy}(m) &= \min \left\{ \hat{\lambda}_{yy}(p), p \in E_{yy}(t) \right\}, & \hat{\lambda}_{yz}(n) &= \min \left\{ \hat{\lambda}_{yz}(p), p \in E_{yz}(t) \right\}, \\ \hat{\lambda}_{zy}(s) &= \min \left\{ \hat{\lambda}_{zy}(p), p \in E_{zy}(t) \right\}, & \hat{\lambda}_{zz}(e) &= \min \left\{ \hat{\lambda}_{zz}(p), p \in E_{zz}(t) \right\}, \end{aligned}$$

then $\hat{\lambda}_{yy}(m) < \hat{\lambda}_{zy}(s), \hat{\lambda}_{yy}(m) < \hat{\lambda}_{zz}(e), \hat{\lambda}_{yz}(n) < \hat{\lambda}_{zy}(s), \hat{\lambda}_{yz}(n) < \hat{\lambda}_{zz}(e)$.

Proof It can be obtained, according to formulae (11–15), that

$$\begin{aligned} \hat{\lambda}_{yy}(p) &= \lambda_x(p) + \lambda_y(p) + \lambda_{yy}(p) - 5S, & p \in E_{yy}(t); & \hat{\lambda}_{yz}(p) = \lambda_x(p) + \lambda_y(p) - 3S, \\ p \in E_{yz}(t); & \hat{\lambda}_{zy}(p) = \lambda_x(p) + \lambda_{zy}(p) - S, & p \in E_{zy}(t); & \hat{\lambda}_{zz}(p) = \lambda_x(p), & p \in E_{zz}(t). \end{aligned}$$

Therefore, $\hat{\lambda}_{yy}(m) - \hat{\lambda}_{zy}(s) = \lambda_x(m) + \lambda_y(m) + \lambda_{yy}(m) - 5S - \lambda_x(s) - \lambda_{zy}(s) + S$. Because the elements in the ranges of both the simple functions and the segmented function are obtained by arranging parts in ascending orders of σ_p in different subsets and assigning integers started with 1 and increased by 1 to them, $1 \leq \lambda_x(m) \leq S, 1 \leq \lambda_x(s) \leq S, 1 \leq \lambda_y(m) \leq S, 1 \leq \lambda_{yy}(m) \leq S, 1 \leq \lambda_{zy}(s) \leq S, S > 1$. $\hat{\lambda}_{yy}(m) - \hat{\lambda}_{zy}(s) \leq -S - 2 < 0, \therefore \hat{\lambda}_{yy}(m) < \hat{\lambda}_{zy}(s)$. Similarly, it can be obtained that $\hat{\lambda}_{yy}(m) - \hat{\lambda}_{zz}(e) = \lambda_x(m) + \lambda_y(m) + \lambda_{yy}(m) - 5S - \lambda_x(e) \leq -2S - 1 < 0, \therefore \hat{\lambda}_{yy}(m) < \hat{\lambda}_{zz}(e)$. Similar to the above, it can be obtained as well that $\hat{\lambda}_{yz}(n) - \hat{\lambda}_{zy}(s) = \lambda_x(n) + \lambda_y(n) - 3S - \lambda_x(s) - \lambda_{zy}(s) + S \leq -2, \therefore \hat{\lambda}_{yz}(n) < \hat{\lambda}_{zy}(s); \hat{\lambda}_{yz}(n) - \hat{\lambda}_{zz}(e) = \lambda_x(n) + \lambda_y(n) - 3S - \lambda_x(e) \leq -S - 1 < 0, \therefore \hat{\lambda}_{yz}(n) < \hat{\lambda}_{zz}(e)$. □

Theorem 5 Formula (16) is valid in obtaining the selected part.

Proof $\hat{\lambda}_q(p)$ in formula (16) are the elements in the range of the overall function $\hat{\lambda}$. The proof is similar to that of Theorem 1, $E_x(t) = P$. Then, $\min_{p \in E_x(t)} \left\{ \hat{\lambda}_q(p) \right\}$ is valid in obtaining the minimal element in the range of the overall function $\hat{\lambda}$ for parts in $E_x(t)$. Similarly, $\min_{p \in E_x(t)} \{a_p\}$ obtains the earliest arrival time for parts in $E_x(t)$. Therefore, formula (16) is valid in identifying the part that corresponds to the minimal $\hat{\lambda}_q(p)$ and has the earliest arrival time among the parts corresponding to the minimal $\hat{\lambda}_q(p)$. □

In summary, according to Theorem 4, a minimal $\hat{\lambda}_q(p)$ corresponding to the parts in subset $E_{yy}(t)$ or $E_{yz}(t)$ is less than that corresponding to the parts in the other subsets. Both of the subsets possess the parts that best balance the workload. Therefore, if a part in subset $E_{yy}(t)$ or $E_{yz}(t)$ is inputted, the FMS tends to be more balanced. Also, at each input decision point, the algorithm selects and inputs the part corresponding to the minimal $\hat{\lambda}_q(p)$ and having the earliest arrival time among the parts that correspond to the minimal $\hat{\lambda}_q(p)$ according to Theorem 5. In addition, according to Theorem 3, a minimal $\hat{\lambda}_q(p)$ corresponds to the parts having the minimal total time of operations in each subset. This indicates that the part selected and inputted at each input decision point balances the workload the most and has the

minimal total time of operations in the same subset. Therefore, the algorithm utilizes the segmental set functions to dynamically balance workload first, and then apply SPT. In other words, the algorithm is effective in dynamic workload balancing.

Application example

The proposed algorithm can be applied to practical situations. It is well recognized in the literature that workload balance can increase productivity in FMSs. Therefore, the proposed algorithm can be used for part input sequencing of the FMSs in the MC environment to increase productivity. It can be applied to FMSs supplying multiple partners in MTO supply chains. It provides a methodology for FMSs to increase productivity and to handle demand fluctuations.

An illustrative example is provided here to explain how the proposed algorithm works. The example is developed by the use of simulation modeling and GPSS simulation software. The simulation of a real FMS in the semiconductor industry by GPSS simulation software was obtained from Department of Industrial Engineering, Texas Tech University and modified for illustrative purposes. A detailed description of the example can be found in He (1996).

The FMS is composed of a loading/unloading station, five process stations, and a material handling robot. The FMS configuration is illustrated in Fig. 1. The robot can only swing in a partial circle. It has to swing backwards at the end of a circle. Each process station provides a local buffer of limited capacity for WIP inventory. Because total parts processed simultaneously in the FMS are limited, parts generally do not exceed the local buffer capacity and there is no blocking of a process station. Researchers have considered dynamic arrival of part types in FMS studies for the machine grouping problem in FMSs (Stecke and Raman 1994), for the scheduling problem in FMSs (Sabuncuoglu 1998), and for the machine loading problem in FMSs (Sabuncuoglu and Lahmar 2003). Here, arriving parts are filled in a preprocess area of the FMS based on their arrival times. Production requirements

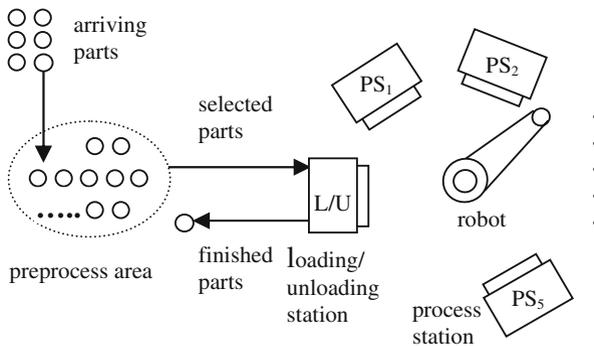


Fig. 1 FMS configuration

are uniformly distributed among all part types. Part routings are random and are predetermined. Production information is listed in Table 2.

First, the algorithm checks available parts in the preprocess area at Step 1 to form the part sets $E_q(t_0)$ so that the same quantity of parts can be initially inputted to each process station as per formula (1). $g_{q,p}(t_0)$ in formula (1) is used to update $E_q(t_0)$. Formula (2) identifies the part having the earliest arriving time for each process station. Four parts are inputted to each process station, resulting in a total of 20 parts initially inputted into the FMS at Step 2. Then, the algorithm updates to the current time, checks available parts in the preprocess area to formulate the part set $E_x(t)$, and turns to the iteration phase at Step 3. The parts of each type inputted are listed at the second column in Table 3. The parts of each type remaining in the preprocess area are also listed in Table 3.

At Step 1 in the iteration phase, the algorithm checks if the production cycle is finished. If so, the algorithm stops; if not, the algorithm checks if there are no parts to be inputted, and if there are no parts, the algorithm stops; if there are parts, the

Table 2 Production information

Part type	Production requirement probability	Routing (process station)				Processing time (s)				Total time of operations (s)
1	0.1	1	2			65	75	140		
2	0.1	2	4			125	145	270		
3	0.1	3					85	85		
4	0.1	2					225	225		
5	0.1	3	5			135	155	290		
6	0.1	1					75	75		
7	0.1	4	2	1		105	85	75	265	
8	0.1	5						355	355	
9	0.1	5	4			255	245	500		
10	0.1	1	2	3	4	15	25	35	45	120

Table 3 Information obtained in the example by applying the proposed algorithm

Part type	Parts input	Parts remain	$\lambda_x(pt)$	$\lambda_y(pt)$	$\lambda_{yz}(pt)$	$\lambda_{zy}(pt)$	$\lambda_{zz}(pt)$
1	1	4	3			2	
2	1	2	6				3
3	1	3	2	1	1		
4	3	1	4				1
5	3	1	7	2	2		
6	1	5	1			1	
7	4	1	5				2
8	3	6	8				4
9	1	7	9				5
10	2	0					

algorithm updates to the current time until a part finishes all its operations and leaves. After a part finishes all its operations, several operations are finished as well. Then, current workload information of the process stations is collected and the least and the second least loaded process stations are identified at Step 2 as per formula (3). The total processing times of the parts inputted are summarized in Table 4. The total processing times of the operations finished are summarized in Table 5. We can see from the tables that currently process station 3 is the least loaded station, having a workload of 290, and that process station 1 is the second least loaded one, having a workload of 390.

Next, at Steps 3, 4, and 5, the algorithm checks available parts in the preprocess area to formulate $E_x(t)$ and classify parts in $E_x(t)$ into different subsets as per

Table 4 Total inputted processing times

Part type	Parts	Process station				
		1	2	3	4	5
1	1	65	75			
2	1		125		145	
3	1			85		
4	3		225			
5	3			135		155
6	1	75				
7	4	75	85		105	
8	3					355
9	1				245	255
10	2	15	25	35	45	
	Total	470	1265	560	900	1785

Table 5 Total finished processing times

Part type	Operations	Process station				
		1	2	3	4	5
1	1	65				
2						
3						
4	1		225			
5	2			135 × 2		
6						
7	2				105 × 2	
8	1					355
9						
10	2	15	25			
	Total	80	250	270	210	355

formulae (4–6). Similar to the above, $g_{q,p}(t)$ in the formulae are used to update $E_q(t)$. The classification is made by three levels of divisions: the division of $E_x(t)$ into subsets to obtain $E_y(t)$ and $E_z(t)$, the division of $E_y(t)$ into subsets to obtain $E_{yy}(t)$ and $E_{yz}(t)$, and the division of $E_z(t)$ into subsets to obtain $E_{zy}(t)$ and $E_{zz}(t)$. Formulae (4–6) describe the sets. The reason for the classification is that different priorities need to be considered for parts to be inputted from different subsets. Also, part input decisions are made dynamically, and different situations may occur at different input decision points, where there are no parts in $E_{yy}(t)$, or there are no parts in both $E_{yy}(t)$ and $E_{yz}(t)$, or others. These steps repeat once more. Part sets of $E_x(t)$, $E_y(t)$, $E_z(t)$, $E_{yy}(t)$, $E_{yz}(t)$, $E_{zy}(t)$, and $E_{zz}(t)$ are formed twice. Ω is used to control the repetition at Step 6. First, the part sets are formed in order to obtain the elements in the ranges of the simple functions and the segmented function at Step 6. The same part sets are formed again as the elements in the domains of the set functions. Because the part sets used to obtain the elements in the domains and in the ranges of the set functions are the same, the set functions obtained are valid. Also, parts in the preprocess area are checked and part sets are formed each iteration in this phase in order to consider newly arriving parts. In the example, the parts in the preprocess area can be classified into different subsets based on their routings and the current workload information as follows: no parts in $E_{yy}(t)$, three type 3 and one type 5 parts in $E_{yz}(t)$, four type 1 and five type 6 parts in $E_{zy}(t)$, and two type 2, one type 4, one type 7, six type 8, and seven type 9 parts in $E_{zz}(t)$. $E_y(t) \cup E_z(t) = E_x(t)$, as described in Step 4. $E_{yy}(t) \cup E_{yz}(t) = E_y(t)$ and $E_{zy}(t) \cup E_{zz}(t) = E_z(t)$, as displayed in Step 5. Therefore, all of the parts are in $E_x(t)$. Among them, type 3 and type 5 parts are in $E_y(t)$. The others are in $E_z(t)$.

At Steps 6 and 7, the algorithm obtains the simple functions and the segmented function as per formulae (7–10). The elements $\lambda_x(p)$, $\lambda_y(p)$, and $\lambda_q(p)$ are obtained by ascending orders of σ_p in the sets $E_x(t)$, $E_y(t)$, $E_{yy}(t)$, $E_{yz}(t)$, $E_{zy}(t)$, and $E_{zz}(t)$ accordingly. The part to be selected and inputted can be identified based on their arriving times among the parts corresponding to the minimal $\hat{\lambda}_q(p)$ according to Theorem 5. The elements $\hat{\lambda}_q(p)$ can be obtained based on current workload information and part processing information that is the same for the same part type. Therefore, the part corresponding to the minimal $\hat{\lambda}_q(p)$ is among the parts of the part type corresponding to the minimal $\hat{\lambda}_q(pt)$, where pt is the part type index. For illustrative purpose, part types instead of parts are utilized to obtain the elements in the ranges of the set functions in the example. The elements $\lambda_x(pt)$ and $\lambda_y(pt)$ are listed at the fourth and fifth columns in Table 3. The elements $\lambda_q(pt)$ in different subsets are also listed in Table 3.

At Steps 8 and 9, the algorithm constructs the transformed function and adds the weight to it to obtain the weighted function as per formulae (11–14). The transformed function is constructed according to the three levels of divisions such that the elements in its range are different in different subsets. $\lambda_y(p)$ is for the first division, $\lambda_{yy}(p)$ is for the second division, and $\lambda_{zy}(p)$ is for the third division. Because the weight for parts in different subsets is either negative integers or zero, it can only result in either reduction or no change in the elements in the range of the weighted function. Also, a smaller element in the range of the weighted function is resulted by a weight of a larger absolute value. At Step 10, the algorithm obtains the

overall function. The elements in the range of the overall function fall into different areas and represent different priorities of parts in different subsets. The elements $\hat{\lambda}_q(pt)$ can be obtained using formulae (11–15) and the following formula can be obtained:

$$\hat{\lambda}_q(pt) = \lambda_x(pt) + \tilde{\lambda}_q(pt) = \lambda_x(pt) + \bar{\lambda}_q(pt) + \zeta S$$

$$= \begin{cases} \lambda_x(pt) + \lambda_y(pt) + \lambda_{yy}(pt) - 5S, & pt \in E_{yy}(t), \\ \lambda_x(pt) + \lambda_y(pt) - 3S, & pt \in E_{yz}(t), \\ \lambda_x(pt) + \lambda_{zy}(pt) - S, & pt \in E_{zy}(t), \\ \lambda_x(pt), & pt \in E_{zz}(t). \end{cases}$$

If the size of the preprocess area is 50, the elements $\hat{\lambda}_q(pt)$ for the part types in the preprocess area can be calculated by using the obtained elements in the ranges of the simple functions and the segmented function according to the above formula:

$$\begin{aligned} \hat{\lambda}_{yz}(3) &= \lambda_x(3) + \lambda_y(3) - 3 \times 50 = 2 + 1 - 150 = -147, \\ \hat{\lambda}_{yz}(5) &= \lambda_x(5) + \lambda_y(5) - 3 \times 50 = 7 + 2 - 150 = -141, \\ \hat{\lambda}_{zy}(1) &= \lambda_x(1) + \lambda_{zy}(1) - 50 = 3 + 2 - 50 = -45, \\ \hat{\lambda}_{zy}(6) &= \lambda_x(6) + \lambda_{zy}(6) - 50 = 1 + 1 - 50 = -48, \quad \hat{\lambda}_{zz}(2) = \lambda_x(2) = 6, \\ \hat{\lambda}_{zz}(4) &= \lambda_x(4) = 4, \quad \hat{\lambda}_{zz}(7) = \lambda_x(7) = 5, \quad \hat{\lambda}_{zz}(8) = \lambda_x(8) = 8, \quad \hat{\lambda}_{zz}(9) = \lambda_x(9) = 9. \end{aligned}$$

Part type 3 corresponds to the minimal $\hat{\lambda}_q(pt)$. Therefore, a part of type 3 corresponds to the minimal $\hat{\lambda}_q(p)$. In this example, there are three type 3 parts and the algorithm selects and inputs the one having the earliest arriving time among them at Step 11. After selecting and inputting the part, the algorithm updates its status and goes to Step 1 of this phase for iteration. The algorithm repeats this phase until the production cycle is finished or there are no parts to be inputted.

In this example, the part selected and inputted based on $\hat{\lambda}_q(pt)$ is a part of type 3. Part type 3 has its first operation at process station 3, which has the least workload at the current input decision point and therefore balances the workload the most. Its total time of operations is 85, less than 290 for part type 5 in the same subset.

Conclusions

In the fiercely competitive global arena, increasing attention is being paid to customer requirements. Customer orders are becoming more frequent, and order sizes are decreasing. Enterprises have to provide quick response to customer orders and produce customized products in a cost-effective manner. Mass customization is adopted to meet these challenges.

This paper proposes a dynamic heuristic-based algorithm for the FMS part input sequencing problem in the MC environment. The proposed algorithm utilizes the segmental set functions to apply the strategy of dynamic workload balancing, and the SPT scheduling rule. Theoretical analysis is performed. The algorithm is effective in dynamic workload balancing. Productivity of FMSs can be increased by the use of the proposed algorithm because it is recognized in the literature that

workload balance can increase productivity of FMSs. Therefore, the proposed algorithm can be used to operate FMSs cost-effectively for increasing the MC capability of FMSs.

Our investigation provides insights into practical applications. FMS managers can utilize the proposed algorithm for part input sequencing of FMSs to increase their MC capability in a complex and dynamic environment. The algorithm can be applied to FMSs supplying multiple partners in MTO supply chains that are confronted by demand fluctuations. The algorithm provides a pragmatic approach for FMSs to increase productivity and handle demand fluctuations so as to reduce inventories.

In summary, the paper contributes in several ways. It proposes an applicable and effective approach for the FMS part input sequencing problem in the MC environment. It establishes the segmental set functions to apply the strategy of dynamic workload balancing, and the SPT scheduling rule for this *NP*-hard problem. It explores the application of the concepts and approaches of discrete mathematics to analyze the effectiveness of the algorithm in dynamic workload balancing in a complex and dynamic environment. It provides an example to illustrate the application of the algorithm and points out its potential applications to practical situations.

Performance improvement in operation management in MC environments needs much attention and further study. A suggestion for further research could be the development of effective algorithms to consider delivery requirements of customer orders as well to improve both productivity- and due-date-based performance of FMSs.

Acknowledgements The authors are grateful to the Editor, Professor Kathryn Stecke, and the Associate Editor of International Journal of Flexible Manufacturing Systems for their deep insights and valuable comments and suggestions, which helped to improve this paper. The authors would also like to thank the reviewers for their comments and suggestions.

References

- Choi RH, Malstrom EM (1988) Evaluation of traditional work scheduling rules in a flexible manufacturing system with a physical simulator. *J Manuf Syst* 7:33–45
- He Y (1996) Part selection in a dynamic job-shop type flexible manufacturing system (FMS). PhD dissertation, Texas Tech University, Lubbock
- Johnsonbaugh R (2004) Discrete mathematics, Chapters 1 and 2. Pearson Education Asia Limited and Publishing House of Electronics Industry
- Kim Y-D, Lee D-H, Yoon C-M (2001) Two stage heuristic algorithms for part input sequencing in flexible manufacturing systems. *Euro J Oper Res* 133:625–635
- Kimura T, Kanda Y (2005) Development of a remote monitoring system for a manufacturing support system for small and medium-sized enterprises. *Comput Ind* 56:3–12
- O'Keefe RM, Rao R (1992) Part input into a flexible flow system: an evaluation of look-ahead simulation and a fuzzy rule base. *Int J Flex Manuf Syst* 4:113–127
- Pine II BJ, Victor B, Boynton AC (1993) Making mass customization work. *Har Bus Rev* September–October:108–119
- Sabuncuoglu I (1998) A study of scheduling rules of flexible manufacturing systems: a simulation approach. *Int J Prod Res* 36(2):527–546

- Sabuncuoglu I, Lahmar M (2003) An evaluation study of operation grouping policies in an FMS. *Int J Flex Manuf Syst* 15:217–239
- Shapiro BP, Rangan VK, Sviokla JJ (2004) Staple yourself to an order. *Har Bus Rev* July–August:162–171
- Stecke KE (1983) Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. *Manag Sci* 29(3):273–288
- Stecke KE (1992) Procedures to determine part mix ratios for independent demands in flexible manufacturing systems. *IEEE Trans Eng Manag* 39(4):359–369
- Stecke KE, Morin TL (1985) The optimality of balancing workloads in certain types of flexible manufacturing systems. *Euro J Oper Res* 20:68–82
- Stecke KE, Kim I (1988) A study of FMS part type selection approaches for short-term production planning. *Int J Flex Manuf Syst* 1(1):7–29
- Stecke KE, Kim I (1991) A flexible approach to part type selection in flexible flow systems using part mix ratios. *Int J Prod Res* 29(1):53–75
- Stecke KE, Toczylowski E (1992) Profit-based FMS dynamic part type selection over time for mid-term production planning. *Euro J Oper Res* 63:54–65
- Stecke KE, Raman N (1994) Production planning decisions in flexible manufacturing systems with random material flows. *IIE Trans* 26(5):2–17
- Stecke KE, Smith TM (1996) On the robustness of using balanced part mix ratios to determine cyclic part input sequences into flexible flow systems. *Int J Prod Res* 34:2925–2941

Author Biographies

Yumin He is currently Associate Professor in Department of Information System, School of Economics and Management, Beihang University, Beijing, P.R. China. She received Ph.D. degree in Industrial Engineering from Texas Tech University in 1996. Her research interests include FMS scheduling, enterprise application integration, information resource management, project management, mass customization, and database technology.

Milton L. Smith, Ph.D., is currently Professor of Industrial Engineering at Texas Tech University and Director of the Texas Tech University off-campus site in Amarillo. He has research interests in production systems and in practical applications of industrial engineering/operations research in nontraditional areas. Specific interests include production scheduling, simulation of manufacturing operations, and project management. He has a broad range of experience in developing simulation models for analysis of automated production facilities; also he has research interests in homeland security topics.