Int J Adv Manuf Technol
DOI 10.1007/s00170-010-3094-4

ORIGINAL ARTICLE

Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects

Yi Sun · Chaoyong Zhang · Liang Gao · Xiaojuan Wang

Received: 8 June 2010 / Accepted: 1 December 2010

Abstract Since multi-objective flow shop scheduling problem (MFSP) plays a key role in practical scheduling, there has been an increasing interest in MFSP according to the literature. However, there still have been wide gaps between theories and practical applications, and the review research of multi-objective optimization algorithms in MFSP (objectives > 2) field is relatively scarce. In view of this, this paper provides a comprehensive review of both former and the state-of-the-art approaches on MFSP. Firstly, we introduce a broad description and the complexity of MFSP. Secondly, a taxonomy of multi-objective optimizations and an analysis of the publications on MFSP are presented. It is noteworthy that heuristic and meta-heuristic methods and hybrid procedures are proven much more useful than other methods in large and complex situations. Finally, future research trends and challenges in this field are proposed and analyzed. Our survey shows that algorithms developed for MFSP continues to attract significant research interest from both theoretical and practical perspectives.

Keywords Decision making · Flow shop · Multi-objective optimization · Scheduling

1 Introduction

The flow shop scheduling problem (FSP) is an important component of scheduling problems. Its importance and practical relevance to industry have attracted researchers to study it from different perspectives for decades. Because of its powerful engineering backgrounds, it is very significant to develop effective algorithms to solve this problem which can enhance production efficiency, improve the optimization of production resources, and increase competitive strength [1].

The majority of the literature of handling FSP has been concentrated on single-criterion scheduling. However, several objectives must be considered in many real-world situations. That is to say, the multi-objective must be satisfied simultaneously over these criteria conflicted with others, and FSP with a single optimization objective is not sufficient for practical applications. It is necessary to search for the trade-off solutions among these objectives, particularly in conflicting situations. Models of multi-objective FSP (MFSP) are established to capture optimization solutions, often called the Pareto optimal solutions, in knotty problems. Besides, there is not even a universally accepted achievement of optimization as that in single-objective optimization. Research on approaches of tackling MFSP is with no doubt a very important and challenging project, and there are still many open questions in this domain.

The conventional approaches to solve single-objective FSP can be mainly categorized into two types, namely, exact and approximation methods. Exact methods, such as enumeration, dynamic programming, branch-and-bound (B&B) method, have been successfully applied to tackle small-sized flow shop problems. However, despite the relative success of exact algorithms, they are still incapable of solving medium and large instances and are too complex for real-world problems. For the medium- and large-scale problems, approximation methods are superior to the exact methods. Actually, multi-objective FSP is encountered much more frequently in real-world engineering applications. In the last decades, there has been an increase in the design of multi-objective programming

Y. Sun · C. Zhang (✉) · L. Gao · X. Wang
The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science & Technology, Wuhan 430074, People's Republic of China
e-mail: zcyhust@mail.hust.edu.cn

Published online: 21 December 2010

techniques to handle FSP with multiple objectives, which can be seen in Fig. 1 (started in 1995). It also suggests that an interesting issue appears in the FSP field. The most successful and popular of the exact methods is the B&B algorithms which use the latest theoretical developments to improve their performances. Approximation methods, especially meta-heuristic methods developed in recent years, are attractive alternatives and are also used to tackle MFSP with Pareto solutions. These efficient meta-heuristic methods mainly include genetic algorithms (GA), particle swarm optimization (PSO), ant colony optimization (ACO), simulated annealing (SA), tabu search (TS), and differential evolution (DE). However, the comprehensive surveys in the literature about methods of dealing with multi-machine flow shop with several objectives, especially in future trends and challenges, are relatively scarce. There have been a few surveys [2–6] about multi-criteria scheduling published in the technical literature. The review studied by Minella et al. [6] is much more comprehensive than others. But it does not discuss further researches and challenges thoroughly, and several other approaches have arisen since the publication of that paper. The intention of the present work was to provide researchers an updated survey and the future research trends of theoretical and practical areas on MFSP.

The remainder of this paper is organized as follows. In Section 2, we introduce the description of MFSP and common measures of performance. In Section 3 and 4, we review the status researches in the development of algorithms on MFSP including the classification and main applications respectively. In Section 5, the trends and challenges of dealing with MFSP research are discussed. At last, some conclusions are made in Section 6.

2 Basic concepts and description

MFSP studies n jobs which are processed on m different machines sequentially. The set of n jobs is $J=\{1, 2,\ldots,n\}$

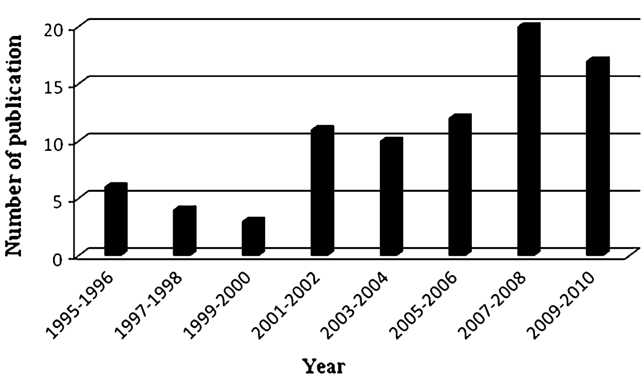

Fig. 1 Number of publications on MFSP

and the set of m machines is $M=\{1, 2,\ldots,m\}$. Machines are available continuously. A job is processed on one machine at a time without preemption, and a machine processes no more than one job at a time. The optimization objectives are ≥2. When the sequence in which the jobs are to be processed is the same for each machine, the scheduling problems are named permutation FSP (PFSP). A hybrid flow shop is a generalization of the flow shop and the parallel machine environments. Instead of m machines in series, there are c stages in series with a number of identical machines in parallel at each stage. Each job has to be processed first at stage 1, then at stage 2, and so on. A stage functions as a bank of parallel machines; at each stage, job j requires processing on only one machine and any machine can be chosen [7].

Scheduling problems can be described by a three-field notation $\alpha|\beta|\gamma$ in [8]. The first field, α, describes the shop (machine) environment. The formula $\alpha=\alpha_1\alpha_2$ is taken here, where α_1 denotes the shop environment and α_2 gives the number of machines. The second field, β, indicates a number of job characteristics such as preemption. The third field, γ, denotes the optimality objectives. The detailed information of the three-field notation, $\alpha|\beta|\gamma$, in scheduling field is shown as follows:

dates

n	Numbers of jobs
r_i	Release date of J_i
d_i	Due date of J_i
s_i	Start time of J_i
p_{ij}	Processing time of J_i operated on machine $j(M_j)$
$\pi_{\mathrm{i,j}}$	Number of J_i on the M_j
$\prod = \left(\pi_{i,j}\right)_{n\times m}$	Sequence of jobs to be processed

Variables

C_i	Completion time of J_i
W_i	Wait time of J_i
L_i	Lateness of J_i
E_i	Earliness of J_i
T_i	Tardiness of J_i
U_i	Flag of tardiness for J_i

Schedule style (α filed)

1	Single
F	Flow shop
J	Job shop
HF	Hybrid flow shop scheduling
PM	Identical parallel machine
QM	Uniform parallel machine
RM	Unrelated machine

Constraints (β field)

prmu	Permutation
r_j	Non-zero release date
SDST	Sequence-dependent setup times
prec	Precedence constraints
lsm	Lot streaming
nwt	No-wait
prmp	Preemption
block	Limited buffers
brkdwn	Breakdowns
M_j	Machine eligibility restrictions
stch	Stochastic
retr	Reentrant
ovlp	Overlapping

Objectives (γ field)

C	Total completion time
F	Total flow time
T	Total tardiness
A	Availability of machine
D	Total delay time
Z_{cost}	Cost of the scheduling
Z_{due}	Cost of due date assignments
Z_{bottle}	Utilization rate of the bottleneck
Z_{pih}	Cost of production and inventory holding
Z_{q}	Degree of satisfaction
$C_{\max}$	Maximum completion
$L_{\max}$	Maximum lateness
$T_{\max}$	Maximum tardiness
$E_{\max}$	Maximum earliness
N_{T}	Total number of tardy jobs
I_{sum}	Total machine idle time
C_{w}	Total weighted completion time
T_{w}	Total weighted tardiness
E_{w}	Total weighted earliness
W_{w}	Total weighted waiting time
C_{ave}	Average completion time
F_{ave}	Average flow time
W_{ave}	Average wait time
$C_{\text{ave,w}}$	Average weighted completion time
$T_{\text{ave,w}}$	Average weighted tardiness

In single-objective optimization, the objective mostly taken into account is the minimization of the maximum completion time, namely, makespan ($C_{\max} = \max\{C_1, C_2, \ldots, C_n\}$). For this objective, PFSP (permutation flow shop scheduling problem) is demonstrated to be strongly NP-hard with more than two machines [9]. The second most studied objective is the minimization of total completion time ($C = \sum_{i=1}^{n} C_i$), which is equal to the total flow time without considering idle times. FSP under this objective is proven to be NP-hard with more than or equal to two machines involved [10]. The minimization of the total tardiness ($T = \sum_{i=1}^{n} T_i$) is the third most studied objective. Just as the other objectives, total tardiness minimization results in a NP-hard problem in the strong sense for more than two machines [11]. The three objectives mentioned above are studied mostly in FSP. Furthermore, there are still some other objectives presented, such as maximum earliness and minimization of the weighted total tardiness.

3 Literature group and analysis

MFSP has received great attention due to its importance in many industrial areas. The methods to solve MFSP can be classified according to decision-making process as follows: *a priori* methods, *a posteriori* methods, and *interactive* methods [12]. In this section, the description of these three kinds of methods is introduced, as well as the applications on MFSP according to particular methods.

3.1 A priori methods

The simplest methods of these approaches are a priori methods which enable the decision maker to intervene before the resolution process. This group of techniques includes those approaches assuming that either certain desired achievable goals or a certain pre-ordering of the objectives can be performed by the decision maker prior to the research. These approaches mainly include lexicographic ordering, linear fitness combination, and nonlinear fitness combination methods [13]. A lexicographic method is described as sorting the objectives according to weightiness and then selecting an objective to be optimized in order, or selecting an objective randomly in each evolutionary step [14]. B&B (e.g., [15]) and other heuristic methods (e.g., [16]) are studied for MFSP in this group. The linear fitness combination approaches give each sub-objective different weight by relative importance and combine them in a linear aggregating function. Thus, a multi-objective flow shop scheduling problem is turned into a single-optimization problem. B&B (e.g. [17]), SA method (e.g., [18]), TS method (e.g., [19]), ACO algorithm (e.g., [20]), PSO approach (e.g., [21]), and other effective strategies (e.g., [22]) are using linear fitness combination techniques to solve MFSP.

The advantages of this group technique are the easy performance and fast implementation in small instance FSP. Whereas, researchers must define a way to gain the weighted value of each objective firstly and only a single solution is returned solution per algorithm run. It is not fit for dealing with MFSP, especially large-scale problems.

3.2 A posteriori methods

The decision-making process comes after completing the search in a posteriori methods. Pareto solutions, obtained by implementing a posteriori approaches, are different from conventional optimization. These techniques do not need priori preference information from the decision maker. The Pareto front solutions obtained are all “best” solutions to the problem in multi-objective cases, and there is no strong dominance relationship among these solutions. From a widespread set of Pareto solutions obtained, administrators will select the only satisfied solution according to the decision maker’s preferences. A posteriori methods are more popular than a priori methods in the literature for engineering applications. They mainly include independent sampling, criterion selection, aggregation selection (linear, nonlinear), and Pareto sampling (Pareto-based selection, Pareto deme-based selection, Pareto elitist-based selection, Pareto rank- and niche-based selection) [13]. The independent sampling methods use a single-objective searching technique to obtain solutions of a multi-objective problem. The weights of sub-objectives are varied during a number of separate multi-objective scheduling method runs. This sort of method is relatively simple and effective in solving MFSP. The number of publications of GA using this technique is the largest (e.g., [23, 24]). The applications using this technique in Pareto sampling techniques are based on Pareto distributive strategy and search all members of non-dominated solutions in each evolutionary run. This sort of method is most popular and effective in solving MFSP. Applications of a posteriori methods in multi-objective scheduling fields are much more plentiful than applications of a priori methods. Besides these mentioned approaches (e.g., B&B, PSO, ACO) in instances of using a priori technique, algorithms like GA [25], DE [26], and IA [27] are proposed in this domain. What is more is that the state-of-the-art theoretical developments are presented to deal with MFSP by adopting posteriori strategies.

3.3 Interactive methods

The interactive approaches are studied less than the other two kinds of approaches. They mainly emphasize the interactive process between solution searching and decision making. Generally speaking, decision makers do not get the relation among the objectives of the problem at the beginning. As for the decision-making process, as well as the observation of intermediate results, decision makers gain a more clear preference and give quantitative descriptions on the objectives in this way. Therefore, these techniques can be summarized as follows. Firstly, find a non-dominated solution using multi-objective algorithms. Then, get the reaction of the consideration of the decision maker about this non-dominated solution and modify the preferences of the objectives accordingly. Lastly, repeat the two previous steps until the decision maker is satisfied or no further improvement is accomplished. Obviously, this method is more appropriate for solving real-world problems. However, few literatures solved the MFSP with the interactive methods being applied due to the complex and difficult nature of its theory. Shi [28] considers that researches about this interactive method should be paid more attention to. We give a simple analysis about the future of this approach in Section 4.2.

The applications of interactive methods in FSP are very scarce. Only some theories and limited literature discussed this group of approaches. Brintrup et al. [29] proposed an interactive genetic algorithm-based framework for optimizing qualitative and quantitative criteria together. An interactive evolutionary multi-objective optimization method is designed using an a priori landscape analysis as a guiding tool in [30]. Kuo [31] proposed an interactive bi-objective programming with valuable trade-off approach to general capacitor placement and scheduling problems. The experiment showed that the proposed method is effective and feasible. The application of interactive methods poses new challenges for devising tackling MFSP and will gain much more attention in the future.

4 Method review in literature

The concise introduction and analysis of each algorithm are discussed in this section. Table 1 gives a summary of these methods in settling MFSP. The first column of the table indicates the handled problems, the second indicates the used methods, and the last column indicates the references, including authors and years. It is obvious that the B&B algorithms in the exact methods are used most successfully, and GA is the most popular meta-heuristic algorithm of approximation methods.

4.1 Branch-and-bound

The B&B method is first proposed by Land and Doig [32]. It is a simple numerical algorithm for the solution of programming problems in which some or all of the variables can take only discrete values. The algorithm requires no special techniques beyond those used in ordinary linear programming. The B&B procedure is valid for small-sized flow shop problems, but it costs much computational time to solve big-sized problems.

Nagar et al. [24] addressed a two-machine FSP with the objective of minimizing a weighted combination of flow time and makespan. Under some strong assumptions and data distributions, they presented a B&B procedure to find the optimal solutions of problem with up to 500 jobs. Liao

Table 1 Summary of applications in MFSP field

Problem	Algorithm	Reference
$F2\|\|(C_{\max},N_T), (C_{\max},T)$	B&B	Liao et al. (1997) [33]
$F2\|\|C_{\max}, F$	B&B	Sayın and Karabat (1999) [35]
$F2\|\|C_{\max}, F$	B&B	Yeh (2001) [38]
$F2\|\|C_{\max}, F$	B&B	Lin and Wu (2006) [39]
$F2\|\|C_{\max}, F$	B&B	Nagar et al. (1995) [2]
$F2\|\|C_{\max}, F$	B&B	Yeh (1999) [36]
$F2\|\|F, T$	B&B	Lee and Wu (2001) [37]
$F2\|prmu\|C_{\max}, F$	B&B, heuristic	Sivrikaya-Serifoğlu and Ulusoy (1998) [34]
$F2\|\|C_{\max}, F$	B&B, heuristics	T'kindt et al. (2003) [15]
$Fm\|prmu\|C_{\max}, T$	B&B, Parallelism	Lemesre et al. (2007) [17]
$Fm\|prmu\|C_{\max}, F$	SA	Suresh and Mohanasundaram (2004) [43]
$F\|prmu\|many$	SA	Loukil et al. (2005) [44]
$HF2(PM)\|SDST\|C,T$	SA	Naderi et al. (2009) [18]
$Fm\|prmu\|C_{\max}, F$	SA	Varadharajan and Rajendran (2005) [45]
$HFm\|\|F_{ave}, T_{\max}$	SA, TS	Hatami et al.(2010) [46]
$Fm\|\|many$	MOTS	Loukil et al. (2000) [48]
$Fm\|prmu\|C_{\max}, T_{\max}$	TS	Armentano and Arroyo (2004) [49]
$F2\|ST\|C,T$	TS, heuristics	Eren and Güner (2006) [19]
$F2\ \|\|C,C_{\max}$	TS, heuristic	Eren and Güner (2008) [50]
$F2\|\|C_w,T, C_{max}$	TS, heuristic, EDD	Eren and Güner (2008) [51]
$Fm\|\|(C_{\max},T), (C_{\max},T,F)$	GA	Murata et al. (1996) [23]
$4F2\|\|C_{\max}, F$	GA	Nagar et al. (1996) [24]
$F2\|\|(C_{\max}, F), (C_{\max},T,F)$	GA	Neppalli et al.(1996) [53]
$Fm\|\|C_{\max}, F,I_{sum}$	GA	Sridhar and Rajendran (1996) [59]
$Fm\|\|C_{\max},T$	GA	Cavalieri and Gaiardelli (1998) [60]
$Fm\|\|(C_{\max},T), (C_{\max}, T, F)$	GA	Ishibuchi and Murata (1998) [25]
$Fm\|\|C_{\max},T$	C-MOGA	Murata et al. (2001) [54]
$Fm\|\|(C_{\max}, T), (C_{\max}, T, F)$	GA	Chang et al. (2002) [56]
$F2\|\|C_{\max}, F$	HGA	Yeh (2002) [61]
$Fm\|prmu\|(C_{\max}, T), (C_{\max}, T, F)$	GA and LS	Ishibuchi et al. (2003) [62]
$Fm\|prmu\|C_{\max}, F$	PGA-ALS	Pasupathy et al. (2006) [65]
$Fm\|\|C_{\max}, I_{sum},C_{ave}$	TSP-GA	Ponnambalam et al. (2004) [63]
$Fm\|\|(C_{\max},T_{\max}), (C_{\max}, T)$	GA	Arroyo and Armentano (2005) [64]
$Fm\|\|C_{max},T$	PGA	Melab et al. (2006) [66]
$Fm\|\|C_{\max},T$	HQGA	Li and Wang (2007) [67]
$F2\|prmu\|C_{\max},C$	SPGA	Chang et al. (2007) [68]
$HFm(PM)\|retr\|Z_{bottle},C_{\max}$	L-NSGA,	Dugardin et al. (2010) [57]
$HFm(PM)\|SDST\|C_{\max}, T_w$	Multi-phase GA	Karimi et al. (2010) [58]
$F2\|\|C_{\max}, F$	ACO	T'kindt et al. (2002) [20]
$Fm\|\|C_{\max}, T, I_{sum}$	ACO	Yagmahan and Yenisey (2008) [75]
$Fm\|ovlp\|T, I_{sum}, W$	ACO	Huang and Yang (2009) [77]
$Fm\|\|C_{\max}, F$	ACO	Yagmahan and Yenisey (2010) [78]
$Fm\|prmu\|C_{\max},T$	ACO	Rajendran and Ziegler (2004) [71]
$Fm\|prmu\|C_{\max}, T_{\max}$	ACO, Path Relinking	Pasia et al. (2006) [74]
$Fm\|ST, lst\|C_{\max}, T$	ACO,TA	Marimuthu et al. (2009) [76]
$Fm\|prmu\|F, I_{sum}$	HPSO	Rahimi-Vahed and Mirghorbani (2007) [82]
$Fm\|prmu\|C_{\max}, T$	DPSO	Guo et al. (2007) [81]
$Fm\|lst\|E_w, T_w$	NBM, DPSO	Tseng and Liao (2008) [21]
$Fm\|prmu\|many$	HPSO	Li et al. (2008) [83]

Table 1 (continued)

Problem	Algorithm	Reference
$Fm\|\|C_{\max}, F_{ave}, I$	PSO	Sha and Lin (2009) [85]
$Fm\|prmu\|C_{\max}, T_{\max}$	MOHDE	Qian et al. (2006) [26]
$Fm\|nwt\|(C_{\max}, T_{\max}),(I_{\mathrm{sum}}, N_{\mathrm{T}})$	MADE	Qian et al. (2009a) [87]
$Fm\|block\|C_{\max}, T_{\max}$	HDE	Qian et al. (2009b) [88]
$Fm\|nwt\|C_{\max}, T_{\max}$	DE	Pan et al. (2009) [89]
$HFm\|prmu\|C_{\max}, F$	QDEA	Zheng andYamashiro (2010) [90]
$Fm\|\|C_{\max},D,W_{\mathrm{ave}},A$	IA	Yang et al. (2002) [92]
$Fm\|nwt\|C_{\mathrm{ave,w}},T_{\mathrm{ave,w}}$	HMOIA	Moghaddam et al. (2007) [27]
$Fm\|nwt\|C_{\mathrm{ave,w}},T_{\mathrm{ave,w}}$	MOIA	Moghaddam et al. (2008) [93]
$Fm\|nwt\|C_{\mathrm{ave,w}}, E_{\mathrm{ave,w}}$	FMOLP	Javadi et al. (2008) [112]
$HFm(PM)\|\|N_T, Z_{\mathrm{pih\_cost}}$	A lexicographic approach	Sawik (2007) [110]
$Fm\|\|C_{\max}, F$	Heuristics	Rajendran (1995) [95]
$F2\|\|C_{\max}, F$	Heuristics	Gupta et al. (2001) [16]
$Fm\|\|C_{\max}, F$	Heuristics	Framinan et al. (2002) [97]
$Fm\|\|C_{\max}, F$	Heuristic	Allahverdi (2003) [98]
$Fm\|\|C_{\max}, T_{\max},F$	Heuristic	Arroyo and Armentano (2004) [100]
$Fm\|\|C_{\max}, T_{\max}$	Heuristic	Allahverdi (2004) [99]
$Fm\|\|C_{\max}, T$	Heuristics	Ravindran et al. (2005) [22]
$Fm\|prum\|C_{\max}/T$	Heuristics	Framinan and Leisten (2006) [101]
$HFm(RM)\|SDST\|C_{\max}, N_{\mathrm{T}}$	Heuristic	Jungwattanakit et al. (2006) [113]
$HFm(PM)\|\|E_{\mathrm{w}}, T_{\mathrm{w}}, W_{\mathrm{w}}$	Meta-heuristics	Janiak (2007) [111]
$HFm(QM)\|SDST\|C_{\max},E,T$	MO-hybrid meta-heuristic	Behnamian et al. (2009) [108]
$HF2\|\|C,F_{\mathrm{ave}}$	forward and backward simulation	Zhang et al. (2002) [106]
$F2\|\|(F,C_{\max}), (T,C_{\max})$	Various methods, Weighted functions	Gupta et al. (2002) [102]
$Fm\|\|C_{\max},T$	ANN	Noorul Haq and Rahda Ramanan (2006) [104]
$HF2\|\|(C_{\max},Z_{\mathrm{cost}}),(C_{\max},Z_{\mathrm{cost}},N_T,Z_q)$	EA	Wei et al. (2006) [107]
$Fm\|\|C_{\max},T_{\mathrm{w}}$	EMEA	Shi and Zhou(2007) [105]
$Fm\|stch\|C_{\max}, T$	indicator-based EA	Figueira et al. (2010) [109]
$Fm\|prmu\|many$	LS	Geiger(2007) [103]
$Fm\|nwt\|C_{\mathrm{ave,w}}, T_{\mathrm{ave,w}}$	MOSS	Rahimi-Vahed et al. (2008) [114]
$Fm\|prmu\|E_{\mathrm{ave,w}},T_{\mathrm{ave,w}}$	SFLA	Rahimi-Vahed et al. (2009) [115]
$HFm(PM)\|SDST\|I_{\mathrm{sum}}, C_{\mathrm{w}}, Z_{\mathrm{due}}$	GRASP	Davoudpour and Ashrafi (2009) [137]
$Fm\|\|C_{\max},T_{\mathrm{w}}$	EM	Naderi et al. (2010) [117]
$HFm\|SDST\|C_{\max},Z_{\mathrm{cost}}$	MOHM	Behnamian and Fatemi Ghomi (2010) [116]

et al. [33] developed a new implementation of a bound and coordination procedure to enhance the B&B algorithm. It is valid to solve the two-machine FSP with the minimization of makespan and sum of completion times. Sivrikaya-Serifoğlu and Ulusoy [34] presented three B&B approaches and two heuristics to solve two-machine FSP with the minimization of a weighted flow time and the makespan. The three B&B methods are compared so as to find which differs mostly in their branching strategies. Sayın and Karabat [35] discussed FSP with the same objectives of Liao et al. [33] by an efficient B&B procedure. Yeh [36] developed an efficient B&B approach to handle two-machine FSP with the objective of minimizing a weighted combination of job flow time and schedule makespan. Lee and Wu [37] also used B&B methods to study the two-machine FSP with different objectives: the minimization of flow time and total tardiness. Yeh [38] developed another efficient B&B algorithm to handle two-machine bi-criteria FSP with the aim of minimizing a weighted flow time and makespan. The proposed algorithm gives better results than the existing ones on former randomly generated problems. T'kindt et al. [15] designed mathematical programming formulations, a B&B algorithm, and a heuristic algorithm for the two-machine FSP. The objective of this FSP is to minimize total completion time which is subject to the constraint of the minimization of makespan. The results show that the B&B method proposed here is quite effective in handing MFSP. Lin and Wu [39] paid special attention to two-machine FSP with the objectives of the makespan and total completion time. They proposed a B&B approach which is approved in the practical significance and potential applications of this field. Lemesre et al. [17] focused on bi-objective FSP with the minimization of makespan and total tardiness. They proposed a B&B method hybridized with a meta-heuristic algorithm. It is managed to confirm solutions given by the meta-heuristic for middle-sized problems and to improve solutions for large-sized problems.

4.2 Simulated annealing

The SA algorithm is a versatile random search approach simulating the annealing process. It has grown into an optimization technology for tackling combinatorial optimization and other NP-hard problems. The development of SA is based on the Metropolis algorithm [40]. Kirkpatrick et al. [41] and Cerny [42] pointed out that a model for simulating the annealing of solids could be used for the optimization of problems. The annealing schedule is very simple and easy to execute. First, a high initial temperature is set for the basic procedure of the SA. This temperature is gradually decreased by the cooling schedule until a freezing temperature is reached. The proper rate of cooling is an essential part of SA as it determines the performance of SA. New solutions will be accepted if the acceptance criterion is achieved. The vital weakness of this method is that sometimes it will get trapped in local optima mainly due to the lack of diversity of the search space. Researchers often design new neighborhood search structures to improve its performance.

Suresh and Mohanasundaram [43] proposed a Pareto-based simulated annealing algorithm for FSP with makespan and total flow time. The experiments were conducted using uniform random numbers. In [44], a new perturbation mechanism named "segment-random insertion" scheme is designed to generate a neighborhood set of candidate solutions. This algorithm utilizes an archive containing the non-dominated solution set. The new current solution is produced by a scale weighted sum of the objective functions. Additionally, they also proposed a restart strategy and a reannealing method. Varadharajan and Rajendran [45] presented a multi-objective simulated annealing algorithm (MOSA) to solve PFSP with the objective of minimizing the makespan and total flow time of jobs. They used simple and fast existing heuristics to obtain two initial sequences which are supplemented by the implementation of three improvement schemes. In order to generate solutions on the Pareto optimal front, MOSA uses the implementation of a simple probability function to obtain non-dominated solutions. The probability function varied in such a way that the entire objective space is covered uniformly. Thus, they could obtain as many non-dominated and well-dispersed solutions as possible. Naderi et al. [18] developed a meta-heuristic approach based on simulated annealing to consider HFSP to minimize both total completion time and total tardiness. Hatami et al. [46] built a novel model of bi-objective flow shop scheduling with the constraints of a production system. The two effective meta-heuristic algorithms, SA and TS, were used to tackle this problem with this new model. The computational results were illustrated to show the efficiency of SA. Two key objectives, minimization of mean flow time and maximum tardiness, are considered in this paper.

4.3 Tabu search

TS proposed by Glover [47] emulates an intelligent attitude using an adaptive memory and tries to create memories itself, which is similar to the use of some memory functions of people, in order to find its way out. TS can avoid being trapped at local optimum with the aid of a memory function and can prevent the searching from exhausting much more time by becoming an oscillation. Simultaneously, TS is proven to be a successful way for solving various combinatorial optimization problems, especially in scheduling field.

Loukil et al. [48] dealt with many different scheduling problems with different combinations of objectives. They mainly used a multi-objective tabu search (MOTS). Armentano and Arroyo [49] proposed a MOTS to handle MFSP. They considered the bi-criteria of minimizing the makespan and the maximum tardiness. The algorithm works with several paths of solutions in parallel, each with its own tabu list. Solutions are initialed and selected from the neighborhoods with some heuristics used. The dispersion of points is achieved by a clustering procedure. Additionally, an external archive for storing all the non-dominated solutions has been found during the execution. However, the initialization procedure of MOTS takes most of the allotted CPU time for large values of n. Eren and Güner [19] proposed four tabu search methods for handling MFSP with setup times. The objective is to minimize the weighted sum of total completion time and total tardiness. Specifically, the model is built with setup times. The same authors developed a heuristic algorithm and a TS method to solve large-sized problems with a learning effect [50]. It is aimed at minimizing the weighted sum of total completion time and makespan. Later, Eren and Güner [51] analyzed the objectives including minimization of the weighted sum of total completion time, total tardiness, and makespan [19, 50]. They designed a TS procedure and compared it with the modified Nawaz, Enscore and Ham (NEH) algorithm, random search, and the earliest due date (EDD) rule. The results show that the performance of the heuristic based on the proposed TS is the best of all in terms of solution quality.

4.4 Genetic algorithms

The concept of GA based on the mechanism of evolution (i.e., natural selection) is developed by Holland and his colleagues in the 1970s [52]. In GA terminology, a solution is called an individual or a chromosome which is made of discrete units called genes. GA uses two operators of crossover and mutation to generate new solutions from existing ones. Reproduction involves the selection of chromosomes for the next generation. In the most general cases, the fitness of an individual determines the probability of its survival for the next generation. This procedure is circulated to perform an adaptive search for an optimal or near-optimal solution to the problem.

A generic single-objective GA can be modified to find a set of multiple non-dominated solutions in a single run. Simultaneously, the ability of GA can search different regions of a solution space. Hence, using GA is possible to find a diverse set of solutions for difficult problems with non-convex, discontinuous, and multi-model solution space features. Specifically, GA has been successfully applied to single-objective sequencing. More and more attention has been paid to making use of GA for MFSP by now. Multi-objective GA is also designed to enhance the efficiency and strength of searching Pareto solution sets in this field. The applications of GA for MFSP are reviewed in this section.

Neppalli et al. [53] designed two GAs in two-machine bi-criteria FSP. The first approach is based on the concept of vector evaluated GA and the second one is referred to as the weighted criteria approach. The performance of the two GAs are compared with the only known heuristics for the problem and showed satisfactory results. Murata et al. [23] proposed a MOGA in cases with two objectives and three objectives, respectively. This approach is combined with a simple GA and the modified selection is based on a weighted sum of multiple objective functions. It gains a concave Pareto front with minimization of the makespan and the total tardiness. Later, Murata et al. [54] designed a cellular multi-objective genetic algorithm. This procedure extends MOGA [23] by introducing a cellular structure. A proportional weight specification method is proposed to find a variety of Pareto optimal solutions for a multi-objective optimization problem. Each individual is located in a cell with a different weight vector. Computer simulations show that this procedure gives good Pareto solutions to two- and three-objective FSPs. Bagchi [55] developed the elitist non-dominated sorting GA to deal with MFSP. It is the enhanced version of non-dominated sorting genetic algorithm (NSGA) which is not to mercilessly discard the old population and replace it completely by the offspring. An additional non-dominated sorting of the combined parents + progeny archive is performed. Then, a controlled fraction (the top 50%) of the non-dominated solutions is selected to form the next generation. Chang et al. [56] proposed a GA for MFSP. This method uses a gradual priority weighting approach to calculate fitness of solutions for searching the Pareto optimal solutions. It is compared with searching for Pareto optimal solutions of variable weights. The results are shown to be better in the CPU times, the average number, and the percent of non-dominated solutions. Dugardin et al. [57] designed three methods to deal with the reentrant hybrid flow shop scheduling problem with the minimization of the cycle time and the maximization of the utilization rate of the bottleneck. The algorithms adapted the Lorenz dominance strategy rather than Pareto dominance to select the non-dominated solution. L-NSGA proposed in this paper achieved a close set to the Pareto front in the computational tests. Karimi et al. [58] presented a multi-phase approach to tackle hybrid flexible FSP considering the minimization of makespan and total weighted tardiness simultaneously. This multi-phase GA including three phases is tested in small-, medium-, and large-sized instances, respectively, compared with MOGA and is shown to be much more effective.

Most real-world manufacturing problems, such as multi-objective flow shop scheduling problems, are too complex to be solved by single approaches. More and more researchers are interested in hybrid algorithms combined with genetic algorithms and other approaches to solve MFSP. Nagar et al. [24] proposed a hybrid GA to solve two-machine FSP with two objectives. This method uses a branch-and-bound procedure to generate some initial populations. The results indicate that the combined approach and its modified versions are better than either of the pure strategies as well as the heuristic algorithms. Sridhar and Rajendran [59] suggested a hybrid GA for MFSP with the aims of minimizing makespan, flow time, and idle time. Effective heuristic algorithms are used for initialization in this GA. Cavalieri and Gaiardelli [60] developed two hybrid GAs for FSP with multi-criteria. The first GA is to allocate and sequence the jobs. The second GA combined with a dispatching rule (EDD) is to limit its task only on the allocation of the jobs. Yeh [61] proposed a GA combined with a modified B&B for two-machine cases. This hybrid GA is compared with the B&B algorithm, the conventional genetic algorithm, and the best-known heuristic algorithm. The computational results demonstrate that the hybrid GA gives better qualities. Ishibuchi and Murata [25] presented a multi-objective genetic local search algorithm (MOGLSA). It extended GA using a local search step which is applied to every new solution after the crossover and mutation procedures. Ishibuchi et al. [62] made a comprehensive study about adding the local search to the former MOGLSA [25]. In order to improve MOGLSA, good initial solutions for local searches are selected and a local search direction is assigned to each initial solution. Ponnambalam et al. [63] discussed a multi-objective evolutionary search algorithm called traveling salesman problem solving by GA (TSPGA). The TSPGA is combined with a traveling salesman algorithm and a genetic algorithm to solve MFSP. It is aimed at minimizing the makespan, mean flow time, and machine idle time. The implemented GA uses randomly generated but specified weights to sum up objective functions. It is not compared with any other methods from the previous literature, but some results on small flow shop instances are shown. Arroyo and Armentano [64] proposed a genetic local search algorithm named MOGALS. To gain Pareto front, preservation of dispersion in the population, elitism, and a parallel multi-objective local search in distinct regions are used in this algorithm. Additionally, the concept of Pareto dominance is used both to assign fitness to the solutions and in the local search procedure. At last, this algorithm is shown to be better than other algorithms, and yet it is not compared with the former algorithms of their work. Pasupathy et al. [65] proposed a multi-objective genetic algorithm named Pareto GA with an archive of non-dominated solutions subjected to a local search (PGA-ALS). Four initial heuristics solutions are introduced in the initial population. Its selection process provides a balance between exploration and exploitation. Additionally, a crowding procedure is also designed and used as a secondary selection criterion. Good non-dominated front can be found in this algorithm process under some restricted conditions. Melab et al. [66] proposed a grid-based parallel GA with the aim of obtaining an accurate Pareto front. They focused on FSP with the minimization of the makespan and the total tardiness criteria. This hybrid genetic–memetic algorithm named AGMA combines a genetic algorithm (GA) and a memetic algorithm (MA). However, the authors did not test the proposed approach against other existing algorithms. Li and Wang [67] solved MFSP using a hybrid quantum-inspired GA (HQGA). This HQGA is based on a quantum-inspired GA and permutation-based GA (PGA). Two trimming techniques for population are proposed to maintain the diversity of the population. The results state that solutions with good proximity and diversity can be gained by the proposed HQGA. Chang et al. [68] presented a mining gene structures on subpopulation genetic algorithm which is combined with mining gene structure approach and subpopulation genetic algorithm (SPGA). It is demonstrated to be effective to balance convergence and diffusivity of the searching.

4.5 Ant colony optimization

The ACO algorithm proposed by Dorigo, Maniezzo, and Colorni [69, 70] is a novel distributed modern meta-heuristic algorithm. The main idea of ACO algorithm is based on the ability of ants to find the shortest paths from their nest to food locations using pheromone trails. Thus, ACO algorithms solve the combinatorial optimization problems by mimicking real ants' behavior. Each step of this constructive procedure is determined by the pheromone trails in real ants. Cooperation between ants depends on the common structure of shared pheromone matrix update principles. General ACO algorithms can gain higher quality Pareto solutions, but need more computational time.

T'kindt et al. [20] proposed an ACO algorithm to solve the two-machine FSP with minimizing both the total completion time and the makespan. Local search and SA are used to emphasize the convergence and diversification of the ACO algorithm, respectively. Rajendran and Ziegler [71] proposed two ant-colony optimization algorithms for solving PFSP. The objective is to minimize total flow time and the makespan. The first algorithm extends the ideas of the ant colony algorithm called max–min ant system [72] by incorporating the summation rule [73] and a newly proposed local search technique. The second ant colony algorithm is newly developed. Computational results show that the proposed algorithms perform better than the existing ACOs. Pasia et al. [74] investigated the performance of a Pareto ACO. This method is hybridized by incorporating path relinking in four different ways. Computational results show that it is competitive to other heuristics mentioned. Yagmahan and Yenisey [75] discussed multi-objective of makespan, total flow time, and total machine idle time in flow shop scheduling. Computational results demonstrate that the proposed algorithm is more effective and better than other compared methods. Recently, Marimuthu et al. [76] used the ACO to solve MFSP with setup times. They prove that this algorithm is powerful in searching for non-dominated solutions. Huang and Yang [77] presented an ACO algorithm to tackle the MFSP with minimization of machine idle time, job wait time, and tardiness. Job overlapping is considered here. The effect and the efficiency of this proposed ACO are not compared with other algorithms. The FSP with minimization of makespan and total flow time is considered in [78]. The authors designed a multi-objective ant colony system algorithm to settle this discrete problem. This algorithm is compared with hybrid algorithm for multi-criteria (HAMC) and GA and can achieve better Pareto solutions than the two algorithms.

4.6 Particle swarm optimization

PSO is originally proposed by Kennedy and Eberhart [79]. It is a swarm intelligence method to search for optimization solutions. The main idea of PSO is based on the observations of the social behaviors of bird flocking. Initial solutions are generated with a randomized velocity and new solutions are gained by competition and corporation between particles. Best previous experience of particles themselves and best experience of all other members are two decisive factors for searching optimal or near-optimal solutions [79]. This is similar to human behaviors in making decision where people consider their own best past experience and the best experience of the other people around them [80]. PSO has been applied in many fields because of its good characteristics, such as quick convergence and easy implementation. But its theory is not consummate until now. For example, the applications of PSO are restricted in discrete optimization problems, though some studies have been progressing in this field.

Guo et al. [81] proposed a discrete particle swarm optimization (DPSO) algorithm to find the global best position for particles. The fitness function of this method uses phenotype sharing function. The result shows that the proposed DPSO can reach a good approximation of true Pareto front. Rahimi-Vahed and Mirghorbani [82] utilized PSO in multi-objective scheduling. Li et al. [83] presented a multi-objective PSO to solve permutation MFSP. Local searches based on NEH and SA are respectively designed in the hybrid algorithm. Simulation results demonstrate that this algorithm produces good solutions approximating the real Pareto front. Tseng and Liao [21] solved a lot-streaming flow shop scheduling problem with minimization of the total weighted earliness and tardiness. They improved the DPSO of [84] by incorporating it with the proposed net benefit of movement (NBM) algorithm searching for optimal and near-optimal solutions, respectively. Sha and Lin [85] settled a multi-objective FSP with minimization of the makespan, mean flow time, and machine idle time using a PSO. This PSO achieves a good set of Pareto solutions by comparison with other traditional heuristics. However, those state-of-the-art algorithms did not appear in the tests.

4.7 Differential evolution

The DE algorithm is one of the latest evolutionary heuristic methods presented by Storn and Price [86] for complex continuous problems. Like other evolutionary algorithms, it includes simple mutation, crossover, greedy selection with some basic principles, and other evolutionary operators. These operators in the iterative evaluating process will stop when a suitable, user-defined stopping criterion is achieved. This new method has been proven to be robust, easy to implement, and with quick convergence. So the DE algorithm gains much more attention and has been successfully used in many fields. Currently, there are several variants of DE proposed to improve efficiency over continuous and discrete problems. Multi-objective flow shop scheduling problems are no exception.

Qian et al. [26] devised a multi-objective hybrid differential evolution (MOHDE) in MFSP with minimization of makespan and maximum tardiness. They designed a variable neighborhood search (VNS) incorporated with the proposed MOHDE to enhance the ability of local search. Later, Qian et al. [87] proposed a memetic algorithm based on differential evolution (MADE) to settle no-wait MFSP. To match this problem, they also designed some rules including the largest-order-value rule, the concept of Pareto dominance, several local searchers, and a speedup computing method. Qian et al. [88] also developed an effective hybrid DE-based algorithm for MFSP with limited buffers between consecutive machines. This HDE has adopted many steps to balance exploration (global search) and exploitation (local search). Pareto dominance is used to solve the updating of solutions for multi-objective optimization. This HDE is competitive to other algorithms in the effectiveness and the efficiency aspects. After that, Pan et al. [89] suggested a discrete differential evolution named DDE. They compared DDE with the HDE of [88] and gained better Pareto solutions. The job permutation-based mutation and crossover operators are taken in this DDE. Then, an elaborate one-to-one selection scheme is designed by the domination status of a candidate. It is considered a counterpart target individual as well as a found archive set of the non-dominated solutions. Additionally, an effective local search algorithm is proposed to merge into the DDE algorithm. Thus, it can balance global exploration and local exploitation. Speedup technologies and their insert neighborhoods are also used to reduce the running time requirements. Tianmin and Mitsuo [90] proposed a novel hybrid DE algorithm based on the basic quantum-inspired evolutionary algorithm to tackle a multi-objective flow shop scheduling problem. Minimization of the makespan, total flow time, and the maximization of lateness of jobs were tested in the benchmark problems separately. The effectiveness of this method was proven in computational results. But these three objectives were not tested simultaneously.

4.8 Immune algorithm

The biological immune system is a robust, complex, and adaptive system which defends the body from foreign pathogens. Depending on the type of pathogen and the way it inters the body, the immune system uses different response mechanisms to either neutralize the pathogenic effect or destroy the infected cells. Immune algorithm is based on the simulation of the biological immune system with excellent performance of optimization. It can prevent premature convergence with the inhibition and the promotion reaction between antibodies. Also, it is adopted to solve large-scale and the high complexity of the scheduling problems according to its self-organization of the memory function. The intelligent scheduling algorithm is based on the physiology immune mechanisms [91].

Yang et al. [92] solved MFSP using an intelligent scheduling method which is based on the physiology immune mechanism. It can prevent premature convergence and promote population diversity and accelerate the convergence speed to some degree. Specially, it is good at dealing with large-scale MFSP. Tavakkoli-Moghaddam et al. [27] presented a hybrid multi-objective immune algorithm named HMOIA. It is based on biological immune system and bacterial optimization. HMOIA focuses on no-wait MFSP with the minimization of the weighted mean completion time and weighted mean tardiness. The performance of this HMOIA outperforms PS-NC GA, NSGA-II, SPEA-II, MOIA, and MISA, especially for large-sized problems. Tavakkoli-Moghaddam et al. [93] designed an effective MOIA for the same problem in [27]. This MOIA performs better than SPEA2. The objectives are to minimize the weighted mean completion time and the weighted mean tardiness.

4.9 Other methods

A variety of distinct procedures have been developed to handle MFSP. Many of these procedures do not fall under any of the general categories defined above. They mainly include some heuristics and meta-heuristics procedures. Heuristic methods or rules are often used in manufacturing field, which do not guarantee optimal or near-optimal solutions. The main goal of heuristics is to produce solutions of acceptable quality in reasonable time. Heuristics are generally regarded as the effective approaches for solving a large variety of combinatorial optimal problems. However, such individual heuristics do not always perform well for the variety of problem instances which may be encountered in practice. There is a wide range of modern heuristics known from the literature. They are specifically designed and tuned to solve certain classes of optimization problems. These methods are based on the partial search of the solution space and often referred to as meta-heuristics [94].

Rajendran [95] first addressed the problem of scheduling to minimize makespan and total flow time. He proposed a new heuristic algorithm which is compared with Ho and Chang's [96]. This new proposed procedure gives competitive results. Gupta et al. [16] paid attention to a minimum total flow time schedule subject to the minimization of makespan. They developed two optimization algorithms for bi-criteria MFSP. Computational experience shows that both algorithms performed better than the current heuristics for the bi-criteria problem. Framinan et al. [97] made a comprehensive computational evaluation for the m machine flow shop problem to minimize makespan and flow time. They developed some new heuristics to solve MFSP. Allahverdi [98] proposed three new heuristic algorithms with the minimization of the weighted sum of makespan and mean flow time. In addition, two dominance relations are developed. The results show that they are superior to all the existing heuristics in the literature including a genetic algorithm. Later, Allahverdi [99] addressed the m-machine flow shop problem. The objective is to minimize the weighted sum of makespan and maximum tardiness. Arroyo and Armentano [100] suggested some heuristics for several two- and three-objective combinations among makespan, flow time, and maximum tardiness. Ravindran et al. [22] solved FSP with the objectives of makespan time and total flow time. Three heuristic algorithms named HAMC1, HAMC2, and HAMC3 are proposed by them. Framinan and Leisten [101] proposed a new heuristic. It is said to perform well at quality of the solution as well as the number of feasible solutions.

Local search is always combined with other algorithms to handle combinatorial optimization problems because it can enhance the exploitation of the proposed algorithms. Gupta et al. [102] presented some local search procedures and three meta-heuristics for two-machine FSP. The methods developed are simulated annealing, threshold accepting, tabu search, and multilevel search algorithms. Then, Geiger [103] studied several local search algorithms to analyze the distribution of several objectives. Several combinations of criteria are tested here. Haq and Ramanan [104] described a bi-criteria FSP using artificial neural network (ANN). Shi and Zhou [105] developed an escalating multi-objective evolutionary algorithm (EMEA) to solve bi-criteria FSP. This algorithm used a new elite duplication strategy and an innovative escalating evolutionary structure. It can prevent premature convergence and promotes population diversity and efficiency. Zhang et al. [106] developed an approach of combing forward simulation and backward simulation to deal with bi-criteria HFSP with minimization of makespan and average flow time. Wei et al. [107] presented an effective EA which could dynamically adjust the fitness assignment in optimizing scheduling process. It is designed for HFSP with two objectives in the first case and four objectives in the second case. Behnamian et al. [108] studied a multi-phase method to solve HFSP of sequence-dependent setup time (SDST). The aim of this problem is to minimize the makespan and the sum of the earliness and tardiness of jobs. Figueira et al. [109] used improved indicator-based evolutionary methods to tackle bi-criteria flow shop problem with stochastic processing times. The objectives of makspan and total tardiness are considered in this paper. No other methods are compared with the three proposed methods. Sawik [110] proposed a lexicographic approach to bi-objective HFSP.

Some computational results are provided in a real-world make-to-order flexible assembly line. Janiak et al. [111] used some constructive algorithms and meta-heuristic algorithms (SA, TS, and hybrid algorithm) to tackle the HFSP. The optimal solutions of the total weighted earliness, the total weighted tardiness, and the total weighted waiting time are minimized in this searching process. Javadi et al. [112] developed a fuzzy multi-objective linear programming method (FMOLP) to settle MFSP with special constraint of no-wait. This FMOLP can satisfy the decision maker's requirements. Jungwattanakit et al. [113] developed some well-known heuristics to handle HFSP with minimization of a combination of makespan and the number of tardy jobs. A new multi-objective scatter search (MOSS) is developed to handle a bi-criteria no-wait flow shop scheduling problem with the minimization of weighted mean completion time and weighted mean tardiness [114]. This MOSS gains a better performance than SPEA-II, but lower efficiency. A hybrid multi-objective algorithm combining shuffled frog-leaping algorithm (SFLA) with VNS is designed to consider minimization of the weighted mean earliness and the weighted mean tardiness in PFSP [115]. This method is shown to have better effectiveness and worse efficiency, just like the performance of the former algorithm. Behnamian and Fatemi Ghomi [116] presented a multi-objective hybrid metaheuristic (MOHM) which is a combination of two multiple objective decision-making methods, min–max and weighted techniques, a new solution presentation method, and robust hybrid meta-heuristic. GA and VNS were the main algorithm structure for MOHM. SDST hybrid flow shop scheduling problem with the objective of minimization of makespan and total resource allocation costs was solved in this paper. It was proven to be an effective method during 252 benchmark problems. However, the contribution of MOHM was focused on the combination of different algorithms. New mechanism is not presented here. EM algorithm is a new and rapid development of applications in solving scheduling and planning problems domain. Naderi et al. [117] combined EM and SA for handling bi-objective flow shop scheduling problems with minimization of makespan and total weighted tardiness. The performance of the presented EM algorithm outperforms SA and other foregoing heuristics applied to this paper in benchmark problems, while the effective meta-heuristics like MOGA and MOPSO were not tested in comparison with the presented EM.

5 Challenges

After the review and analysis of literature for MFSP, we have to face an urgent problem: where are we heading now in the field of solving MFSP. Over the last decade, researchers have developed many innovative algorithms in conventional MFSP. In this paper, we hope to provide some research directions which would be worthwhile to head this field in the future.

5.1 Computational efficiency of solving problems

Flow shop scheduling is known to be a difficult, strongly NP-hard problem. For the multi-objective FSP, the implementation of obtaining non-dominated solutions needs to consider the evaluation of several different objectives. Therefore, MFSP is more complex than single-objective FSP and needs to consume more excessive computational time. At present, most of the literature focused on the effectiveness of the algorithm and ignored computational efficiency. Future work would focus on improving the quality/time for MFSP. Theoretical study is needed to surmount these tough problems. There are two areas of improving computational efficiency which are worth paying much attention to in the future.

Different methods have their advantages and disadvantages and could be applied to solving the suitable problem. They are not a global panacea for every MFSP. For example, a priori algorithms turn out to be more efficient in solving the small MFSP instances than the other two approaches. For example, objective-weighted method, lexicographic ordering, and goal programming can settle small instances efficiently, and they are very simple and easy to implement. Unfortunately, most of the review articles in the literature fall into categories of models not solvable or not appropriate to a priori methods.

Another interesting issue is that innovations in algorithms should be studied for the sake of eligible efficiency. According to the published literature for MFSP, the algorithms pay much attention to pursue high-quality Pareto solutions and the results are obtained by complicated implementation as well as excessive computational time. The design of novel algorithms of MFSP which achieve a satisfactory balance between exploration (global search) and exploitation (local search) is an interesting direction in future research.

5.2 Study of dynamic MFSP under uncertain environment

A good deal of previous researches in MFSP concentrates on static scheduling. However, in a manufacturing system, production is an activity of sustained pursuit and scheduling parameters are not always precise due to both human and machine resource factors. As a result, neither the release times of jobs nor their precise processing requirements are known in advance. Therefore, scheduling is a non-deterministic problem, and these disruptions include machine failure, quality problems, arrival of urgent jobs, operator unavailability, out-of-stock condition, changes in availability date, and so on. Consequently, classical approaches, within a deterministic scheduling theory, relying on precise data might not be suitable for the representation of uncertain and dynamic scenarios, and there still remains a gap between theory and practice.

Dynamic scheduling problem in uncertain environment is more complex than static scheduling problem because of the additional unforeseen events. From the current literature, periodic and event-driven rescheduling strategies are often employed to dynamic scheduling, which decompose the dynamic scheduling problem into a series of continual and static scheduling problems, and the multi-objective algorithm is applied to optimizing each of the static scheduling problems. The most proposed solutions to deal with uncertain situations are fuzzy mathematics methods, the combination of hypothesis test, and intelligent optimization techniques. These methods presented in the current literature are only feasible in theory and the range of their application is extremely narrow. How to evaluate the performance of MFSP under uncertain and dynamic conditions is also an interesting theme to be considered.

In addition, it is difficult to merge the uncertain and dynamic conditions in practice models, and these directions will attract more attention in further research.

5.3 Study of incorporating preferences into MFSP

Measurement, search, and decision making are involved in dealing with many multi-objective flow shop scheduling problems. Generally, a large number of Pareto solutions are attained after implementation of the multi-objective algorithm, and the decision maker has to choose one from the massive alternative results. It is really necessary to reduce the large workload and take into account the preferences of the decision maker in the multi-objective algorithm. So incorporating preferences into the algorithms of MFSP needs more attention in further research, which could guide direction of the search process and reduce the computational expense.

Some proposals in handling preferences for multi-objective optimization problems are presented [118–125], but researches on incorporating preferences in MFSP are rather sparse. It is noteworthy that interactive methods require the decision maker to change his preference interactively, release the decision maker's cognition burden, and gain the decision maker's satisfied solution rapidly. Special issues on designing preference-based mechanism within novel algorithms, handling incorporation of preferences from several decision makers, as well as applications of interactive approach are challenging. It is very significant for tackling a large number of objectives which are often encountered in practical MFSP.

5.4 Study of handling constraints for real-world application

Some researchers have contributed to bridge the gap between the theory and practice of flow shop scheduling. Many real-world FSPs are modeled into constrained single-objective problems in [126–133] and multi-objective problems in [134]. The real-world models of MFSP would be established with release time, lot sizing, setup times, limited buffer, no-wait, no-idle, due date, maintenance availability, machine availability, the cost of production, the constraining resource, and other factors. However, these constraints would lead to increasing unfeasible regions, high complexity of the problems, and the difficulty of the search. The methods of handling constraints are divided into two categories: generic methods and specific methods. The former ones mainly include the penalty function method, the Lagrange multiplier method, and the complex search method [135, 136]. They are much popular for being easily implemented while their performance is not eligibly satisfied. The latter ones mainly include the cutting plane method, the reduced gradient method, and the gradient projection method [135, 136]. The execution can achieve excellent results, but the computational burden is increased due to the large number of variables. Actually, utilizing properties of problems artistically would reduce the difficulty of problems (e.g., the use of the converse property of PFSP). This is an interesting theme which provides insights into the design of handling constraints procedures.

6 Conclusions

In this paper, we provide a comprehensive review of the most popular approaches for solving MFSP together with some insights of the research of their operations. The study of MFSP has attracted a great many researchers to develop effective and efficient approaches. The approaches involved are EA, GA, TS, SA, GP, ACO, PSO, IA, TA, DE, local search, B&B, and some other dispatching rules, which could be classified into three groups: *a priori* methods, *interactive* methods, and *a posteriori* methods. This review can be a good reference in this field with helpful suggestions for further research.

This survey starts with a broader introduction of FSP and the complexity of it under the criteria being used frequently in single-objective FSP. A simple and clear description of HFSP and PFSP is discussed. Three-category literature groups and analysis of multi-objective optimization are introduced as well as the main methods review. Later, some subjects of further studies on MFSP are given. The review clearly shows that some interesting progress has already been made in the area, but much more research still needs to be done. In this paper, some promising directions are proposed for further research: (1) Focusing on higher quality/time ratio of algorithms to solve MFSP is advisable. (2) Settling dynamic MFSP under uncertain conditions is challenging. (3) Paying much attention to the decision maker's preferences in searching process would be beneficial and the interactive methods would handle it. (4) Increasing of studies on real-world application involving constraints.

Finally, properties of problems themselves would provide more chances to search for higher quality solutions and reduce the difficulty of problems for evaluating. Algorithms based on properties of problems would be competitive and challenging for future work.

Acknowledgment This project is supported by 863 High Technology Plan Foundation of China under grant no. 2007AA04Z107, the National Natural Science Foundation of China no. 50825503, the State Key Program of National Natural Science of China no. 51035001 and the program for New Century Excellent Talents in University under grant no. NCET-08-0232. The authors would like to thank the anonymous referees for the insightful comments that have helped improve this work.

References

1. Shi RF (2008) Current progress in evolutionary algorithm based multi-objective production scheduling. Journal of Jishou University (Natural Science Edition) 29(6):42–46 (in Chinese)
2. Nagar A, Heragu SS, Haddock J (1995) Mutiple and bi-criteria scheduling: a literature survey. Eur J Oper Res 81:88–104
3. T'kindt V, Billaut JC (2001) Multicriteria scheduling problems: a survey. Rairo Oper Res 35:143–163
4. Jones DF, Mirrazavi SK, Tamiz M (2002) Multiobjective metaheuristics: an overview of the current state-of-the-art. Eur J Oper Res 137:1–94
5. Hoogeveen H (2005) Multicriteria scheduling. Eur J Oper Res 167:592–623
6. Minella G, Ruiz R, Ciavotta M (2008) A review and evaluation of multiobjective algorithms for the flowshop scheduling problem. Informs J Comput 20(3):451–471
7. Pinedo ML (2008) Scheduling: theory, algorithm, and systems, 3rd edn. Springer, Berlin
8. Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discrete Math 5:287–326
9. Garey MR, Johnson DS, Sethi R (1976) The complexity of flowshop and job shop scheduling. Math Oper Res 1:117–129
10. Gonzalez T, Sahni S (1978) Flowshop and jobshop schedules: complexity and approximation. Oper Res 26:36–52
11. Du J, Leung JYT (1990) Minimizing total tardiness on one machine is NP-hard. Math Oper Res 15:483–495
12. T'kindt V, BILLANT J (2005) Multicriteria scheduling: theory models and algorithms [M], 2nd edn. Springer, Berlin
13. Carlos Coello, Gary Lamont, David Veldhuizen (2007) Evolutionary algorithms for solving multi-objective problems, 2nd edn, vol 2. Springer, New York, pp 5–60
14. Fourman MP (1985) Compaction of symbolic layout using genetic algorithms. In: Grefenstette JJ (ed) Genetic and algorithms and their applications: Proceedings of the First International Conference on Genetic Algorithms. Lawrence Erlbaum, Hillsdale, pp 141–153
15. T'kindt V, Gupta JND, Billaut JC (2003) Two-machine flowshop scheduling with a secondary criterion. Comput Oper Res 30:505–526
16. Gupta JND, Neppalli VR, Werner F (2001) Minimizing total flow time in a two-machine flowshop problem with minimum makespan. Int J Prod Econ 69:323–338
17. Lemesre J, Dhaenens C, Talbi EG (2007) An exact parallel method for a biobjective permutation flowshop problem. Eur J Oper Res 177:1641–1655
18. Naderi B, Zandieh M, Balagh AKG, Roshanaei V (2009) An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. Expert Syst Appl 36:9625–9633
19. Eren T, Güner E (2006) A bi-criteria flowshop scheduling problem with setup time. Appl Math Comput 183:1292–1300
20. T'kindt V, Monmarche N, Tercinet F, Laugt D (2002) An ant colony optimization algorithm to solve a 2-machine bi-criteria flowshop scheduling problem. Eur J Oper Res 142:250–257
21. Tseng CT, Liao CJ (2008) A discrete particle swarm optimization for lot-streaming flowshop scheduling problem. Eur J Oper Res 191:360–373
22. Ravindran D, Haq AN, Selvakuar SJ, Sivaraman R (2005) Flow shop scheduling with multiple objective of minimizing makespan and total flow time. Int J Adv Manuf Technol 25:1007–1012
23. Murata T, Ishibuchi H, Tanaka H (1996) Multiobjective genetic algorithm and its applications to flowshop scheduling. Comput Ind Eng 30:957–968
24. Nagar A, Heragu SS, Haddock J (1995) A branch-and-bound approach for a two-machine flowshop scheduling problem. J Oper Res Soc 46:721–734
25. Ishibuchi H, Murata T (1998) A multiobjective genetic local search algorithm and its application to flowshop scheduling. IEEE Trans Syst Man Cybern C 28(3):392–403
26. Qian B, Wang L, Huang DX, Wang X (2006) Multi-objective flow shop scheduling using differential evolution. Lect Notes Control Inf 345:1125–1136
27. Tavakkoli-Moghaddam R, Rahimi-Vahed A, Mirzaei AH (2007) A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness. Inf Sci 177:5072–5090
28. Shi RF (2006) Current progress in evolutionary algorithm based multi-objective production scheduling. Journal of Jishou University (Natural Science Edition) 29(6):42–46 (in Chinese)
29. Brintrup AM, Ramsden J, Tiwari A (2007) An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization. Comput Ind 58:279–291
30. Tantar E, Dhaenens C, Figueira JR, Talbi EG (2008) A priori landscape analysis in guiding interactive multi-objective metaheuristics. CEC (IEEE World Congress on Computational Intelligence), pp 4104–4111
31. Kuo CC (2009) Capacitor placement and scheduling using interactive bi-objective programming with valuable trade off approach. Energ Convers Manage 50:995–1003
32. Land AH, Doig AG (1960) An automatic method for solving discrete programming problems. The Econometric Society 28 (3):497–520
33. Liao CJ, Yu WC, Joe CB (1997) Bicriterion scheduling in the two-machine flowshop. J Oper Res Soc 48:929–935
34. Sivrikaya-Serifoğlu F, Ulusoy G (1998) A bicriteria two-machine permutation flowshop problem. Eur J Oper Res 107:414–430

35. Sayın S, Karabatı S (1999) A bicriteria approach to the two machine flow shop scheduling problem. Eur J Oper Res 113:435–449
36. Yeh WC (1999) A new branch-and-bound approach for the $n/2/$ flowshop/$\alpha F+\beta C_{\max}$ flowshop scheduling problem. Comput Oper Res 26:1293–1310
37. Lee WC, Wu CC (2001) Minimizing the total flow time and the tardiness in a two-machine flow shop. Int J Syst Sci 32:365–373
38. Yeh WC (2001) An efficient branch-and-bound algorithm for the two-machine bi-criteria flowshop scheduling problem. J Manuf Syst 20:113–123
39. Lin BMT, Wu JM (2006) Bi-criteria scheduling in a two-machine permutation flowshop. Int J Prod Res 44:2299–2312
40. Metropolis N, Rosenblatt AW, Rosenblatt MN, Teller AH, Teller E (1953) Equation of state calculation by fast computing machines. J Chem Phys 21(6):1087–1092
41. Kirkpatrick S, Gelatt C, Vecchi P (1983) Optimization by simulated annealing. Science 220:671–679
42. Cerny V (1985) Thermodynamics approach to the traveling salesman problem: an efficient simulation algorithm. J Optimiz Theory App 45:41–51
43. Suresh RK, Mohanasundaram KM (2004) Pareto archived simulated annealing for permutation flow shop scheduling with multiple objectives. Proceedings of IEEE Conference on Cybernetics and Intelligent Systems (CIS), vol 2. Singapore, December 1–3, pp 712–717
44. Loukil T, Teghem J, Tuyttens D (2005) Solving multiobjective production scheduling problems using metaheuristics. Eur J Oper Res 161:42–61
45. Varadharajan TK, Rajendran C (2005) A multiobjective simulated annealing algorithm for scheduling in flowshops to minimize the makespan and total flow time of jobs. Eur J Oper Res 167:772–795
46. Hatami S, Ebrahimnejad S, Tavakkoli-Moghaddam R, Maboudian Y (2010) Two meta-heuristics for three-stage assembly flowshop scheduling with sequence-dependent setup times. Int J Adv Manuf Technol 50:1153–1164
47. Glover F (1986) Future paths for integer programming and links to artificial intelligence. Comput Oper 13(5):533–549
48. Loukil T, Teghem J, Fortemps P (2000) Solving multiobjective production scheduling problems with tabu search. Control Cybern 29(3):819–828
49. Armentano VA, Arroyo JEC (2004) An application of a multi-objective tabu search algorithm to a bi-criteria flowshop problem. J Heuristics 10:463–481
50. Eren T, Güner E (2008) A bi-criteria flowshop scheduling with a learning effect. Appl Math Comput 32(9):1719–1733
51. Eren T, Güner E (2008) The triceiteria flowshop scheduling problem. Int J Adv Manuf Technol 36:1210–1220
52. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
53. Neppalli VR, Chen CL, Gupta JND (1996) Genetic algorithms for the two-stage bi-criteria flowshop problem. Eur J Oper Res 95:356–373
54. Murata T, Ishibuchi H, Gen M (2001) Specification of genetic search directions in cellular multiobjective genetic algorithms. In: Zitzler E, Deb K, Thiele L, Coello Coello CA, Corne D (eds) First International Conference on Evolutionary Multi-Criterion Optimization. Springer, Lecture Notes in Computer Science No. 1993, pp 82–95
55. Bagchi TP (2001) Pareto-optimal solutions for multiobjective production scheduling problems. In: Zitzler E, Deb K, Thiele L, Coello Coello CA, Corne D (eds) First International Conference on Evolutionary Multi-Criterion Optimization. Springer, Lecture Notes in Computer Science No. 1993, pp 458–471
56. Chang PC, Hsieh JC, Lin SG (2002) The development of gradual-priority weighting approach for the multiobjective flowshop scheduling problem. Int J Prod Econ 79:171–183
57. Dugardin F, Yalaoui F, Amodeo L (2010) New multi-objective method to solve reentrant hybrid flow shop scheduling problem. Eur J Oper Res 203:22–31
58. Karimi N, Zandieh M, Karamooz HR (2010) Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach. Expert Syst Appl 37:4024–4032
59. Sridhar J, Rajendran C (1996) Scheduling in flowshop and cellular manufacturing systems with multiple objectives: a genetic algorithmic approach. Prod Plan Control 7:374–382
60. Cavalieri S, Gaiardelli P (1998) Hybrid genetic algorithms for a multiple-objective scheduling problem. J Intell Manuf 9:361–367
61. Yeh WC (2002) A memetic algorithm for the $n/2/$flowshop/$\alpha F+\beta C_{\max}$ scheduling problem. Int J Adv Manuf Technol 20:464–473
62. Ishibuchi H, Yoshida T, Murata T (2003) Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Trans Evol Comput 7:204–223
63. Ponnambalam SG, Jagannathan H, Kataria M, Gadicherla A (2004) A TSP-GA multiobjective algorithm for flow-shop scheduling. Int J Adv Manuf Technol 23:909–915
64. Arroyo JEC, Armentano VA (2005) Genetic local search for multiobjective flowshop scheduling problems. Eur J Oper Res 167:717–738
65. Pasupathy T, Rajendran C, Suresh RK (2006) A multiobjective genetic algorithm for scheduling in flow shops to minimize the makespan and total flow time of jobs. Int J Adv Manuf Technol 27:804–815
66. Melab N, Mezmaz M, Talbi EG (2006) Parallel cooperative meta-heuristics on the computational grid. A case study: the biobjective flow-shop problem. Parallel Comput 32:643–659
67. Li BB, Wang L (2007) A hybrid quantum-inspired genetic algorithm for multi-objective flow shop scheduling. IEEE Trans Syst Man Cybern B 37:576–591
68. Chang PC, Chen SH, Liu CH (2007) Sub-population genetic algorithm with mining gene structures for multi-objective flow shop scheduling problems. Expert Syst Appl 33:762–771
69. Dorigo M (1992) Optimization, learning and natural algorithms. Dissertation, Dipartimento di Elettronica, Politecnico di Milano, Italy (in Italian)
70. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 26:29–41
71. Rajendran C, Ziegler H (2004) Ant-colony algorithms for permutation flow shop scheduling to minimize makespan/total flow time of jobs. Eur J Oper Res 155:426–438
72. Stuetzle T (1998) An ant approach for the flow shop problem. Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT_98), vol 3. Verlag Mainz, Aachen, Germany, pp 1560–1564
73. Merkle D, Middendorf M (2000) An ant algorithm with a new pheromone evaluation rule for total tardiness problems. Proceedings of the EvoWorkshops. Lecture Notes in Computer Science, vol 1803). Springer, Berlin, pp 287–296
74. Pasia JM, Hartl RF, Doerner KF (2006) Solving a bi-objective flowshop scheduling problem by Pareto-ant colony optimization. In: Dorigo M, Gambardella LM, Birattari M, Martinoli A, Poli R, Stützle T (eds) Lecture Notes in Computer Science, vol 4150. Ant Colony Optimization and Swarm Intelligence, pp 294–305
75. Yagmahan B, Yenisey MM (2008) Ant colony optimization for multi-objective flow shop scheduling problem. Comput Ind Eng 54:411–420

76. Marimuthu S, Ponnambalamb SG, Jawahar N (2009) Threshold accepting and ant-colony optimization algorithms for scheduling *m*-machine flow shops with lot streaming. J Mater Process Technol 209:1026–1041
77. Huang RH, Yang CL (2009) Solving a multi-objective overlapping flow-shop scheduling. Int J Adv Manuf Technol 42:955–962
78. Yagmahan B, Yenisey MM (2010) A multi-objective ant colony system algorithm for flow shop scheduling problem. Expert Syst Appl 37:1361–1368
79. Kennedy J, Eberhart RC (1995) Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, 4. IEEE Service Center, Piscataway, NJ, pp 1942–1948
80. Kennedy J, Eberhart RC, Shi Y (2001) Swarm intelligence. Morgan Kaufmann, San Francisco
81. Guo WZ, Chen GL, Huang M, Chen SL (2007) A discrete particle swarm optimization algorithm for the multi-objective permutation flow shop sequencing problem. Proceeding of International Conference on Fuzzy Information and Engineering, pp 323–331
82. Rahimi-Vahed AR, Mirghorbani SM (2007) A multiobjective particle swarm for a flow shop scheduling problem. J Comb Optim 13:79–102
83. Li BB, Wang L, Liu B (2008) An effective PSO-based hybrid algorithm for multiobjective permutation flow shop scheduling. IEEE Trans Syst Man Cybern A38(4):818–831
84. Liao CJ, Tseng CT, Luarn P (2007) A discrete version of particle swarm optimization for flowshop scheduling problems. Comput Oper Res 34:3099–3111
85. Sha DY, Lin HH (2009) A particle swarm optimization for multi-objective flowshop scheduling. Int J Adv Manuf Technol 45:749–758
86. Storn R, Price K (1997) Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. J Glob Optim 11:341–59
87. Qian B, Wang L, Huang DX, Wang X (2009) Mutli-objective no-wait flow-shop scheduling with a memetic algorithm based on differential evolution. Soft Comput 13:847–869
88. Qian B, Wang L, Huang DX, Wang WL, Wang X (2009) An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers. Comput Oper Res 36:209–233
89. Pan QK, Wang L, Qian B (2009) A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems. Comput Oper Res 36:2498–2511
90. Zheng TM, Yamashiro M (2010) Solving flow shop scheduling problems by quantum differential evolutionary algorithm. Int J Adv Manuf Technol 49:643–662
91. Forrest S, Javornik B, Smith R, Perelson AS (1993) Using genetic algorithms to explore pattern recognition in the immune system. Evol Comput 1(3):191–211
92. Yang JG, Ding HM, Li P (2002) An immune scheduling algorithm for solving multi-objective flow-shop problem. Mach Des Res 18(4):28–31 (in Chinese)
93. Tavakkoli-Moghaddam R, Rahimi-Vahed A, Mirzaei AH (2008) Solving a multi-objective no-wait flow shop scheduling problem with an immune algorithm. Int J Adv Manuf Technol 36:969–981
94. Chakhlevitch K, Cowling P (2006) Hyperheuristics: recent developments. In: Cotta C, Sevaux M, and Sörensen K (eds) Adaptive and multilevel metaheuristics, studies in computational intelligence, 136. Springer 2008, pp 3–29
95. Rajendran C (1995) Heuristics for scheduling in flowshop with multiple objectives. Eur J Oper Res 82:540–555
96. Ho JC, Chang YL (1991) A new heuristic for the *n*-job, *M*-machine flow-shop problem. Eur J Oper Res 52:194–202
97. Framinan JM, Leisten R, Ruiz-Usano R (2002) Efficient heuristics for flowshop sequencing with the objectives of makespan and flow time minimisation. Eur J Oper Res 141:559–569
98. Allahverdi A (2003) The two- and *m*-machine flowshop scheduling problems with bi-criteria of makespan and mean flow time. Eur J Oper Res 147:373–396
99. Allahverdi A (2004) A new heuristic for *m*-machine flowshop scheduling problem with bi-criteria of makespan and maximum tardiness. Comput Oper Res 31:157–180
100. Arroyo JEC, Armentano VA (2004) A partial enumeration heuristic for multiobjective flowshop scheduling problems. J Oper Res Soc 55:1000–1007
101. Framinan JM, Leisten R (2006) A heuristic for scheduling a permutation flowshop with makespan objective subject to maximum tardiness. Int J Prod Econ 99:28–40
102. Gupta JND, Hennig K, Werner F (2002) Local search heuristics for two-stage flow shop problems with secondary criterion. Comput Oper Res 29:123–149
103. Geiger MJ (2007) On operators and search space topology in multiobjective flow shop scheduling. Eur J Oper Res 181:195–206
104. Haq AN, Ramanan TR (2006) A bi-criterian flow shop scheduling using artificial neural network. Int J Adv Manuf Technol 30:1132–1138
105. Shi RF, Zhou H (2007) Escalating evolutionary algorithm with application to bi-objective flow shop scheduling problems. J Manage Sci 10(5):11–20 (in Chinese)
106. Zhang CK, Li XF, Shao HH, Ren DX (2002) Bi-directional simulation approach for multi-objective scheduling problem of hybrid flow shop. J Shanghai Jiaotong Univ 36(4):547–550 (in Chinese)
107. Wei Z, Xu XF, Deng SC (2006) Evolutionary algorithm for solving multi-objective hybrid flow-shop scheduling problem. Journal of Nanjing University of Science and Technology 30 (3):327–331 (in Chinese)
108. Behnamian J, Fatemi Ghomi SMT, Zandieh M (2009) A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. Expert Syst Appl 36:11057–11069
109. Figueira JR, Liefooghe A, Talbi EG, Wierzbicki AP (2010) A parallel multiple reference point approach for multi-objective optimization. Eur J Oper Res 205:390–400
110. Sawik T (2007) A lexicographic approach to bi-objective scheduling of single- period orders in make-to-order manufacturing. Eur J Oper Res 180(3):1060–1075
111. Janiak A, Kozan E, Lichtenstein M, Oguz C (2007) Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion. Int J Prod Econ 105(2):407–424
112. Javadi B, Saidi-Mehrabad M, Haji A, Mahdavi I, Jolai F, Mahdavi-Amiri N (2008) No-wait flow shop scheduling using fuzzy multi-objective linear programming. J Franklin Inst 345:452–467
113. Jungwattanakit J, Reodecha M, Chaovalitwongse P, Werner F (2006) Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. Int J Adv Manuf Technol 37(3–4):354–370
114. Rahimi-Vahed AR, Javadi B, Rabbani M, Tavakkoli-Moghaddam R (2008) A multi-objective scatter search for a bi-criteria no-wait flow shop scheduling problem. Eng Optim 40 (4):331–346
115. Rahimi-Vahed A, Dangchi M, Rafiei H, Salimi E (2009) A novel hybrid multi-objective shuffled frog-leaping algorithm for a bi-criteria permutation flow shop scheduling problem. Int J Adv Manuf Technol 41:1227–1239
116. Behnamian J, Fatemi Ghomi SMT (2010) Hybrid flowshop scheduling with machine and resource-dependent processing times. Appl Math Modell 35:1107–1123

117. Naderi B, Tavakkoli-Moghaddam R, Khalili M (2010) Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan. Knowl Based Syst 23:77–85
118. Coello Coello CA (2000) Handling preferences in evolutionary multiobjective optimization: a survey. 2000 Congress on Evolutionary Computation 1. IEEE Service Center, Piscataway, New Jersey, pp 30–37
119. Cvetković D, Parmee IC (2002) Preferences and their application in evolutionary multiobjective optimization. IEEE Trans Evol Comput 6(1):42–57
120. Junker U (2004) Preference-based search and multi-criteria. Ann Oper Res 130:75–115
121. Ishibuchi H, Nojima Y, Narukawa K, Doi T (2006) Incorporation of decision maker's preference into evolutionary multiobjective optimization algorithms. Proceedings of Genetic and Evolutionary Computation Conference—GECCO, New York, pp 741–742
122. Zitzler E, Thiele L, Bader J (2006) SPAM: set preference algorithm for multiobjective optimization. Proceedings of Conference on Parallel Problem Solving from Nature (PPSN X), Springer, pp 847–858
123. Luque M, Miettinen K, Eskelinen P, Ruiz F (2009) Incorporating preference information in interactive reference point methods for multiobjective optimization. Omega-Int J Manage S 37:450–462
124. Thiele L, Miettinen K, Korhonen PJ, Molina J (2009) A preference-based evolutionary algorithm for multiobjective optimization. Evol Comput 17(3):411–436
125. Wang Y, Yang Y (2009) Particle swarm optimization with preference order ranking for multi-objective optimization. Inf Sci 179:1944–1959
126. Cheng TCE, Wang G (2000) An improved heuristic for two-machine owshop scheduling with an availability constraint. Oper Res Lett 26:223–229
127. Allaoui H, Artiba A (2006) Scheduling two-stage hybrid flow shop with availability constraints. Comput Oper Res 33:1399–1419
128. Breit J (2006) A polynomial-time approximation scheme for the two-machine flow shop scheduling problem with an availability constraint. Comput Oper Res 33:2143–2153
129. Voβ S, Witt A (2007) Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: a real-world application. Int J Prod Econ 105:445–458
130. Wang X, Cheng TCE (2007) Heuristics for two-machine flowshop scheduling with setup times and an availability constraint. Comput Oper Res 34:152–162
131. Ruiz R, Sivrikaya-Serifoğlu F, Urlings T (2006) Modeling realistic hybrid flexible flowshop scheduling problems. Comput Oper Res 35:1151–1175
132. Low C, Hsu CJ, Su CT (2006) A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines. Comput Oper Res 35:845–853
133. Luo H, Huang GQ, Zhang YF, Dai QY, Chen X (2009) Two-stage hybrid batching flowshop scheduling with blocking and machine availability constraints using genetic algorithm. Rob Com-Int Manuf 25:962–971
134. Allaoui H, Artiba A (2004) Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. Comput Ind Eng 47:431–450
135. Reklaitis GV, Ravindran A, Ragsdell KM (1983) Engineering optimization methods and applications. Wiley, New York
136. Deb K (1995) Optimization for engineering design: algorithms and examples. Prentice-Hall, New Delhi
137. Davoudpour H, Ashrafi M (2009) Solving multi-objective SDST flexible flow shop using GRASP algorithm. Int J Adv Manuf Technol 44:737–747