Available online at www.sciencedirect.com
SCIENCE DIRECT
International Journal of Machine Tools & Manufacture 45 (2005) 949–957
INTERNATIONAL JOURNAL OF MACHINE TOOLS & MANUFACTURE DESIGN, RESEARCH AND APPLICATION
www.elsevier.com/locate/ijmactool

A novel approach to production planning of flexible manufacturing systems using an efficient multi-objective genetic algorithm

Jian-Hung Chen[a], Shinn-Ying Ho[b,*]

[a]Department of Information Engineering and Computer Science, Feng Chia University, 100 Wen-hwa Road, Taichung 407, Taiwan, ROC
[b]Department of Biological Science and Technology, Institute of Bioinformatics, National Chiao Tung University, Hsinchu 300, Taiwan, ROC

Received 11 May 2004; accepted 20 October 2004
Available online 19 December 2004

Abstract

In this paper, a novel approach using an efficient multi-objective genetic algorithm EMOGA is proposed to solve the problems of production planning of flexible manufacturing systems (FMSs) having four objectives: minimizing total flow time, machine workload unbalance, greatest machine workload and total tool cost. EMOGA makes use of Pareto dominance relationship to solve the problems without using relative preferences of multiple objectives. High efficiency of EMOGA arises from that multiple objectives can be optimized simultaneously without using heuristics and a set of good non-dominated solutions can be obtained providing additional degrees of freedom for the exploitation of resources of FMSs. Experimental results demonstrate effectiveness of the proposed approach using EMOGA for production planning of FMSs.

Keywords: Flexible manufacturing system; Multi-objective optimization; Genetic algorithm; Production planning

1. Introduction

A flexible manufacturing system (FMS) is a production system consisting of a set of identical and/or complementary numerically controlled machines which are connected through an automated guided vehicle (AGV) system. Since FMS is capable of producing a variety of part types and handling flexible routing of parts instead of running parts in a straight line through machines, FMS gives great advantages through its flexibility such as dealing with machine and tool breakdowns, changes in schedule, product mix, and alternative routes. Flexible manufacturing is of increasing importance in advancing factory automation that keeps a manufacturer in a competitive edge.

While FMS offers many strategic and operational benefits over conventional manufacturing systems, its efficient management requires solutions to complex product planning problems with multiple objectives and constraints. The aim of production planning is to develop a cost effective and operative production plan over planning phases. Decisions regarding production planning problems have to be made before the start of actual production, and consist of organizing the limited production resource constraints efficiently. Generally, production planning of FMSs consists of many optimization problems, such as routing optimization, equipment optimization and machine optimization [1].

During the past decades, a number of production planning approaches have been developed for automated planning and increased efficiency of production planning [1]. Many approaches usually optimize a single objective and treat other objectives as constraints [1,2]. However, it is known that many problems in production planning of FMSs are multi-objective optimization problems (MOOPs) in nature [1,2]. From a system designer's point of view, it is very desirable to obtain a set of non-dominated solutions providing the flexibility of reconfigurable manufacturing via simultaneously considering all the objectives. Recently, some approaches [3–9] have been proposed to deal with MOOPs in production planning. They can be classified into two categories:

* Corresponding author. Tel.: +886 3 5712121 ext. 56905; fax: +886 3 5729288.
E-mail address: syho@mail.nctu.edu.tw (S.-Y. Ho).

0890-6955/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijmachtools.2004.10.010

(1) *Decomposition approach* [3–6]. Problem is decomposed into several sub-problems according to its characteristics. Hereafter, the sub-problems are solved in multiple stages. A solution of a sub-problem is usually used as an initial solution of its succeeding sub-problem. The advantage of the decomposition approach is that heuristics of a specific objective can be utilized separately. However, the decomposition of problems requires prior domain knowledge, and the final solution is sensitive to the solution of previous stages. Chen and Askin [3] solved a multi-objective machine loading problem sequentially by heuristic algorithms. Kumar et al. [4] proposed a min–max approach to solving a grouping and loading problem in multiple stages. Liang [5] proposed a two-stage approach to jointly solving part selection, machine loading and machining speed selection problems. Lee et al. [6] proposed a two-stage approach to solving an operation sequence and tool selection problem.
(2) *Preference-based approach* [7–9]. Given relative preferences to each individual objective, the preference-based approach generally combines multiple objectives into a single objective function using a weighted linear combination of all objectives, and then a single-objective optimization algorithm is used to find a single solution at a time. The main advantage of the preference-based approach is that a suitable non-dominated solution can be easily obtained. However, relative preferences require prior domain knowledge and the solution quality is sensitive to the relative preferences used [10]. Liang and Dutta [7] proposed a mixed-integer programming approach to solving a machine loading and process planning problem by aggregating the makespan and manufacturing costs of the problems into a single objective function. Sodhi et al. [8] proposed a heuristic algorithm to solving a multi-period tool and production problem by aggregating the resource costs of the problems into an overall function. Swarnkar and Tiwari [9] aggregated two objective functions of a bicriteria machine loading problem into a single function and employed a hybrid algorithm based on tabu search and simulated annealing to solve the problem.

Multi-objective evolutionary algorithms (MOEAs) have been recognized to be well-suited for solving MOOPs because their abilities to exploit and explore multiple solutions in parallel and to find a widespread set of non-dominated solutions in a single run [10]. Several MOEAs based on Pareto dominance relationship [11] are proposed to solve MOOPs directly, and present more promising results than single-objective optimization techniques theoretically and empirically [10,12]. By making use of Pareto dominance relationship, MOEAs are capable of performing the fitness assignment of multiple objectives without using relative preferences of multiple objectives. Thus, all the objective functions can be optimized simultaneously. As a result, MOEA seems to be an alternative approach to solving production planning problems on the assumption that no prior domain knowledge is available.

In this paper, a novel approach using an efficient multi-objective genetic algorithm EMOGA is proposed to solve multi-objective production planning problems (MOPPPs) having four objectives: minimizing total flow time, machine workload unbalance, greatest machine workload and total tool cost. The fundamental difference of the proposed approach from the above-mentioned decomposition and preference-based approaches is that the problem decomposition and relative preferences are not necessary. In addition, the proposed approach can obtain a set of non-dominated solutions for decision makers in a single run. Decision makers can easily distinguish between the costs of different production plans and choose more than one satisfactory production plans at a time. Six benchmark problems with different complexities are derived to evaluate the performance of the proposed approach. An efficient multi-objective evolutionary algorithm SPEA [12], which outperforms many existing MOEAs, is used for performance comparisons. It is shown empirically that EMOGA can converge to better solutions than SPEA in solving MOPPPs.

This paper is organized as follows: Section 2 describes the investigated problem MOPPP. Section 3 presents the efficient multi-objective genetic algorithm EMOGA for solving MOPPPs. Section 4 gives the experimental results and analysis of the proposed algorithm. Section 5 summarizes our conclusions.

2. Problem statement

In this paper, we focus on *operation flexibility* in the production planning phase of FMSs. Operation flexibility is concerned with an operation which can be performed on alternative machines with different processing time, transportation time and resource costs [1]. Therefore, optimizations on routing, machine and equipment are essential for operation flexibility. With the assignment of operations to machines, four optimization objectives: minimizing total flow time, machine workload unbalance, greatest machine workload and total tool cost, are considered in our problems.

2.1. The FMS environment

An FMS consists of a set of identical and/or complementary numerically controlled machines and tool systems. All components are connected through an AGV system. Fig. 1 shows the layout of a simple FMS with several machines, AGVs and a tool system.

In order to design the production planning of FMSs, the environment within which the FMS under consideration operates can be described below.

(1) The term *machine* is to describe a machine cell. A machine cell consists of several identical

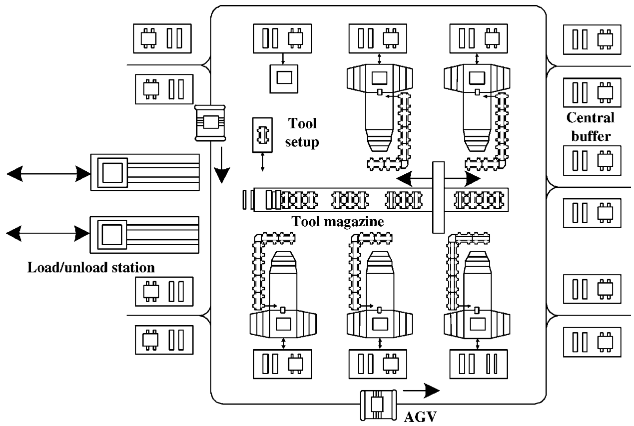

Fig. 1. FMS with several machines, AGVs and a central tool magazine.

devices/machines. The types and number of machines are known. There is a sufficient input/output buffer space at each machine.

(2) A *part* type requires a number of *operations*. A number of part types will be manufactured simultaneously in batches. Parts can choose one or more machines at each of their operation stages, and the transportation of the parts within different machines is handled by an AGV system.

(3) A machine can perform several types of operations, and an operation call be performed on alternative machines.

(4) A machine can only process an operation at one time. Operations to be performed in the machine are non-preemptive. Operation lot splitting is ignored in this paper.

(5) A *production plan* consists of part indices, operation indices, and a series of machine indices corresponding to operations of all parts. Based on a production plan, each operation is operated on its corresponding machine. An illustrative production plan of 3 parts and 10 operations is presented in Fig. 2, and the operations are operated on 3 different machines. An example of the series of machine indices to be optimized is $Y=[1\ 1\ 1\ 3\ 1\ 2\ 2\ 2\ 3\ 3]$.

(6) The tool costs of operations in machines are known. Processing times of operations in machines are available and deterministic.

(7) Workload on each machine is contributed by those operations assigned to a machine.

Part index	1	2	3
Operation index	1 2 3 4	1 2 3	1 2 3
Machine index	1 1 1 3	1 2 2	2 3 3

Fig. 2. A production plan of 3 parts and 10 operations, operated on 3 different machines. For example, the operation 4 of the part 1 is assigned to the machine 3.

(8) A load/unload (L/U) station serves as a distribution center for parts not yet processed and as a collection center for parts finished. All vehicles start from the L/U station initially and return to there after accomplishing all their assignments. There are sufficient input/output buffer spaces at the L/U station.

(9) The number of AGVs is given and the transportation time of AGVs are known. Some machines may not be linked.

(10) AGVs carry a limited number of products at a time. They move along predetermined paths, with the assumption of no delay because of congestion. Preemption of trips is not allowed.

(11) It is assumed that all the design, layout and set-up issues within FMS have already been resolved.

(12) Real-time issues, such as traffic control, congestion, machine failure or downtime, scraps, rework, and vehicle dispatches for battery changer are ignored here and left as issues to be considered during real-time control.

2.2. Mathematical formulation of MOPPPs

2.2.1. Notations

In order to formulate MOPPPs, the following notations are introduced:

- i: part index, $i=1,2,3,\ldots,I$.
- j: operation index for part i, $j=1,2,3,\ldots,J_i$.
- k, l: machine index k, $l=1,2,3,\ldots,K$.
- Y: a series of machine indices corresponding to operations of all parts in a production plan.
- pv_i: production volume (unit) for part i.
- pt_{ijk}: processing time per unit to perform operation j of part i using machine k.
- m_k: maximum workload of machine k.
- tw_k: workload in machine k, $tw_k=pt_{ijk}\times pv_i$.
- rtw_k: workload ratio in machine k, $rtw_k=\frac{tw_k}{m_k}$.
- ew: average workload of machines
- $s_{ikl}:\begin{cases}1, & \text{if part } i \text{ is to transfer from machine } k \text{ to } l;\\ 0, & \text{otherwise.}\end{cases}$
- $x_{ijk}:\begin{cases}1, & \text{if machine } k \text{ is selected to perform operation } j \text{ of part } i;\\ 0, & \text{otherwise.}\end{cases}$
- abl: available capacity of AGV per trip, abl is set to 10 in this paper.
- n_{ikl}: the number of trips between machine k and l for part i,

$$n_{ikl} = s_{ikl} \times \left\lceil \frac{pv_i}{abl} \right\rceil,$$

where the bracket represents a ceiling operation.

- tm_{kl}: transportation time from machine k to l. If machines k and l are not linked, it is set to be a negative value for constraint handling.
- t_{ikl}: total transportation time between machines k and l for part i,

$$t_{ikl} = n_{ikl} \times tm_{kl}.$$

- c_{ijk}: tool costs to perform operation j of part i using machine k.

2.2.2. Objectives

There are four objectives to be optimized in FMSs according to the suggestion of Tempelmeier and Kuhn [1], described below.

(1) Minimization of total flow time. This objective is to minimize the processing time and transportation time for producing the parts. The total machine processing time (f_1) is defined as Eq. (1), the transportation time (f_2) is defined as Eq. (2), and the total flow time (F_1) is defined as Eq. (3). Transportation between unlinked machines are penalized in f_2

$$f_1 = \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{k=1}^{K} pv_i \times pt_{ijk} \times x_{ijk}, \tag{1}$$

$$f_2 = \sum_{i=1}^{I} \sum_{j=1}^{J_i-1} \sum_{k=1}^{K} \sum_{l=1}^{K} t_{ikl} \times x_{ijk} \times x_{i(j+1)l}, \tag{2}$$

$$F_1 = f_1 + f_2. \tag{3}$$

(2) Minimization of machine workload unbalance. Balancing the machine workload can avoid creating bottleneck machines. The objective function (F_2) is defined as Eq. (4)

$$F_2 = \sum_{k=1}^{K} (rtw_k - ew)^2. \tag{4}$$

(3) Minimization of greatest machine workload. Pursuing this objective also implies attempting to minimize the total flow time. The objective function (F_3) is defined as Eq. (5)

$$F_3 = \max(rtw_k). \tag{5}$$

(4) Minimization of total tool cost. Tool costs consider the consumptions of tools, tool life issues, tool expenses and the number of tool copies. The objective function (F_4) is defined as Eq. (6)

$$F_4 = \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{k=1}^{K} c_{ijk} \times x_{ijk}. \tag{6}$$

2.2.3. Multi-objective mathematical model

The overall multi-objective mathematical model of MOPPPs can be formulated as follows. Given the production volume pv_i, the processing time pt_{ijk}, the maximum workload m_k, the available capacity of AGV per trip abl, the transportation time tm_{kl} and the tool costs c_{ijk}, find a series of machine indices, Y, for operations of all parts such that

$$\text{minimize} \quad F_1, F_2, F_3, F_4, \tag{7}$$

subject to

$$\sum_{k=1}^{K} x_{ijk} = 1, \quad \forall (i,j), \tag{8}$$

$$tm_{kl} \geq 0, \quad \forall (k,l), \tag{9}$$

$$rtw_k \leq 1, \quad \forall i. \tag{10}$$

The constraint, Eq. (8), ensures that only one machine is selected for each operation of a part. Eq. (9) ensures an AGV path exists between machines k and l. Eq. (10) is to ensure the machine workload tw_k is smaller or equal to its maximum machine workload m_k.

If the total number of machines is x and the total number of operations is y, then the complexity of the investigated problem is $O(x^y)$.

2.2.4. An illustrative example

An illustrative MOPPP *m3o10* with $I=3$ parts, $K=3$ machines, and 10 operations $(J_1, J_2, J_3)=(4, 3, 3)$ is presented. Table 1 shows the processing time, tool costs of 10 different operations on 3 different machines and production volumes of 3 parts. Transportation time of machine k to machine l is given in Table 2. The maximum machine workload is $m_k=1000$ for each machine in this

Table 1
Processing time, tool costs of 10 different operations to 3 machines and production volume of 3 parts

Operation index		Part 1				Part 2			Part 3		
		1	2	3	4	1	2	3	1	2	3
Pt_{ijk}	Machine 1	1	3	3	5	9	2	9	7	8	7
	Machine 2	7	5	4	6	4	1	4	1	6	2
	Machine 3	6	9	5	1	2	5	1	3	3	5
C_{ijk}	Machine 1	1	2	1	6	1	8	4	8	3	6
	Machine 2	2	3	7	5	9	2	5	9	8	5
	Machine 3	4	5	4	2	8	7	8	9	6	2
Pv_i		51				39			23		

Table 2
Transporation time tm_{kl} of the illustrative MOPPP *m3o10*

K\l	Machine 1	Machine 2	Machine 3
Machine 1	4	11	17
Machine 2	11	3	9
Machine 3	7	18	5

problem. The complexity of *m3o10* problem is 3^{10}. Taking the production plan with a series of machine indices $Y = [1\ 1\ 1\ 3\ 1\ 2\ 2\ 2\ 3\ 3]$ in Fig. 2 as an example, its corresponding objective function values can be calculated as follows:

$$rtw_1 = (51 \times (1 + 3 + 3) + 39 \times 9)/1000 = 0.708,$$

$$rtw_2 = (39 \times (1 + 4) + 23 \times 1)/1000 = 0.218,$$

$$rtw_3 = 51 \times 1 + 23 \times (3 + 5))/1000 = 0.235,$$

$$ew = 0.708 + 0.218 + 0.235)/3 = 0.387,$$

$$f_1 = 51 \times (1 + 3 + 3 + 1) + 39 \times (9 + 1 + 1) + 23 \times (1 + 3 + 5) = 1161,$$

$$f_2 = \left\lceil \frac{51}{10} \right\rceil \times (4 + 4 + 17) + \left\lceil \frac{39}{10} \right\rceil \times (11 + 3) + \left\lceil \frac{23}{10} \right\rceil \times (9 + 5) = 248,$$

$$F_1 = f_1 + f_2 = 1161 + 248 = 1409,$$

$$F_2 = (0.708 - 0.387)^2 + (0.218 - 0.387)^2 + (0.235 - 0.387)^2 = 0.154706,$$

$$F_3 = \max\{0.708, 0.218, 0.235\} = 0.708,$$

$$F_4 = (1 + 2 + 1 + 2) + (1 + 2 + 5) + (9 + 6 + 2) = 31.$$

3. Efficient multi-objective genetic algorithm EMOGA

EMOGA differs from conventional genetic algorithms (GAs) [13] only in the fitness assignment strategy and the elitism strategy in the selection step of EMOGA. A summary of Pareto dominance relationship and the fitness assignment strategy for handling multiple objective functions is described in Section 3.1. EMOGA for solving MOPPPs is presented in Section 3.2, including the representation of chromosomes, genetic operators, constraint handling, and the procedure of EMOGA.

3.1. Fitness assignment strategy

Many MOEAs differ mainly in the fitness assignment strategy which is known as an important issue in solving MOOPs [10]. EMOGA uses a generalized Pareto-based scale-independent fitness function GPSIFF considering the quantitative fitness values in Pareto space for both dominated and non-dominated individuals. GPSIFF makes the best use of Pareto dominance relationship to evaluate individuals using a single measure of performance.

3.1.1. Pareto dominance relationship

Assume all the objective functions F_m are to be minimized. Mathematically, MOOPs can be represented as the following vector mathematical programming problems:

$$\text{Minimize} \quad F(Y) = \{F_1(Y), F_2(Y), \ldots, F_m(Y)\}, \tag{11}$$

where Y denotes a solution and $F_m(Y)$ is generally a nonlinear objective function. When the following inequalities hold between two solutions Y_1 and Y_2, Y_2 is a *non-dominated solution* and is said to *dominate* $Y_1 (Y_2 \succ Y_1)$:

$$\forall m : F_m(Y_1) \geq F_m(Y_2) \text{ and } \exists\, n : F_n(Y_1) > F_n(Y_2). \tag{12}$$

When the following inequality holds between two solutions Y_1 and Y_2, Y_2 is said to *weakly dominate* $Y_1 (Y_2 \succcurlyeq Y_1)$:

$$\forall m : F_m(Y_1) \geq F_m(Y_2). \tag{13}$$

A feasible solution Y^* is said to be a *Pareto-optimal solution* if and only if there does not exist a feasible solution Y where Y dominates Y^*. The corresponding vector of Pareto-optimal solutions is called *Pareto-optimal front*. An example in a bicriteria space is shown in Fig. 3, where the circles represent non-dominated solutions and the black dots are dominated solutions.

3.1.2. GPSIFF

The used GPSIFF is described below. Let the fitness value of an individual Y be a tournament-like score obtained from all participant individuals by the following

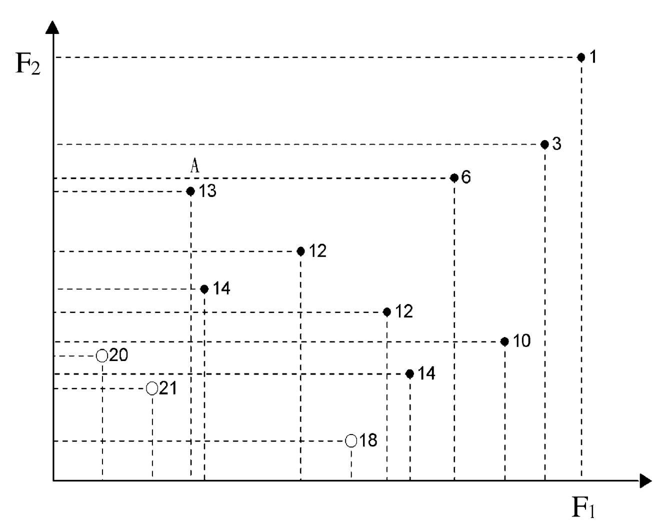

Fig. 3. The circles represent non-dominated solutions and the black dots are dominated solutions. The fitness values are calculated by GPSIFF.

function:

$$F(Y) = p - q + c, \tag{14}$$

where p is the number of individuals which can be dominated by the individuals Y, and q is the number of individuals which can dominate the individual Y in the objective space. Generally, a constant c can be optionally added in the fitness function to make fitness values positive. In this paper, c is the number of all participant individuals.

GPSIFF uses a pure Pareto-ranking fitness assignment strategy, which differs from the traditional Pareto-ranking methods, such as non-dominated sorting [13] and Zitzler and Thiele's method [12]. GPSIFF can assign discriminative fitness values not only to non-dominated individuals but also to dominated ones. Fig. 3 illustrates an example of fitness values of 12 participant individuals for a bicriteria optimization problem ($c=12$). For example, considering the individual A with a fitness value 13, in the rectangle formed by A, two individuals dominates A ($q=2$) and three individuals is dominated by A ($p=3$). Therefore, the fitness value of A is $3-2+12=13$.

3.2. EMOGA for solving MOPPP

3.2.1. Chromosome representation

A series of machine indices Y for operations of all parts is directly encoded as a chromosome with integer-valued genes. In the chromosome, each gene with the range [1, K] stands for a machine index. A chromosome is also called as an *individual* in GAs.

3.2.2. Genetic operators

The genetic operators used in the proposed approach are widely used in literature. The selection operator of EMOGA uses a binary tournament selection which works as follows. Choose two individuals randomly from the population and copy the better individual into the intermediate population.

Crossover is a recombination process in which genes from two selected parents are recombined to generate offspring chromosomes. The single-point crossover is used in EMOGA. In a single-point crossover operation, a cutting point is selected randomly, and the genes on the sides of the cutting point are exchanged between the parent chromosomes. A crossover operation is illustrated as follows. Suppose two chromosomes $Y_1=[1\ 1\ 1\ 3\ 1\ 2\ 2]$ and $Y_2=[2\ 2\ 3\ 1\ 1\ 2\ 2]$ are selected as parents, each chromosome has seven genes (i.e. seven operations). Assuming the generated cutting point is 2, then the following $C_1=[\underline{2\ 2}\ 1\ 3\ 1\ 2\ 2]$ and $C_2=[1\ 1\ \underline{3\ 1\ 1\ 2\ 2}]$ are generated.

A simple mutation operator is used to alter genes. For each gene, randomly generate a real value from the range [0, 1]. If the value is smaller than the mutation probability p_m, replace its machine index with an integer randomly generated from the range [1, K].

3.2.3. Constraint handling

Based on the proposed chromosome representation, Eq. (8) is always satisfied. If Eq. (9) is violated, the transportation time between machines k and l, tm_{kl}, is set to be a large value, 10^7. In this way, f_2 will be penalized. For each machine k, if Eq. (10) is not satisfied, one is added to r_{twk}, as follows:

$$r_{twk} = \begin{cases} \dfrac{tw_k}{m_k}, & \text{if } tw_k \le m_k; \\ \dfrac{tw_k}{m_k} + 1, & \text{otherwise.} \end{cases} \tag{15}$$

3.2.4. Efficient multi-objective genetic algorithm

Since it has been recognized that the incorporation of elitism may be useful in maintaining diversity and improving the performance of multi-objective EAs [10], EMOGA selects a number of elitists from an elite set E in the selection step. The elite set E with capacity $E_{\max}$ maintains the best non-dominated solutions generated so far. In addition, an external set $\bar{E}$ with no capacity is used to store all the non-dominated solutions ever generated so far. The procedure of EMOGA is written as follows:

Step 1 (Initialization) Randomly generate in initial population of N_{pop} individuals and create two empty elite sets E, $\bar{E}$ and an empty temporary elite set E'.

Step 2 (Evaluation) For each individual Y in the population, compute $F_1(Y)$, $F_2(Y)$, $F_3(Y)$, and $F_4(Y)$.

Step 3 (Fitness assignment) Assign each individual a fitness value by using GPSIFF.

Step 4 (Update elite sets) Add the non-dominated individuals in both the population and E' to E, and empty E'. Considering all individuals in E, remove the dominated ones in E. Add E to $\bar{E}$, remove the dominated ones in $\bar{E}$. If the number of non-dominated individuals in E is larger than $E_{\max}$, randomly discard excess individuals.

Step 5 (Selection) Select $N_{\text{pop}}-N_{ps}$ individuals from the population using the binary tournament selection and randomly select N_{ps} individuals from E to form a new population, where $N_{ps}=N_{\text{pop}}\times p_s$ and p_s is a selection proportion. If N_{ps} is greater than the number N_E of individuals in E, let $N_{ps}=N_E$.

Step 6 (Recombination) Perform the single-point crossover operation with a recombination probability p_c.

Step 7 (Mutation) Apply the mutation operator to each gene in the individuals with a mutation probability p_m.

Step 8 (Termination test) If a stopping condition is satisfied, stop the algorithm and output $\bar{E}$ Otherwise, go to Step 2.

Table 3
The parameter settings of EMOGA and SPEA

Algorithm	Parameters	m3o10	m4o10	m5o100	m5o200	m10o100	m10o200
EMOGA	N_{pop}	100	100	200	200	200	200
	E_{max}	100	100	200	200	200	200
	p_s	0.25	0.25	0.25	0.25	0.25	0.25
SPEA	N_{pop}	100	100	200	200	200	200
	E_{max}	25	25	50	50	50	50
EMOGA and	p_c	0.6	0.6	0.6	0.6	0.6	0.6
SPEA	p_m	0.05	0.05	0.05	0.05	0.05	0.05
	N_{eval}	2000	2000	20,000	20,000	40,000	80,000

4. Results and discussion

Considering the real manufacturing environment, we derived the AGV transportation time matrix and six benchmark problems: *m3o10*, *m4o20*, *m5o100*, *m5o200*, *m10o100* and *m10o200*, where *mxoy* stands for the *x* machine and *y* operation problem. In order to further investigate the performance of EMOGA, SPEA [12] is also implemented to solve MOPPPs. The solutions obtained by SPEA are used as the baseline performance for comparisons. The parameter settings of EMOGA and SPEA are given in Table 3. All the parameters of EMOGA and SPEA in each experiment are the same, and the maximum number of participated elitism in population of EMOGA, $N_{pop}\times p_s$, is identical to that of SPEA. Thirty independent runs were performed per test problems, compared with the same number N_{eval} of function evaluations. The benchmark problems and the experimental results are available in the authors' website.

The coverage metric $C(A,B)$ of two solution sets A and B [12] used to compare the performance of two corresponding algorithms considering the four objectives:

$$C(A,B) = \frac{|\{a \in A, b \in B, a \succcurlyeq b\}|}{|B|}, \tag{16}$$

where $\succcurlyeq$ stands for weakly dominate in Pareto dominance relationship. The value $C(A,B)=1$ means that all individuals in B are weakly dominated by A. On the contrary, $C(A,B)=0$ denotes that none of individuals in B is weakly dominated by A. Because the C measure considers the weakly dominance relationship between two sets A and B, $C(A,B)$ is not necessarily equal to $1-C(B,A)$. The comparison results of two solution sets using the coverage metric are depicted using box plots. A box plot provides an excellent visual result of a distribution. The box stretches from the lower hinge (defined as the 25th percentile) to the upper hinge (the 75th percentile) and therefore contains the middle half of the scores in the distribution. The median is shown as a line across the box.

For each run, the solution set of two algorithms are compared using the coverage metric. Fig. 4 depicts the coverage metrics of $C(EMOGA, SPEA)$ and $C(SPEA, EMOGA)$ from 30 runs. In solving the small problem *m3o10*, Fig. 4 shows that the performance of EMOGA is slightly better than SPEA. The average $C(EMOGA, SPEA)=0.7290$, and $C(SPEA, EMOGA)=0.6183$. However, as the complexity of problems increases, Fig. 4 shows that 60%–80% of the non-dominated solutions obtained by SPEA are weakly dominated by the non-dominated solutions obtained by EMOGA in solving the problems *m4o20*, *m5o100*, *m5o200* and *m10o00*. On the contrast, the non-dominated solutions of SPEA dominate nearly 5% of the non-dominated solutions obtained by EMOGA. The results indicate that EMOGA can converge to better solutions more quickly than SPEA. Regarding the large problem *m10o200* with the search space 10^{200}, the non-dominated solutions of EMOGA only dominate 35% of the non-dominated solutions obtained by SPEA. The non-dominated solutions of SPEA dominate 15% of the non-dominated solutions obtained by EMOGA. It may be due to that the search space is too large, so that some boundary solutions obtained by SPEA are not obtained by EMOGA.

Table 4 depicts the average numbers of non-dominated solutions, averaged from 30 runs of EMOGA and SPEA. Fig. 5 depicts the distribution of non-dominated solutions obtained from a run of EMOGA and SPEA. For visualization, Fig. 6 depicts the projection of Fig. 5 on the selected objectives F_1, F_2 and F_3. The figures show that both EMOGA and SPEA can obtain widespread non-dominated

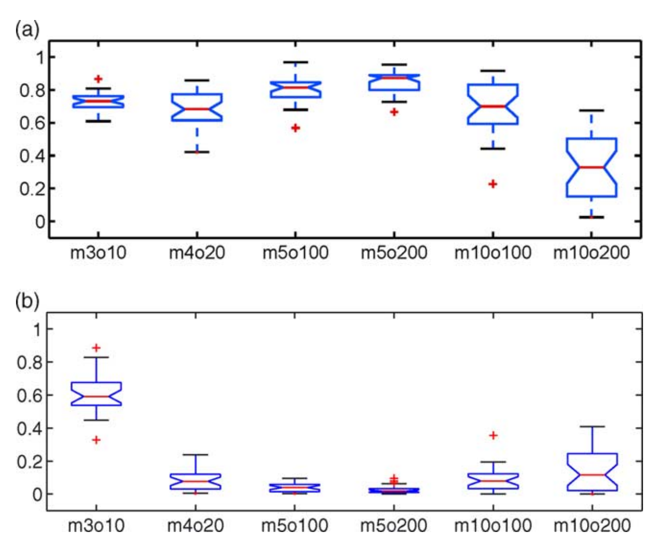

Fig. 4. Box plots based on the cover metric. (a) C(EMOGA, SPEA), (b) C(SPEA, EMOGA).

Table 4
The average numbers of non-dominated solutions obtained by EMOGA and SPEA, averaged from 30 runs

Problem	Algorithm	Average	Std. dev.
m3o10	EMOGA	43.90	6.77
	SPEA	44.03	4.33
m4o20	EMOGA	203.37	19.64
	SPEA	212.03	32.16
m5o100	EMOGA	525.47	46.32
	SPEA	710.80	89.03
m5o200	EMOGA	493.17	42.67
	SPEA	703.73	67.77
m10o100	EMOGA	443.17	64.14
	SPEA	405.23	55.98
m10o200	EMOGA	110.43	33.58
	SPEA	155.57	56.03

solutions on the four objectives. In this run, $C(EMOGA, SPEA)=0.7634$ and $C(SPEA, EMOGA)=0.0851$.

Figs. 5; 6 and Table 4 reveal an important feature of the proposed approach that differed from conventional production planning approaches, that is, a set of non-dominated solutions can be provided for decision makers in a single run. A satisfactory production plan can be fast obtained by given relative preferences from decision makers, and decision makers can also choose several alternative production plans at a time. These additional degrees of freedom could provide a better exploitation of the resources of FMSs. On the contrary, single-objective approaches have to perform multiple runs in order to obtain a set of non-dominated solutions.

5. Conclusions

In this paper, a novel approach to production planning of flexible manufacturing systems (FMSs) using an efficient multi-objective genetic algorithm EMOGA is proposed. The investigated multi-objective production planning problem (MOPPP) has four objectives: minimizing total flow time, machine workload unbalance, greatest machine workload and total tool cost. The advantages of the proposed approach are that EMOGA can optimize multiple objectives without decomposing problems into sub-problems, and EMOGA makes use of Pareto dominance relationship to solve problems without using relative preferences of multiple objectives. While prior domain knowledge for the decomposition of problems or relative preferences of multiple objectives are not available, the proposed approach is an expedient method to solve production planning of FMSs, compared with the decomposition and preference-based approaches.

In addition, the proposed approach can obtain a set of non-dominated solutions for decision makers in a single run. Decision makers can easily distinguish between the costs of different production plans and choose more than one satisfactory production plans at a time. These additional degrees of freedom could provide a better exploitation of the resources of FMSs. Experimental results demonstrated that the quality of non-dominated solutions obtained by EMOGA is better than that of SPEA in terms of convergence speed and accuracy using the same number of function evaluations. The results indicate that the proposed approach is a generalized and efficient approach to solving MOPPPS.

The complexity of the investigated problem is determined by the numbers of operations and machines. If the complexity increases, a large computation time may be necessary to solve a large-scale problem. In practical, EMOGA can utilize some specific heuristic rules (smallest processing time, smallest tool cost, etc.) such that the results may be better. The proposed approach can also be extended to optimize other resource costs related to operation assignment.

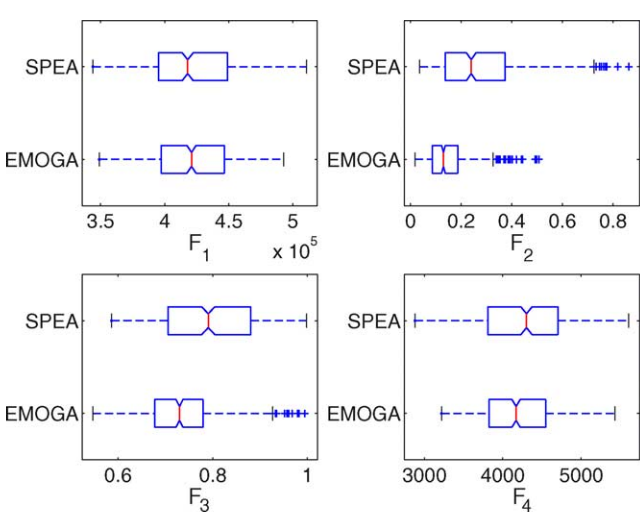

Fig. 5. The distribution of non-dominated solutions from a run of EMOGA and SPEA in solving the m10ol00 problem.

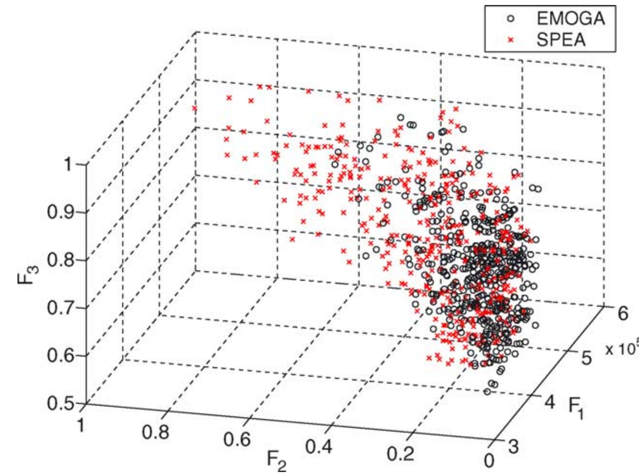

Fig. 6. The projection of non-dominated solutions from a run of EMOGA and SPEA on the objective F_1, F_2 and F_3.

References

[1] H. Tempelmeier, H. Kuhn, Flexible Manufacturing Systems: Decision Support for Design and Operation, Wiley Series in Systems Engineering, Wiley, New York, 1993.

[2] A. Grieco, Q. Semeraro, T. Tolio, A review of different approaches to the FMS loading problem, International Journal of Flexible Manufacturing Systems 13 (4) (2001) 361–384.

[3] Y.-J. Chen, R.G. Askin, A multiobjective evaluation of flexible manufacturing system loading heuristics, International Journal of Production Research 28 (5) (1990) 895–911.

[4] P. Kumar, N.K. Tewari, N. Singh, Joint consideration of grouping and loading problems in a flexible manufacturing system, International Journal of Production Research 28 (7) (1990) 1345–1356.

[5] M. Liang, Integrating machine speed, part selection and machine loading decision in FMSs, Computer and Industrial Engineering 26 (3) (1994) 599–608.

[6] C.S. Lee, S.S. Kim, J.S. Choi, Operation sequence and tool selection in flexible manufacturing systems under dynamic tool allocation, Computers and Industrial Engineering 45 (1) (2003) 61–73.

[7] M. Liang, P.S. Dutta, A mixed-integer programming approach to the machine loading and process planning problem in a process layout environment, International Journal of Production Research 28 (8) (1990) 1471–1484.

[8] M.S. Sodhi, R.G. Askin, S. Sen, Multiperiod tool and production assignment in flexible manufacturing systems, International Journal of Production Research 32 (6) (1994) 1281–1294.

[9] R. Swarnkar, M.K. Tiwari, Modeling machine loading problem of FMSs and its solution methodology using a hybrid tabu search and simulated annealing-based heuristic approach, Robotics and Computer-Integrated Manufacturing 20 (3) (2004) 199–209.

[10] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, Wiley Interscience Series in Systems and Optimization, Wiley, New York, 2001.

[11] V. Pareto, Cours d'economie politique professe a l'universite de Lausanne, vol. 1, F. Rouge, Lausanne, 1896.

[12] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strengthen Pareto approach, IEEE Transaction on Evolutionary Computation 4 (3) (1999) 257–271.

[13] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.