REVIEW ON JOB-SHOP AND FLOW-SHOP SCHEDULING USING MULTI CRITERIA DECISION MAKING

Sultana Parveen* and Hafiz Ullah
Department of Industrial and Production Engineering
Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh
*Corresponding email: sparveen@ipe.buet.ac.bd

Abstract: *Scheduling is widely studied and complex combinatorial optimization problems. A vast amount of research has been performed in this particular area to effectively schedule jobs for various objectives. The multi-criteria scheduling problem is one of the main research subjects in the field of modern manufacturing where most of them are considered as NP-hard. This paper discusses the more recent literature on scheduling using multi criteria decision making (MCDM). This article addresses both job-shop and flow-shop scheduling problem.*

Key Words: *Flow-shop scheduling, Genetic algorithms, Job-shop scheduling, Multi-objective optimization.*

INTRODUCTION

Scheduling is broadly defined as the process of assigning a set of tasks to resources over a period of time[1] or it may be defined as ''the allocation of resources over time to perform a collection of tasks''[2]. Scheduling problems in their simple static and deterministic forms are extremely simple to describe and formulate, but are difficult to solve because they involve complex combinatorial optimization. For example, if n jobs are to be performed on m machines, there are potentially $(n!)^m$ sequences, although many of these may be infeasible due to various constraints.

MULTI OBJECTIVE SCHEDULING

Single criterion is deemed as insufficient for real and practical applications. Multi-objective optimization is without a doubt a very important research topic not only because of the multi-objective nature of most real-world problems, but also because there are still many open questions in this area. Over the last decade, multi-objective optimization has received a big impulse in Operations Research. Some new techniques have been developed in order to deal with functions and real-world problems that have multiple objectives, and many approaches have been proposed. In this section some important terms are defined. At first the structure of the flow-shop scheduling has been started and then several scheduling terms have been discussed. Finally, some number of performance measure used in this context has been introduced.

Flow-shop Scheduling

A flow-shop is a shop design in which machines are arranged in series Jobs begin processing on an initial machine, proceed through several intermediary machines, and conclude on a final machine[3]. The series arrangement implies a linear structure to the shop, as illustrated in Figure 1. This figure shows a pure flow-shop in which jobs must be processed on each machine in exactly the same order. A general flow-shop is somewhat different, in that a job may skip a particular machine. For instance, although every job must proceed from left to right in Figure 1, some jobs may go from machine 1 to, say, machine 3 and then machine 4.

Job-shop Scheduling

A job-shop does not have the same restriction on workflow as a flow-shop. In a job-shop, jobs can be processed on machines in any order. The usual job-shop, from a research standpoint, is one in which there are m machines and n jobs to be processed. Each job requires m operations, one on each machine, in a specific order, but the order can be different for each job[3]. Real job-shops are more complicated. Jobs may not require all in machines; and yet they may have to visit some machines more than once. Clearly, workflow is not unidirectional in a job-shop. Any given machine may observe new jobs arriving from outside the shop (as new inputs), and from other machines within the shop (as WIP), the same machine may be the last machine for a particular job, or it may be an intermediate

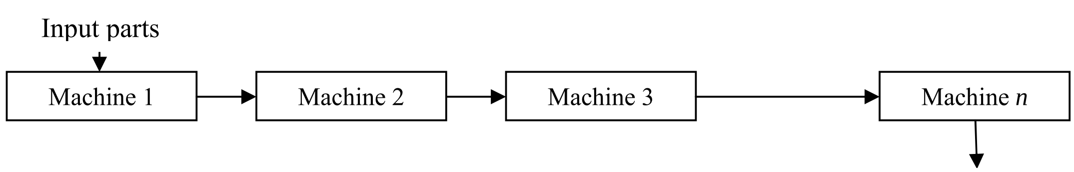

Figure 1. A pure flow-shop

processing step. Thus, the workflow can be illustrated as in Figure 2.

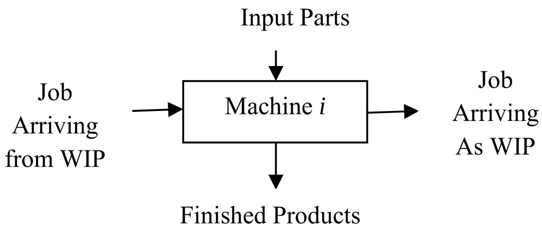

Figure 2. One machine in job shop

Flow time (F_j). It is the amount of time job *j* spends in system.
Makespan. It is the total amount of time for all jobs to finish processing
Lateness (L_j). It is the amount of time by which the completion time differs from the due date. Positive lateness of a job means job is completed after the due date[4].
Tardiness (T_j). It is the lateness of the job *j* if it fails to meet its due date, or zero. *i.e* $T_j = \max\{0, L_j\}$
The different measures of performances which are used in scheduling are listed below with their formulas[5].

Mean flow time: $\overline{F} = \frac{1}{n}\sum_{j=1}^{n} F_j$

Mean tardiness: $\overline{T} = \frac{1}{n}\sum_{j=1}^{n} T_j$

Maximum flow time: $F_{\max} = \max_{1\le j\le n}\{F_j\}$

Maximum tardiness: $T_{\max} = \max_{1\le j\le n}\{T_j\}$

Number of tardy job: $N_T = \sum_{j=1}^{n} f(T_j)$

where $f(T_j) = 1$, If $T_j > 0$ and $f(T_j) = 0$, otherwise.

BACKGROUND ON MULTI-CRITERIA SCHEDULING (FLOW-SHOP)

In the multi-objective case, the majority of studies use the simpler "a priori" approach where multiple objectives are weighted into a single one. The main problem in this method is that the weights for each objective must be given. The "a posteriori" multi-objective approach is more complex since in this case, there is no single optimum solution, but rather lots of "optimum" solutions. The majority of the literature for the Permutation Flow-shop Problem or PFSP is centered on a single optimization criterion or objective.

The literature on multi-objective optimization is plenty. However, the multi-objective Permutation Flow-shop Problem or PFSP field is relatively scarce, especially when compared against the number of papers published for this problem that consider one single objective. The few proposed multi-objective methods for the PFSP are mainly based on evolutionary optimization and some in local search methods like simulated annealing or tabu search. It could be argued that many reviews have been published about multi-objective scheduling. Review by Nagar *et al.*[6] is mostly centered on single machine problems. As a matter of fact there are only four survey papers related with flow-shop. In another review by T'kindt and Billaut[7] reviewed 15 flow-shop papers where most of them are about the specific two machine case. Another review is given by Jones *et al.*[8]. However, this is more a quantification of papers in multi-objective optimization. Finally, the more recent review of Hoogeveen[9] contains mainly results for one machine and parallel machines scheduling problems. The papers reviewed about flow-shop scheduling are all restricted to the two machine case. For all these reasons, this paper provides a complete and comprehensive review about multi-objective flow-shop and job-shop. In the following, the notation of T'kindt and Billaut[10] will be used to specify the technique and objectives studied by each reviewed paper. For example, a weighted makespan and total tardiness bi-criteria flow-shop problem is denoted as $F//F_l(C_{max}, T)$. For more details, the reader is referred to T'kindt and Billaut[7] or T'kindt and Billaut[10].

Lexicographic Approaches

Lexicographical approaches have been also been explored in the literature. Daniels and Chambers[11] proposed a constructive heuristic for the *m* machine flow-shop where makespan is minimized subject to a maximum tardiness threshold, a problem denoted by $F/prmu/\varepsilon(C_{max}/T_{max})$.This heuristic along with the one of Chakravarthy and Rajendran[12] is compared with a method recently proposed in Framinan and Leisten[13]. In this later paper, the newly proposed heuristic is shown to outperform the methods of Daniels and Chambers[11] and Chakravarthy and Rajendran[12] both on quality and on the number of feasible solutions found. A different set of objectives is considered in Rajendran[14] were the authors minimize total flowtime subject to optimum makespan value in a two machine flow-shop. Such an approach is valid for the PFSP problem since the optimum makespan can be obtained by applying the well known algorithm of Johnson[15]. Rajendran proposes a branch and bound (B&B) method together with some heuristics for the problem. However, the proposed methods are shown to solve 24 jobs maximum. In Neppalli *et al.*[16] two genetic algorithms were proposed for solving the two machine bicriteria flow-shop problem also in a lexicographical way as in Rajendran[14]. The first algorithm is based in the VEGA (Vector Evaluated Genetic Algorithm) of Schafer[17]. In this algorithm, two subpopulations are maintained (one for each objective) and are combined by the selection operator for obtaining new solutions. In the second GA, referred to as the weighted criteria approach, a linear combination of the two criteria is considered. This weighted sum of objectives is used as the fitness value. The same problem is studied by Gupta *et al.*[18] where a tabu search is employed. This algorithm is finely-tuned by means of statistical experiments and shown to outperform some of the earlier existing methods. Gupta *et al.*[19] present some local search procedures and three metaheuristics for a two machine flow-shop. The methods developed are simulated annealing (SA), threshold accepting and tabu search. The criteria to optimize are composed of several lexicographic pairs involving makespan, weighted flowtime and weighted tardiness. The proposed methods are compared against the GA of Neppalli *et al.*[16] and the results discussed. Gupta *et al.*[20] proposed nine heuristics for the two machine case minimizing flowtime subject to optimum makespan, i.e., Lex (C_{max}, F). The authors identify some polynomially solvable cases and carry out a comprehensive analysis of the proposed heuristics. Insertion based methods are shown to give the best results. The same problem is approached by T'kindt *et al.*[21] where the authors propose an ant colony optimization (ACO) algorithm. The method is compared against a heuristic from T'kindt *et al.*[22] and against other single objective methods form the literature. Although in some cases is slower, the proposed ACO method is shown to give higher quality results. T'kindt *et al.*[22] work with the same problem. The authors propose a B&B method capable of solving instances of up to 35 jobs in a reasonable time. Some heuristics are also provided. Venditti[23] addresses a practical scheduling problem arising in the packaging department of a pharmaceutical industrial plant. The objective functions are minimization of makespan and maximum tardiness in lexicographic order. Representing a solution with a directed graph allows devising an effective tabu search algorithm to solve the problem. Computational experiments, carried on real and randomly generated instances, show the effectiveness of this approach.

Weighted Objectives Approaches

Most studies make use of the "a priori" approach which was mentioned earlier. This means that objectives are weighted (mostly linearly) into a single combined criterion. After this conversion, most single objective algorithms can be applied. Nagar *et al.*[6] proposed a B&B procedure for solving a two machine flow-shop problem with a weighted combination of flowtime and makespan as objective. The algorithm initializes the B&B tree with an initial feasible solution and an upper bound, both obtained from a greedy heuristic. This algorithm was able to find the optimal solutions of problems with two machines and up to 500 jobs but only under some strong assumptions and data distributions. The same authors use this B&B in Nagar *et al.*[24] as a tool for providing the initial population in a GA. The hybrid B&B+GA approach is tested for the same two-job bi-criteria flow-shop and it is shown to outperform the pure B&B and GA algorithms. Another GA is presented in Sridhar and Rajendran[25] for makespan and flowtime, including also idle time as a third criterion. The algorithm uses effective heuristics for initialization. Cavalieri and Gaiardelli[26] study a realistic production problem that they modelize as a flow-shop problem with makespan and tardiness criteria. Two genetic algorithms are proposed where many of their parameters are adaptive. Yeh[27] proposes another B&B method that compares favorably against that of Nagar *et al.*[6]. For un-structured problems, Yeh's B&B is able to solve up to 14-job instances in less time than the B&B of Nagar *et al.*[6]. The same author improved this B&B in Yeh[28] and finally proposed a hybrid GA in Yeh[29] showing the best results among all previous work. Note that all these papers of Yeh deal with the specific two machine case only. Lee and Chou[30] proposed heuristic methods and a mixed integer programming model for the *m* machine problem combining makespan and flowtime objectives. Their study shows that the integer programming approach is only valid for very small instances. A very similar work and results was given in a paper by the same authors[31]. Sivrikaya-Şerifoğlu and Ulusoy[32] presented three B&B algorithms and two heuristics for the two machine flow-shop with makespan and flowtime objectives. All these methods are compared among them in a series of experiments. The largest instances solved by the methods contain 18 jobs. A linear combination of makespan and tardiness is studied in Chakravarthy and Rajendran[12] but in this case a SA algorithm is proposed. Chang *et al.*[33] studies the gradual-priority weighting approach in place of the variable weight approach for genetic and genetic local search methods. These two methods are related to those of Murata *et al.*[34] and Ishibuchi and Murata[35], respectively. In numerical experiments, the gradual-priority weighting approach is shown superior. Framinan *et al.*[36] proposed several heuristics along with a comprehensive computational evaluation for the *m* machine makespan and flowtime flow shop problem. Allahverdi[37] also studies the same objectives. A total of 10 heuristics are comprehensively studied in a computational experiment. Among the studied methods, three proposed heuristics from the author outperform the others. Several dominance relations for special cases are proposed as well.

A different set of objectives, namely makespan and maximum tardiness, are studied by Allahverdi[38]. Two variations are tested, in the first one; a weighted combination of the two objectives subject to a maximum tardiness value is studied. In the second, the weighted combination of criteria is examined. The author proposes a heuristic and compares it against the results of Daniels and Chambers[11] and Chakravarthy and Rajendran[18]. The proposed method is shown to outperform these two according to the results. Ponnambalam *et al.*[39] proposed a GA that uses some ideas from the Traveling Salesman Problem (TSP). The implemented GA is a straightforward one that just uses a weighted combination of criteria as the fitness of each individual in the population. The algorithm is not compared against any other method from the literature and just some results on small flow-shop instances are reported. Lin and Wu[40] focus on the two machine case with a weighted combination of makespan and flowtime. The authors present a B&B method that is tested against a set of small instances. The proposed method is able to find optimum solutions to instances of up to 15 jobs in all cases. Lemesre *et al.*[41] have studied the *m* machine problem with makespan and total tardiness criteria. A special methodology based on a B&B implementation, called two-phase method is employed. Due to performance reasons, the method is parallelized. As a result, some instances of up to 20 jobs and 20 machines are solved to optimality. However, the reported solving times for these cases are of seven days in a cluster of four parallel computers. Madhushini et. al.[42] minimizing the sum of weighted flowtime/sum of weighted tardiness/sum of weighted flowtime and weighted tardiness/sum of weighted flowtime, weighted tardiness and weighted earliness of jobs, with each objective considered separately. Lower bounds on the given objective are developed by solving an assignment problem. B&B algorithms are developed to obtain the best permutation sequence in each case. The proposed algorithms are evaluated by solving many randomly generated problems of different problem sizes.

Pareto Approaches

When focusing on the “a posteriori” approach the number of existing studies drops significantly. In the previously commented work of Daniels and Chambers[11], the authors also propose a B&B procedure for the C_{max} and T_{max} objectives that computes the Pareto global front for the case of two machines. A genetic algorithm was proposed by Murata *et al.*[34] which was capable of obtaining a Pareto front for makespan and total tardiness. This algorithm, referred to as MOGA (Multi objective genetic algorithm), applies elitism by copying a certain number of individuals in the non-dominated set to the next generation. The non-dominated solutions are kept externally in an archive. The algorithm selection is based on a fitness value given to each solution on the basis of a weighted sum of the objective’s values. The weights for each objective are randomly assigned at each iteration of the algorithm. The authors also test their proposed GA with three objectives including flowtime. Later, in Ishibuchi and Murata[35] the algorithm is extended by using a local search step that is applied to every new solution, after the crossover and mutation procedures.

Saym and Karabatı[43] studied a B&B algorithm that generates the optimum Pareto front for a two machine flow-shop with makespan and flowtime objectives. The experimental evaluation compares only against heuristics like those of Johnson[15] and Rajendran[14]. Some instances of up to 24 jobs are solved to optimality. Liao *et al.*[44] proposed a B&B algorithm for the two machine bi-criteria optimization problem, with the objectives of minimizing makespan and number of tardy jobs and also with the objectives of makespan and total tardiness. The lower bound values are obtained by means of the Johnson algorithm for makespan, and the Moore’s EDD (Early Due Date) algorithm for the number of tardy jobs. For each node of the partial schedules, two lower bounds are calculated using the above heuristics. The accepted non dominated schedules are kept in an external set. At the end of the algorithm, this set contains optimal Pareto front for the problem. Lee and Wu[45] also study the two machine case with B&B methods but with a combination of flowtime and total tardiness criteria. The authors do not compare their proposed approach with the literature and just report the results of their algorithm. A new type of genetic algorithm is shown by Bagchi[46]. This method is based on the NSGA method by Srinivas and Deb[47]. Some brief experiments are given for a single flow-shop instance with flowtime and makespan objectives. Murata *et al.*[48] improve the earlier MOGA algorithm of Murata *et al.*[34]. This new method, called CMOGA, refines the weight assignment. A few experiments with makespan and total tardiness criteria are conducted. The new CMOGA outperforms MOGA in the experiments carried out. Ishibuchi *et al.*[49] present a comprehensive study about the effect of adding local search to their previous algorithm[35]. The local search is only applied to good individuals and by specifying search directions. This form of local search was shown to give better solutions for many different multi-objective genetic algorithms. In Loukil *et al.*[50] many different scheduling problems are solved with different combinations of objectives. The main technique used is a multi-objective tabu search (MOTS). The paper contains a general study involving single and parallel machine problems as well. Later, in Loukil *et al.*[51], a similar study is carried out, but in this case the multi-objective approach employed is the simulated annealing algorithm (MOSA). A B&B approach is also shown by Toktaş *et al.*[52] for the two machine case under makespan and maximum earliness criteria. To the best of our knowledge, such combination of objectives has not been studied in the literature before. The procedure is able to solve problems of up to 25 jobs. The authors also propose a heuristic method. Suresh and Mohanasundaram[53] propose a Pareto-based simulated annealing algorithm for makespan and total flowtime criteria. The proposed method is compared against that of Ishibuchi *et al.*[49] and against an early version of the SA proposed later by Varadharajan and Rajendran[54]. The results, shown only for small problems of up to 20 jobs, show the proposed algorithm to be better on some specific performance metrics. Arroyo and Armentano[55] studied heuristics for several two and three objective combinations among makespan, flowtime and maximum tardiness. For two machines, the authors compare the heuristics proposed against the existing B&B methods of Daniels and Chambers[11] and Liao *et al.*[44]. For the general m machine case, the authors compare the results against those of Framinan *et al.*[36]. The results favor the proposed method that is also shown to improve the results of the GA of Murata *et al.*[34] if used as a seed sequence. The same authors developed a tabu search for the makespan and maximum tardiness objectives in Armentano and Arroyo[56]. The algorithm includes several advanced features like diversification and local search in several neighborhoods. For the two machine case, again the proposed method is compared against Daniels and Chambers[11] and for more than two machines against Ishibuchi and Murata[35]. The proposed method is shown to be competitive in numerical experiments. In a more recent paper Arroyo and Armentano[57] carry out a similar study but in this case using genetic algorithms as solution tools. Although shown to be better than other approaches, the authors do not compare this GA with their previous methods. Makespan and total flowtime are studied by Varadharajan and Rajendran[54] with the help of simulated annealing methods. These algorithms start from heuristic solutions that are further enhanced by improvement schemes. Two versions of these SA (MOSA and MOSA-II) are shown to outperform the GA of Ishibuchi and Murata[35]. Pasupathy *et al.*[58] have proposed a Pareto-archived genetic algorithm with local search and have tested it with the makespan and flowtime objectives. The authors test this approach against Ishibuchi and Murata[35] and Chang *et al.*[33]. Apparently, the newly proposed GA performs better under some limited tests. Melab *et al.*[59] propose a grid-based parallel genetic algorithm aimed at obtaining an accurate Pareto front for makespan and total tardiness criteria. While the authors do not test their approach against other existing algorithms, the results appear promising. However, the running days are of 10 days in a set of computers operating as a grid. More recently, Rahimi-Vahed and Mirghorbani[60] have proposed a complex hybrid multi-objective particle swarm optimization (MOPS) method. The considered criteria are flowtime and total tardiness. In this method, an elite tabu search algorithm is used as an initialization of the swarm. A parallel local search procedure is employed as well to enhance the solution represented by each particle. This complex algorithm is compared against the SPEAII multi-objective genetic algorithm of Zitzler *et al.*[61]. MOPS yields better results than SPEAII according to the reported computational experimentation albeit at a higher CPU time requirements. Finally, Geiger[62] has published an interesting study where the topology of the multi-objective flow-shop problem search space is examined. Using several local search algorithms, the author analyzes the distribution of several objectives and tests several combinations of criteria.

Goal Programming and Other Approaches

There are some cases of other multi-objective methodologies like goal programming. For example, Selen and Hott[63] proposed a mixed-integer goal programming formulation for a bi-objective PFSP dealing with makespan and flowtime criteria. As with every goal programming method, a minimum desired value for each objective has to be introduced. Later, Wilson[64] proposed a different model with fewer variables but a larger number of constraints. However, both models have the same number of binary variables. The comparison between both models results in the one of Selen and Hott[63] being better for problems with $n \geq 15$. Many algorithms in the literature have been proposed that do not explicitly consider many objectives as in previous sections. For example, Ho and Chang[65]propose a heuristic that is specifically devised for minimizing machine idle time in a m machine flow-shop. Although the heuristic does not allow for setting weights or threshold values and does not work with the Pareto approach either, the authors test it against a number of objectives. A similar approach is followed by Gangadharan and Rajendran[66] where a simulated annealing is proposed for the m machine problem and evaluated under makespan and flowtime criteria. Along with the SA method, two heuristics are also studied. Rajendran[67]proposes a heuristic for the same problem dealt with in Ho and Chang[65]. After a comprehensive numerical experimentation, the new proposed heuristic is shown to be superior to that of Ho and Chang's. A very similar study is also presented by the same author in Rajendran[68]. Ravindran *et al.*[69] present three heuristics aimed at minimizing makespan and flowtime. The authors test the three proposed method against the heuristic of Rajendran[67] but using only very small instances of 20 jobs and 20 machines maximum. The three heuristics appear to outperform Rajendran's albeit slightly. It is difficult to draw a line in these types of papers since many authors test a given proposed heuristic under different objectives. However, the heuristics commented above were designed with several objectives in mind and therefore we have included them in the review.

In total, 54 papers have been reviewed. Among them, 21 deal with the specific two machine case. From the remaining 33 that study the more general m machines, a total of 16 uses the "a posteriori" or Pareto based approach. The results of these methods are not comparable for several reasons. First, the authors do not always deal with the same combination of criteria. Second, comparisons are many times carried out with different benchmarks and against heuristics or older methods. Last and most importantly, the quality measures employed are not appropriate as recent studies have shown.

Multi-objective Quality Measures

As commented in previous sections, comparing the solutions of two different Pareto approximations coming from two algorithms are not straightforward. Two approximation sets A and B can be even incomparable. Recent studies like those of Zitzler *et al.*[70], Paquete[71] or more recently, Knowles *et al.*[72] are an example of the enormous effort being carried out in order to provide the necessary tools for a better evaluation and comparison of multi-objective algorithms. However, the multi-objective literature for the PFSP frequently uses quality measures that have been shown to be misleading. For example, in the two most recent papers reviewed[60, 62] some metrics like generational distance or maximum deviation from the best Pareto front are used. These metrics, among other ones are shown to be non Pareto-compliant in the study of Knowles *et al.*[72], meaning that they can give a better metric for a given Pareto approximation front B and worse for another front A even in a case where A < B. What is worse, in the comprehensive empirical evaluation of quality measures given in Knowles *et al.*[72], it is shown that the most frequently used measures are non Pareto-compliant and are demonstrated to give wrong and misleading results more often than not. Therefore, special attention must be given to the choice of quality measures to ensure sound and generalizable results. Knowles *et al.*[72] propose three main approaches that are safe and sound. The first one relies on the Pareto dominance relations among sets of solutions. It is possible to rank a given algorithm over another based on the number of times the resulting Pareto approximation fronts dominate (strong, regular or weakly) each other. The second approach relies on quality indicators, mainly the hypervolume I_H and the Epsilon indicators that were already introduced in Zitzler and Thiele[73] and Zitzler *et al.*[70], respectively. Quality indicators usually transform a full Pareto approximation set into a real number. Lastly, the third approach is based on empirical attainment functions. Attainment functions give, in terms of the objective space, the relative frequency that each region is attained by the approximation set given by an algorithm. These three approaches range from straightforward and easy to compute in the case of dominance ranking to the not so easy and computationally intensive attainment functions. According to Knowles *et al.*[72], I_H and $I^1{}_{\varepsilon}$ are Pareto-compliant and represent the state-of-the-art as far as quality indicators are concerned. Additionally, combining the analysis of these two indicators is a powerful approach since if the two indicators provide contradictory conclusions for two algorithms; it means that they are incomparable. The hypervolume indicator I_H, first introduced by Zitzler and Thiele[73] just measures the area (in the case of two objectives) covered by the approximated Pareto front given by one algorithm. A reference point is used for the two objectives in order to bound this area. A greater value of I_H indicates both a better convergence to as well as a good coverage of the optimal Pareto front. Calculating the hypervolume can be costly and I use the algorithm proposed in Deb[74]. This algorithm already calculates a normalized and scaled value. The binary epsilon indicator I_ε proposed initially by Zitzler *et al.*[70].

Computational Evaluation

This work implemented not only algorithms specifically proposed for the multi-objective PFSP but also many other multi-objective optimization algorithms. In these cases, some adaptation has been necessary. In the following discussed the algorithms that have been considered.

The MOGA algorithm of Murata *et al.*[34] was designed to tackle the multi objective flow-shop problem. It is a simple genetic algorithm with a modified selection operator. During this selection, a set of weights for the objectives are generated. In this way the algorithm tends to distribute the search toward different directions. The authors also incorporate an elite preservation mechanism which copies several solutions from the actual Pareto front to the next generation. Chakravarthy and Rajendran[12] presented a simple simulated annealing algorithm which tries to minimize the weighted sum of two objectives. The best solution between those generated by the Earliest Due Date (EDD), Least Static Slack (LSS) and NEH methods is selected to be the initial solution[74]. The adjacent interchange scheme (AIS) is used to generate a neighborhood for the actual solution. Notice that this algorithm, referred to as SA_Chakravarty, is not a real Pareto approach since the objectives are weighted. Bagchi[45] proposed a modification of the well known NSGA procedure and adapted it to the flow-shop problem. This algorithm, referred to as ENGA, differentiates from NSGA in that it incorporates elitism. In particular, the parent and offspring populations are combined in a unique set, then a non-dominated sorting is applied and the 50% of the non-dominated solutions are copied to the parent population of the following generation. Murata *et al.*[47] enhanced the original MOGA of Murata *et al.*[33]. A different way of distributing the weights during the run of the algorithm is presented. The proposed weight specification method makes use of a cellular structure which permits to better select weights in order to find a finer approximation of the optimal Pareto front. Suresh and Mohanasundaram[52] proposed a Pareto archived simulated annealing (PASA) method. A new perturbation mechanism called "segment-random insertion (SRI)" scheme is used to generate the neighborhood of a given sequence. An archive containing the non-dominated solution set is used. A randomly generated sequence is used as an initial solution. The SRI is used to generate a neighborhood set of candidate solutions and each one is used to update the archive set. A fitness function that is a scaled weighted sum of the objective functions is used to select a new current solution. A restart strategy and a reannealing method are also implemented. Armentano and Arroyo[55] developed a multiobjective tabu search method called MOTS. The algorithm works with several paths of solutions in parallel, each with its own tabu list. A set of initial solutions is generated using a heuristic. A local search is applied to the set of current solutions to generate several new solutions. A clustering procedure ensures that the size of the current solution set remains constant. The algorithm makes also use of an external archive for storing all the non-dominated solutions found during the execution. After some initial experiments we found that under the considered stopping criterion (to be detailed later), less than 12 iterations were carried out. Arroyo and Armentano[56] proposed a genetic local search algorithm with the following features: preservation of population's diversity, elitism (a subset of the current Pareto front is directly copied to the next generation) and usage of a multi-objective local search. The concept of Pareto dominance is used to assign fitness (using the non-dominated sorting procedure and the crowding measure both proposed for the NSGAII) to the solutions and in the local search procedure. A multi-objective simulated annealing (MOSA) is presented in Varadharajan and Rajendran[53]. The algorithm starts with an initialization procedure which generates two initial solutions using simple and fast heuristics. These sequences are enhanced by three improvement schemes and are later used, alternatively, as the solution of the simulated annealing method. MOSA tries to obtain non dominated solutions through the implementation of a simple probability function that attempts to generate solutions on the Pareto optimal front. The probability function is varied in such a way that the entire objective space is covered uniformly obtaining as many non-dominated and well dispersed solutions as possible. Varadharajan. Pasupathy *et al.*[57] proposed a genetic algorithm which we refer to as PGA_ALS. This algorithm uses an initialization procedure which generates four good initial solutions that are introduced in a random population. PGA_ALS handles a working population and an external one. The internal one evolves using a Pareto-ranking based procedure similar to that used in NSGAII. A crowding procedure is also proposed and used as a secondary selection criterion. The non-dominated solutions are stored in the external archive and two different local searches are then applied to half of archive's solutions for improving the quality of the returned Pareto front. Geiger[61] proposed a new algorithm is based on iterated local search which in turn relies on two main principles, intensification using a variable neighborhood local search and diversification using a perturbation procedure. The Pareto dominance relationship is used to store the non-dominated solutions. This scheme is repeated through successive iterations to reach favorable regions of the search space.

The multi-objective literature is marred with many interesting proposals, mainly in the form of evolutionary algorithms that have not been applied to the PFSP before. Therefore, in this section review some of these methods that have been re-implemented and adapted to the PFSP. Srinivas and Deb[46] proposed the well known non-dominated sorting genetic algorithm, referred to as NSGA. This method differs from a simple genetic algorithm only for the way the selection is performed. The non-dominated Sorting procedure (NDS) iteratively divides the en tire population into different Pareto fronts. The individuals are assigned a fitness value that depends on the Pareto front they belong to. Furthermore, this fitness value is modified by a factor that is calculated according to the number of individuals crowding a portion of the objective space. A sharing parameter σ share is used in this case. All other features are similar to a standard genetic algorithm. Zitzler and Thiele[72] presented another genetic algorithm referred to as SPEA. The most important characteristic of this method is that all non-dominated solutions are stored in an external population. Fitness evaluation of individuals depends on the number of solutions from the external population they dominate. The algorithm also incorporates a clustering procedure to reduce the size of the non-dominated set without destroying its characteristics. Finally, population's diversity is maintained by using the Pareto dominance relationship. Later, Zitzler *et al.*[60] proposed an improved SPEAII version that incorporates a different fine-grained fitness strategy to avoid some drawbacks of the SPEA procedure. Other improvements include a density estimation technique that is an adaptation of the *k*-th nearest neighbor method, and a new complex archive truncation procedure. Knowles and Corne[76] presented another algorithm called PAES. This method employs local search and a population archive. The algorithm is composed of three parts, the first one is the candidate solution generator which has an archive of only one solution and generates a new one making use of random mutation. The second part is the candidate solution acceptance function which has the task of accepting or discarding the new solution. The last part is the non-dominated archive which contains all the non-dominated solutions found so far. According to the authors, this algorithm represents the simplest nontrivial approach to a multi-objective local search procedure. In the same paper, the authors present an enhancement of PAES referred to as $(\mu+\lambda)$ −PAES. Here a population of μ candidate solutions is kept. By using a binary tournament, a single solution is selected and λ mutant solutions are created using random mutation. Hence, a $\mu + \lambda$ population is created and a dominance score is calculated for each individual. μ Individuals are selected to update the candidate population while an external archive of non-dominated solutions is maintained. Another genetic algorithm is proposed by Corne *et al.*[76]. This method, called PESA uses an external population EP and an internal one IP to pursuit the goal of finding a well spread Pareto front. A selection and replacement procedure based on the degree of crowding is implemented. A simple genetic scheme is used for the evolution of IP while EP contains the non-dominated solutions found. The size of the EP is upper bounded and a hyper-grid based operator eliminates the individuals in the more crowded zones. Later, in Corne *et al.*[78] an enhanced PESAII method is provided. This algorithm differs from the preceding one only in the selection technique in which the fitness value is assigned according to a hyperbox calculation in the objective space. In this technique, instead of assigning a selective fitness to an individual, it is assigned to the hyperboxes in the objective space which are occupied by at least one element. During the selection process, the hyperbox with the best fitness is selected and an individual is chosen at random among all inside the selected hyperbox. In Deb[79] an evolution of the NSGA was presented. This algorithm, called NSGAII, uses a new Fast Non-Dominated Sorting procedure (FNDS). Unlike the NSGA, here a rank value is assigned to each individual of the population and there is no need for a parameter to achieve fitness sharing. Also, a crowding value is calculated with a fast procedure and assigned to each element of the population.

The selection operator uses the rank and the crowding values to select the better individuals for the mating pool. An efficient procedure of elitism is implemented by comparing two successive generations and preserving the best individuals. This NSGAII method is extensively used in the multi objective literature for the most varied problem domains. Later, Deb *et al.*[80] introduced yet another GA called CNSGAII. Basically, in this algorithm the crowding procedure is replaced by a clustering approach. The rationale is that once a generation is completed, the previous generation has a size of P_{size} (parent set) and the current one (offspring set) is also of the same size. Combining both populations yields a $2P_{size}$ set but only half of them are needed for the next generation. To select these solutions the non-dominated sorting procedure is applied first and the clustering procedure second. Deb *et al.*[80] studied another different genetic algorithm. This method, called ε−MOEA uses two co-evolving populations, the regular one called P and an archive A. At each step, two parent solutions are selected, the first from P and the second from A. An offspring is generated, and it is compared with each element of the population P. If the offspring dominates at least a single individual in P then it replaces this individual. The offspring is discarded if it is dominated by P. The offspring individual is also checked against the individuals in A. In the archive population the ε-dominance is used in the same way.

Zitzler and Künzli [81] proposed another method called B−IBEA. The main idea in this method is defining the optimization goal in terms of a binary quality measure and directly using it in the selection process. B-IBEA performs binary tournaments for mating selection and implements environmental selection by iteratively removing the worst individual from the population and updating the fitness values of the remaining individuals. A ε−indicator is used. In the same work, an adaptive variation called A−IBEA is also presented. An adapted scaling procedure is proposed with the goal of making the algorithm's behavior independent from the tuning of the parameter *k* used in the basic B−IBEA version. Finally, Kollat and Reed[82] proposed also a NSGAII variation referred to as ε−NSGAII by adding ε−dominance archiving and adaptive population sizing. The ε parameter establishes the size of the grid in the objective space. Inside each cell of the grid no more than one solution is allowed. Furthermore, the algorithm works by alternating two phases. It starts using a very small population of 10 individuals and several runs of NSGAII are executed. During these runs all the non-dominated solutions are copied to an external set. When there are no further improvements in the current Pareto front, the second phase starts. Wang[83] devoted to some flow-shop scheduling problems with a learning effect. The objective is to minimize one of the two performance criteria, makespan and total flowtime. A heuristic algorithm with worst-case bound m for each criterion is given, where *m* is the number of machines. Furthermore, a polynomial algorithm is proposed for both of the special cases: identical processing time on each machine and an increasing series of dominating machines. An example is also constructed to show that the classical Johnson's rule is not the optimal solution for the two-machine flow-shop scheduling to minimize makespan with a learning effect. Tavakkoli-Moghaddam[84] investigates a novel multi-objective model for a no-wait flow-shop scheduling problem that minimizes both the weighted mean completion time and weighted mean tardiness. Obtaining an optimal solution for this type of complex, large-sized problem in reasonable computational time by using traditional approaches and optimization tools is extremely difficult. The authors presents a new hybrid multi-objective algorithm based on the features of a biological immune system (IS) and bacterial optimization (BO) to find Pareto optimal solutions for the given problem. Further, the efficiency of the proposed algorithm, based on various metrics, is compared against five prominent multi-objective evolutionary algorithms: PS-NC GA, NSGA-II, SPEA-II, MOIA, and MISA and find HMOIA outperforms the five foregoing algorithms, especially for large-sized problems. Yagmahan[85] consider the flow-shop scheduling problem with multi-objectives of makespan, total flow time and total machine idle time.

Ant colony optimization (ACO) algorithm is proposed to solve this problem. They compared the algorithm with solution performance obtained by the existing multi-objective heuristics and show that proposed algorithm is more effective and better than other methods compared. Allouche et. al. (2009)[86] proposes an aggregation procedure that integrates three different criteria to find the best sequence in a flow-shop production environment. The compromise programming model and the concept of satisfaction functions will be utilized to integrate explicitly the manager's preferences according to the deviations between the achievement and the aspiration levels of the following criteria: Makespan, total flow time and total tardiness. Yagmahan[87] present a multi-objective ant colony system algorithm (MOACSA), which combines ant colony optimization approach and a local search strategy in order to solve this scheduling problem. Its solution performance was compared with the existing multi-objective heuristics and get more efficient and better than other methods. Dugardin[88] focuses on the multi-objective resolution of a reentrant hybrid flow-shop scheduling problem (RHFS) and objectives are: the maximization of the utilization rate of the bottleneck and the minimization of the maximum completion time and solve this problem with a new multi-objective genetic algorithm called L-NSGA which uses the Lorenz dominance relationship. The results of L-NSGA are compared with NSGA2, SPEA2 and an exact method. A stochastic model of the system is proposed and used with a discrete event simulation module. A test protocol is applied to compare the four methods on various configurations of the problem. The comparison is established using two standard multi-objective metrics. The Lorenz dominance relationship provides a stronger selection than the Pareto dominance and gives better results than the latter. The computational tests show that L-NSGA provides better solutions than NSGA2 and SPEA2; moreover, its solutions are closer to the optimal front. The efficiency of our method is verified in an industrial field-experiment.

BACKGROUND ON MULTI-CRITERIA SCHEDULING (JOB-SHOP)

Mellor[89] discusses the literature on job-shop like sequencing problems. The effectiveness of evolutionary computation methodologies in the solution of multi-objective optimization problems has generated significant research interest in recent years. A number of evolutionary multiobjective optimization (EMO) methodologies have been developed and are being continuously improved in order to achieve better performance. These techniques have illustrated their competency against traditional multiobjective optimization techniques in the solution of this type of problems and are now considered to be a robust optimization tool in the hands of researchers and practitioners. An excellent introduction to the concepts of multiobjective optimization as well as a review of EMO techniques can be found[90].

Single-objective scheduling optimization problems have traditionally attracted considerable research interest from evolutionary computation researchers, since the encoding of solutions is straightforward and a number of well-tested recombination operators enhance the robustness of the optimization process[91]. While EMO research in the same area has not been as fruitful, optimization methodologies have been proposed during the last decade.

Udo[92] reports of a simulation study that investigates a dynamic approach to scheduling jobs in a multi-machine job-shop. The workload information of a job is used in different forms to evaluate the shop performance based on three measures: mean job lateness, percentage of tardy jobs and lateness variance. Different combinations of due-date assignment methods and sequencing rules are compared based on specific performance criteria. The results indicate that using the cumulative distribution function of workload information can yield a better performance than using a proportional function of workload information or ignoring shop congestion information. A few situations are identified in which workload information is not critical. Toker[93] consider the job-shop scheduling problem under a discrete non-renewable resource constraint. The authors assume that jobs have arbitrary processing times and resource requirements and there is a unit supply of the resource at each time period and develop an approximation algorithm for this problem and empirically test its effectiveness in finding the minimum makespan schedules. Mesghouni *et al.*[94] considered the typical job-shop scheduling problem with the primary objective of minimizing the makespan of all jobs to be processed. The solution methodology consisted of a Constrained Logic Programming algorithm that provided initial solutions for the evolutionary optimization process. Multicriteria analysis followed the identification of a set of solutions that satisfied the objective of minimum makespan. The PROMETHEE multicriteria analysis technique was employed for the identification of non-dominated solutions based on the minimization of makespan, minimization of the standard deviation of the workload of the resource, minimization of the mean completion time and the minimization of the standard deviation of completion time. The end-user was responsible for choosing among the alternative solutions generated and changing the objective weights according to his/her preferences. An interesting study on the solution of the job-shop scheduling problem using evolutionary computation algorithms was presented by Esquivel *et al.*[95] and showed that using the concepts of multi-recombination (Production of multiple offspring by the same pair of parents during crossover and choice of the best one) and incest prevention (recombination restricted only to individuals without common ancestors) evolutionary computation algorithms produce better results both in the single and multiobjective instances of the problem. For the multi objective case the author proposed both a sub population based approach, where each subpopulation optimized a separate objective and the combined population optimized an aggregated combination of the objectives considered, and a Pareto-ranking based approach utilizing the concept of elitism. Encouraging results were reported for both approaches; however, no comparison with alternative multiobjective optimization techniques was attempted. Thiagarajan[96] addresses the problem of scheduling in dynamic assembly job-shops with the consideration of jobs having different earliness, tardiness and holding costs. In the first phase of the study, relative costs, earliness and tardiness of jobs are considered, and the dispatching rules are presented in order to minimize the sum of weighted earliness and weighted tardiness of jobs. In the second phase of the study, the objective considered is the minimization of the sum of weighted earliness, weighted tardiness and weighted flowtime of jobs, and the dispatching rules are presented by incorporating the relative costs of earliness, tardiness and flowtime of jobs. Gao *et. al.*[97] addresses the fJSP problem with three objectives: min makespan, min maximal machine workload and min total workload and develop a new GA hybridized with an innovative local search procedure (bottleneck shifting) for the problem. The GA uses two representation methods to depict solution candidates of the fJSP problem. Advanced crossover and mutation operators are proposed to adapt to the special chromosome structures and the characteristics of the problem.

Cheng[98] study the problem of scheduling *n* deteriorating jobs on *m* identical parallel machines. Each job's processing time is a nondecreasing function of its start time. The problem is to determine an optimal combination of the due-date and schedule so as to minimize the sum of the due-date, earliness and tardiness penalties and showed that this problem is NP-hard, and present a heuristic algorithm to find near-optimal solutions for the problem. When the due-date penalty is 0, they also present a polynomial time algorithm to solve it. Vilcot[99] minimize the makespan and the maximum lateness, and they are interested in finding an approximation of the Pareto frontier. A fast and elitist genetic algorithm based on NSGA-II proposed for solving the problem. The initial population of this algorithm is either randomly generated or partially generated by using a tabu search algorithm that minimizes a linear combination of the two criteria. Both the genetic and the tabu search algorithms are tested and computational results show the interest of both methods to obtain an efficient and effective resolution method. Tavakkoli-Moghaddam[100] presents a fuzzy-neural approach for constraint satisfaction of a generalized job-shop scheduling problem (GJSSP) fuzzy processing times. It was an extension of recently developed research in a GJSSP where the processing time of operations was constant. But they assume that the processing time of jobs is uncertain. The proposed fuzzy-neural approach can be adaptively adjusted with weights of connections based on sequence resource and uncertain processing time constraints of the GJSSP during its processing. The computational results show that the proposed neural approach is able to find good solutions in reasonable time. Tay[101] solve, evaluate and employ suitable parameter and operator spaces for evolving composite dispatching rules using genetic programming, with an aim towards greater scalability and flexibility for the multi-objective flexible job-shop problems. Lei[102] present a particle swarm optimization for multi-objective job-shop scheduling problem. The objective is to simultaneously minimize makespan and total tardiness of jobs. They design a Pareto archive particle swarm optimization, in which the global best position selection is combined with the crowding measure-based archive maintenance. Manikas[103] demonstrate GA can be used to produce solutions in times comparable to common heuristics but closer to optimal. Changing criteria or their relative weights does not affect the running time, nor does it require programming changes. Therefore, a GA can be easily applied and modified for a variety of production optimization criteria in a job-shop environment that includes sequence-dependent setup times. Xing[104] presented a simulation model to solve the multi-objective flexible job-shop scheduling problem which was coded by Matlab and a special mathematical computation language. After modeling the pending problem, the model is validated by five representative instances based on practical data. Zhang[105] proposed a particle swarm optimization (PSO) algorithm and a tabu search (TS) algorithm are combined to solve the multi-objective FJSP with several conflicting and incommensurable objectives. PSO which integrates local search and global search scheme possesses high search efficiency. And, TS is a meta-heuristic which is designed for finding a near optimal solution of combinatorial optimization problems.

A hybrid metaheuristic method for the job-shop scheduling problem is proposed by Zobolas[106]. The optimization criterion is the minimization of makespan and the solution method consists of three components: a Differential Evolution-based algorithm to generate a population of initial solutions, a Variable Neighbourhood Search method and a Genetic Algorithm to improve the population; the latter two are interconnected. Computational experiments on benchmark data sets demonstrate that the proposed hybrid metaheuristic reaches high quality solutions in short computational times using fixed parameter settings. Zhu[107]proposed a cost-based job-shop problem (JIT-JSP). The objective of JIT-JSP is to minimize three costs: work-in-process holding cost of half-finished orders, inventory holding cost of finished orders and backorder cost of unfulfilled orders. A modified tabu search (MTS) method is developed to improve the schedule quality by searching the neighbourhood of a feasible schedule iteratively. The MTS method is comprised of three components that help to ensure a more effective searching procedure: neighbourhood structure, memory structure and filter structure. Computational results show that the MTS method significantly improves the initial schedule generated by an arbitrarily selected dispatching rule. Huang[108] is used to solve the job-shop scheduling problem using ant colony optimization (ACO) algorithm and compared with the solution obtained by LINGO, the ACO algorithm performs well in scheduling and uses less time to solve the problem.Dynamic job-shop scheduling that considers random job arrivals and machine breakdowns is studied by Adibi[109]. Considering an event driven policy rescheduling, is triggered in response to dynamic events by variable neighborhood search (VNS). A trained artificial neural network (ANN) updates parameters of VNS at any rescheduling point. Also, a multi-objective performance measure is applied as objective function that consists of makespan and tardiness.

A number of researchers have illustrated how the principles of EMO can be used for the solution of practical multiobjective optimization problems in the area of scheduling. Arumugam[110] reports a case study carried out in an Engineering industry manufacturing nineteen types of products against orders. The objective of this research was to select the sequencing rule that will optimize the combined performance of work-in-process inventory in monetary terms and delivery performance. Past studies in this area show that rules based on either the value of the orders or the value-processing time of the orders have not been investigated. This study evaluates the performance of the two sequencing rules based on the value and processing time of the orders: (i) Lowest Value Time rule (LVT) (ii) Highest Value Time rule (HVT). Incidental to this study, a few other sequencing rules were evaluated. To carry out the investigations, a simulator using GPSS (General Purpose Simulation System) was developed. Shaw & Fleming[111] considered a practical scheduling problem for a company that produces chilled ready mills. The MOGA evolutionary multiobjective optimization technique[112] was proposed for the simultaneous minimization of omissions, lateness and shift ends. A comparison of the proposed algorithm with typical weighted sum approach illustrated its ability to provide a wealth of potential solutions while maintaining its optimization ability. Tamaki *et al.*[113] presented a case study on the application of a multiobjective evolutionary computation methodology for the solution of a scheduling problem in a plastics forming plant. The problem was modeled as an unrelated parallel machines scheduling problem. A typical Pareto-based ranking technique with elitism was used during the evolutionary process with the objective of simultaneously minimizing the sum of idle time of every machine, the maximum tardiness of jobs and the makespan of jobs. A typical example of the algorithm's application was provided.

Finally, Khoo *et al.*[114] illustrated how a generic practical scheduler for a manufacturing production system can be built. Their proposed scheduler consisted of a database that provided scheduling data, an evolutionary optimizer that generated near-optimal schedules, and a Schedule-builder that was responsible for the transformation of any evolved sequence into a legal schedule. The scheduler was capable of handling various types of scheduling problems (job-shop, flow-shop, cellular manufacturing) with various types of objectives and constraints. The scheduler also provided a user interface with front-end analysis capabilities. One of the features of the proposed scheduler was its ability to handle multiple objectives. However, optimization was not achieved in the typical Pareto-ranking fashion. A schedule was initially generated that was optimal with regards to the makespan objective. The schedule builder was responsible for transforming this schedule in order to simultaneously minimize the tardiness objective.

CONCLUSION

The contribution of the study conducted a comprehensive survey of the multi-objective literature for both the flow-shop and the job-shop problem, which is one of the most common and thoroughly studied problems in the scheduling field. The papers surveyed include exact as well as heuristic techniques for many different multi-objective approaches. To comparative evaluation not only includes scheduling specific algorithms but also adaptations of other general methods proposed in the multi-objective optimization literature. Finally multi-objective studies for more complex scheduling problems with additional characteristics like setup time and parallel machine are scarce and new algorithms for such problems are desirable in practice.

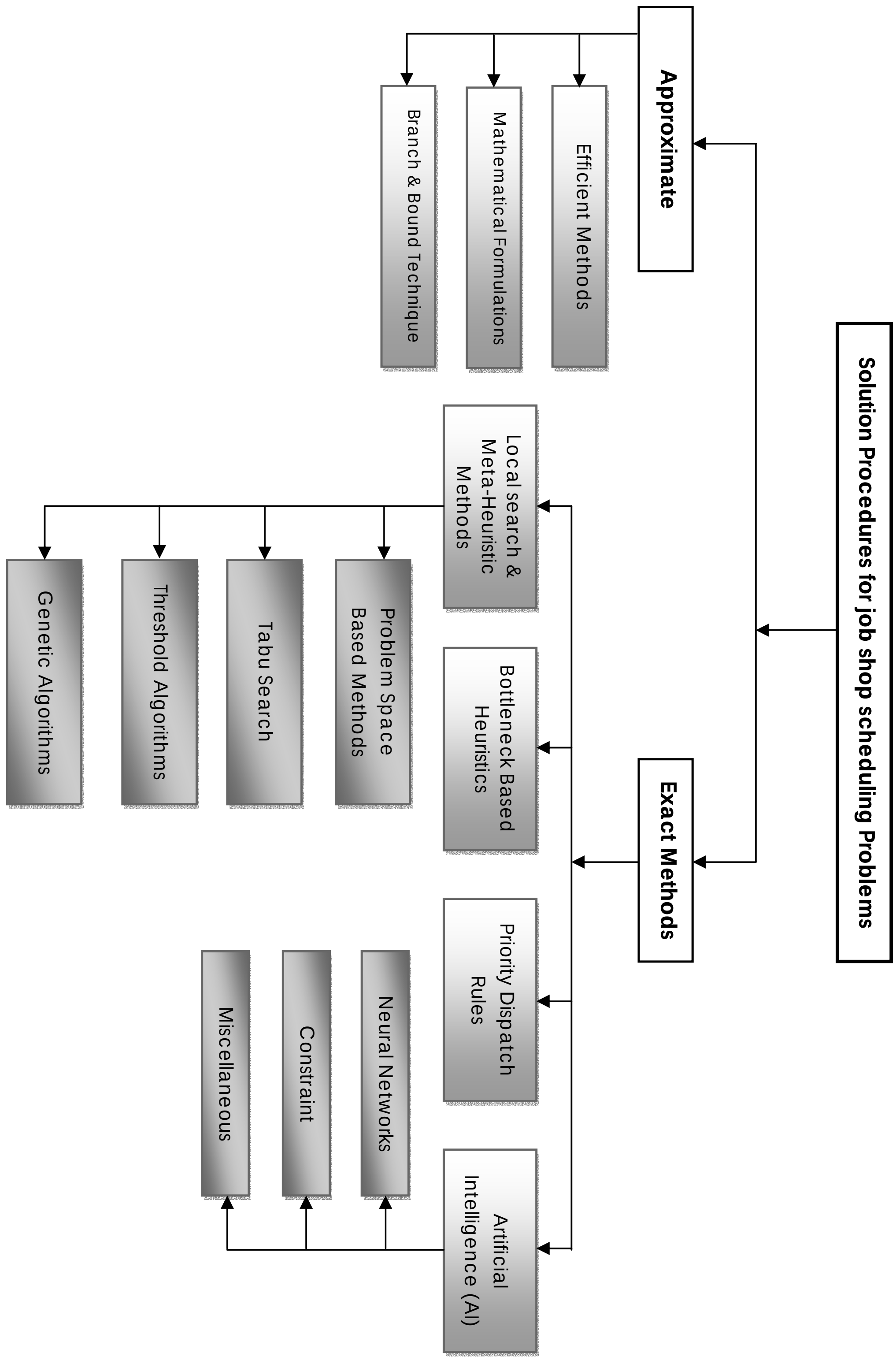

Figure 3. Solution Approaches for Job Shop Problems

REFERENCES

[1] Pinedo, M., 2001, Scheduling: Theory, Algorithms, and Systems, Prentice-Hall, New York.

[2]. Baker, K.R., 1974. Introduction to Sequencing and Scheduling, Wiley and Sons, New York.

[3]. Silver, E. A., Pyke, D. F., Peterson, R., 1998, Inventory Management and Production Planning and Scheduling Management, John Wiley & sons, USA.

[4]. Hopp, W. J., Spareman, M. L., 2001, Factory Physics Foundation of Manufacturing Management, McGraw-Hill, Singapore.

[5]. Panneerselvam, R., 2006, Production and Operation Management, Prentice Hall, New Delhi, India.

[6]. Nagar, A., Heragu, S. S., Haddock, J., 1995, “A branch-and-bound approach for a two-machine flow-shop scheduling problem,” Journal of the Operational Research Society, 46, pp. 721–734.

[7]. T’kindt, V., Billaut, J. C., 2001, “Multicriteria scheduling problems: A survey,” RAIRO Recherche opérationnelle - Operations Research, 35, pp. 143–163.

[8]. Jones, D. F., Mirrazavi,S. K., Tamiz, M., 2002, “Multi-objective meta-heuristics: An overview of the current state-of-the-art,” European Journal of Operational Research, 137, pp. 1–9.

[9]. Hoogeveen, H. 2005. “Multicriteria scheduling.” European Journal of Operational Research 167, 592–623.

[10]. T’kindt, V., Billaut, J. C., 2002, “Multicriteria scheduling: Theory, models and algorithms,” Springer, Berlin.

[11]. Daniels, R. L., Chambers, R.J., 1990, “Multiobjective flow-shop scheduling,” Naval research logistics, 37, pp. 981–995.

[12]. Chakravarthy, K., Rajendran, C. 1999. “A heuristic for scheduling in a flow-shop with the bicriteria of makespan and maximum tardiness minimization,” Production Planning and Control, 10, pp.707–714.

[13]. Framinan, J. M., Leisten, R., 2006, “A heuristic for scheduling a permutation flow-shop with makespan objective subject to maximum tardiness,” International Journal of Production Economics, 99, pp. 28–40.

[14]. Rajendran, C., 1992, “Two-stage flow-shop scheduling problem with bicriteria,” Journal of the Operational Research Society, 43, pp. 871–884.

[15]. Johnson, S. M., 1954, “Optimal two- and three-stage production schedules with setup times included,” Naval Research Logistics Quarterly, 1, pp. 61–68.

[16]. Neppalli, V. R., Chen, C. L., Gupta, J. N. D., 1996, “Genetic algorithms for the two-stage bicriteria flow-shop problem,” European Journal of Operational Research, 95, pp. 356–373.

[17]. Schaffer, J. D., 1985, “Multiple objective optimization with vector evaluated genetic algorithms,” Proceedings of the 1st International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, pp. 93–100.

[18]. Gupta, J. N. D., Palanimuthu, N., Chen, C. L., 1999, “Designing a tabu search algorithm for the two-stage flow shop problem with secondary criterion,” Production Planning & Control 10, pp. 251–265.

[19]. Gupta, J. N. D., K. Hennig, F. Werner. 2002. “Local search heuristics for two-stage flow shop problems with secondary criterion.” Computers & Operations Research, 29, pp. 123–149.

[20]. Gupta, J. N. D., Neppalli,V.R., Werner, F., 2001, “Minimizing total flow time in a two-machine flow-shop problem with minimum makespan,” International Journal of Production Economics, 69, pp. 323–338.

[21]. T’kindt, V., Monmarché, N., Tercinet, F., Laügt, D., 2002, “An ant colony optimization algorithm to solve a 2-machine bicriteria flow-shop scheduling problem,” European Journal of Operational Research,142, pp. 250–257.

[22]. T’kindt, V., Gupta, J. N. D., Billaut, J. C., 2003, “Two-machine flow-shop scheduling with a secondary criterion,” Computers & Operations Research, 30, pp. 505–526.

[23]. Venditti, L., Pacciarelli, D., and Meloni, C., 2010, “A tabu search algorithm for scheduling pharmaceutical packaging operations,” European Journal of Operational Research, 202(2), pp. 538-546.

[24]. Nagar, A., Heragu, S. S., Haddock, J., 1996, “A combined branch-and-bound and genetic algorithm based approach for a flow-shop scheduling problem” Annals of Operations Research, 63, pp. 397–414.

[25]. Sridhar, J., Rajendran, C., 1996, “Scheduling in flow-shop and cellular manufacturing systems with multiple objectives: A genetic algorithmic approach,” Production Planning & Control, 7, pp. 374–382.

[26]. Cavalieri, S., Gaiardelli, P. 1998. “Hybrid genetic algorithms for a multiple-objective scheduling problem,” Journal of Intelligent Manufacturing, 9, pp. 361–367.

[27]. Yeh, W.C., 1999, “A new branch-and-bound approach for the n/2/flow-shop/$\alpha F + \beta C_{max}$,”

Computers & Operations Research, 26, pp.1293–1310.

[28]. Yeh,W.C., 2001, "An efficient branch-and-bound algorithm for the two-machine bicriteria flow-shop scheduling problem," Journal of Manufacturing Systems, 20, pp. 113–123.

[29]. Yeh, W.C., 2002, "A memetic algorithm for the n/2/flow-shop/$\alpha F + \beta C_{max}$ scheduling problem", The International Journal of Advanced Manufacturing Technology, 20, pp.464–473.

[30]. Lee, C. E., Chou, F. D., 1998, "A two-machine flow-shop scheduling heuristic with bicriteria objective," International Journal of Industrial Engineering, 5, pp. 128–139.

[31]. Chou, F. D., Lee, C. E., 1999. "Two-machine flow-shop scheduling with bicriteria problem. Computers & Industrial Engineering," 36, pp. 549–564.

[32]. Sivrikaya-Şerifoğlu, F., Ulusoy, G., 1998, "A bicriteria two-machine permutation flow-shop problem," European Journal of Operational Research, 107, pp. 414–430.

[33]. Chang, P. C., Hsieh, J. C., Lin, S. G. 2002. "The development of gradual-priority weighting approach for the multi-objective flow-shop scheduling problem," International Journal of Production Economics, 79, pp. 171–183.

[34]. Murata, T., Ishibuchi, H., Tanaka, H., 1996, "Multi-objective genetic algorithm and its applications to flow-shop scheduling," Computers & Industrial Engineering, 30, pp. 957–968.

[35]. Ishibuchi, H., Murata, T., 1998, "A multi-objective genetic local search algorithm and its application to flow-shop scheduling," IEEE Transactions on Systems Man and Cybernetics, 28, pp. 392–403.

[36]. Framinan, J. M., Leisten, R., Ruiz-Usano, R., 2002, "Efficient heuristics for flow-shop sequencing with the objectives of makespan and flowtime minimization," European Journal of Operational Research, 141, pp. 559–569.

[37]. Allahverdi, A. 2003. "The two- and m-machine flow-shop scheduling problems with bicriteria of makespan and mean flowtime," European Journal of Operational Research, 147, pp. 373–396.

[38]. Allahverdi, A. 2004. "A new heuristic for m-machine flow-shop scheduling problem with bicriteria of makespan and maximum tardiness," Computers & Operations Research, 31, pp. 157–180.

[39]. Ponnambalam, S. G., Jagannathan, H., Kataria, M., Gadicherla, H., 2004, "A TSP-GA multiobjective algorithm for flow-shop scheduling," The International Journal of Advanced Manufacturing Technology, 23, pp. 909–915.

[40]. Lin, B. M. T., Wu, J, M., 2006, " Bicriteria scheduling in a two-machine permutation flow-shop,"International Journal of Production Research, 44, pp. 2299–2312.

[41]. Lemesre, J., Dhaenens,C., Talbi, E. G., 2007, "An exact parallel method for a bi-objective permutation flow-shop problem," European Journal of Operational Research, 177, pp. 1641–1655.

[42]. Madhushini,N., Rajendran, C., and Deepa, Y., 2009, "Branch-and-bound algorithms for scheduling in permutation flow-shops to minimize the sum of weighted flowtime/sum of weighted tardiness/sum of weighted flowtime and weighted tardiness/sum of weighted flowtime, weighted tardiness and weighted earliness of jobs," Journal of the Operational Research Society, 60, pp. 991-1004.

[43]. Saym, S., Karabati, S., 1999, "A bicriteria approach to the two-machine flow shop scheduling problem," European Journal of Operational Research, 113, pp. 435–449.

[44]. Liao, C. J., Yu, W.C., Joe, C, B., 1997, "Bicriterion scheduling in the two-machine flow-shop," Journal of the Operational Research Society, 48, pp. 929–935.

[45]. Lee, W. C., Wu, C. C, 2001, "Minimizing the total flow time and the tardiness in a two-machine flow shop," International Journal of Systems Science, 32, pp. 365–373.

[46]. Bagchi, T. P., 2001. Pareto-optimal solutions for multi-objective production scheduling problems. E. Zitzler, K. Deb, L. Thiele, Carlos A. Coello Coello, D. Corne, eds., Evolutionary Multi-Criterion Optimization, First International Conference, EMO 2001, Zurich, Switzerland, Proceedings, Lecture Notes in Computer Science, vol. 1993. Springer, 458–471.

[47]. Srinivas, N. and Deb, K., 1994,"Multiobjective optimization using nondominated sorting in genetic algorithms", Evolutionary Computation, 2(3), pp. 221-248.

[48]. Murata, T., Ishibuchi,H.,Gen. M., 2001. Specification of genetic search directions in cellular multiobjective genetic algorithms. E. Zitzler, K. Deb, L. Thiele, Carlos A. Coello Coello, D. Corne, eds., Evolutionary Multi-Criterion Optimization, First International Conference, EMO 2001, Zurich, Switzerland, March 7-9, 2001, Proceedings, Lecture Notes in Computer Science, Springer, 82–95.

[49]. Ishibuchi, H., Yoshida, T., Murata, T., 2003, Balance between genetic search and local search in memetic algorithms for multiobjective permutation flow-shop scheduling," IEEE Transactions on Evolutionary Computation, 7, pp. 204–223.

[50]. Loukil, T., Teghem,J., Fortemps, P., 2000, "Solving multi-objective production scheduling

problems with tabu search" Control and Cybernetics, 29, pp. 819–828.

[51]. Loukil, T., Teghem, J., Tuyttens, D., 2005, "Solving multi-objective production scheduling problems using metaheuristics," European Journal of Operational Research, 161, pp. 42–61.

[52]. Toktaş, Berkin, Azizoğlu, M., Köksalan, S. K., 2004. "Two-machine flow shop scheduling with two criteria: Maximum earliness and makespan." European Journal of Operational Research, 157,pp 286–295.

[53]. Suresh, R. K., Mohanasundaram, K. M., 2004. Pareto archived simulated annealing for permutation flow-shop scheduling with multiple objectives. IEEE Conference on Cybernetics and Intelligent Systems (CIS), Singapore, December 1-3, 2004, Proceedings, vol. 2. 712–717.

[54]. Varadharajan, T.K., Rajendran, C., 2005, "A multi-objective simulated-annealing algorithm for scheduling in flow-shops to minimize the makespan and total flowtime of jobs," European Journal of Operational Research, 167,pp. 772–795.

[55]. Arroyo, J. E. C., Armentano, V. A. 2004. "A partial enumeration heuristic for multi-objective flow-shop scheduling problems," Journal of the Operational Research Society, 55, pp. 1000–1007.

[56]. Armentano, V. A., Arroyo, J. E. C. 2004. "An application of a multi-objective tabu search algorithm to a bicriteria flow-shop problem," Journal of Heuristics, 10, pp. 463–481.

[57]. Arroyo, J. E. C., Armentano, V. A. 2005. "Genetic local search for multi-objective flow-shop scheduling problems," European Journal of Operational Research, 167, pp. 717–738.

[58]. Pasupathy, T., Rajendran,C., Suresh, R. K., 2006, "A multi-objective genetic algorithm for scheduling in flow shops to minimize the makespan and total flow time of jobs," The International Journal of Advanced Manufacturing Technology, 27, pp. 804–815.

[59]. Melab, N., Mezmaz,M., Talbi, E.G., 2006, "Parallel cooperative meta-heuristics on the computational grid. a case study: the bi-objective flow-shop problem," Parallel Computing, 32, pp. 643–659.

[60]. Rahimi-Vahed, A. R., Mirghorbani, S. M., 2007, "A multi-objective particle swarm for a flow shop scheduling problem," Journal of Combinatorial Optimization, 13, pp. 79–102.

[61]. Zitzler, E., Laumanns, M., Thiele, L., 2001, "SPEA2: Improving the strength pareto evolutionary algorithm," Tech. Rep. 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland.

[62]. Geiger, M. J., 2007, "On operators and search space topology in multi-objective flow shop scheduling," European Journal of Operational Research, 185(1), pp. 195-206.

[63]. Selen,W. J., Hott, D. D., 1986, "A mixed-integer goal-programming formulation of the standard flow-shop scheduling problem," Journal of the Operational Research Society, 37, pp. 1121–1128.

[64]. Wilson. 1989. "Alternative formulations of a flow-shop scheduling problem." Journal of the Operational Research Society, 40, pp 395–399.

[65]. Ho, Johnny C., Chang, Y., 1991, "A new heuristic for the n-job, M-machine flow-shop problem," European Journal of Operational Research 52 194–202.

[66]. Gangadharan, R., Rajendran, C., 1994, "A simulated annealing heuristic for scheduling in a flow-shop with bicriteria," Computers & Industrial Engineering, 27, pp. 473–476.

[67]. Rajendran, C., 1995, "Heuristics for scheduling in flow-shop with multiple objectives," European Journal of Operational Research, 82, pp. 540–555.

[68]. Rajendran, C., 1994, "A heuristic for scheduling in flow-shop and flowline-based manufacturing cell with multicriteria," International Journal of Production Research, 32, pp. 2541–2558.

[69]. Ravindran, D., Haq, A. N.,Selvakuar, S. J., Sivaraman, R., 2005. "Flow-shop scheduling with Multipleobjective of minimizing makespan and total flowtime. The International Journal of Advanced Manufacturing Technology 25 1007–1012.

[70]. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., da Fonseca, V. G., 2003, "Performance assessment of multiobjective optimizers: an analysis and review," IEEE Transactions on Evolutionary Computation, 7, pp. 117–132.

[71]. Paquete, L. F., 2005. "Stochastic local search algorithms for multiobjective combinatorial optimization: Method and analysis." Ph.D. thesis, Computer Science Department. Darmstadt University of Technology. Darmstadt, Germany.

[72]. Knowles, J., Thiele,L., Zitzler, E., 2006, "A tutorial on the performance assessment of stochastic multiobjective optimizers," Tech. Rep. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland. Revised version.

[73]. Zitzler, E., Thiele, L., 1999, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," IEEE Transactions on Evolutionary Computation, 3, pp. 257–271.

[74]. Deb, K., 2001, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, West Sussex, England.

[75]. Nawaz, M., Emory, E., Enscore, Jr, Ham, I., 1983, "A heuristic algorithm for the *m*-machine, n-job flow-shop sequencing problem. OMEGA," The International Journal of Management Science, 11, pp. 91–95.

[76]. Corne, D.W., Knowles, J. D., Oates, M. J. 2000. The pareto envelope-based selection algorithm for multiobjective optimization. M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, Juan J. Merelo Guervós, H. P. Schwefel, eds., Parallel Problem Solving from Nature – PPSN VI, 6th International Conference, Paris, France, Proceedings, Lecture Notes in Computer Science, 1917. Springer, pp 839–848.

[77].Knowles,J.,D.W.Corne.2000.Approximatingthe nondominatedfrontusingtheparetoarchived evolution strategy. Evolutionary Computation 8 149–172.

[78]. Corne, D.W., Jerram, N. R., Knowles, J. D., Oates, M. J. 2001. PESA-II: Region-based selection in evolutionary multiobjective optimization. L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H. M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, Max H. Garzon, E. Burke, eds., Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001).Morgan Kaufmann, San Francisco, California, USA, pp 283–290.

[79]. Deb, K., 2002, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, 6, pp. 182–197.

[80]. Deb, K.,Mohan, M., Mishra, S., 2002, "A fast multi-objective evolutionary algorithm for finding well spread pareto-optimal solutions," Technical Report 2003002, Kanpur Genetic Algorithms Laboratory (KanGAL), Kampur, India.

[81]. Zitzler, E., Künzli, S., 2004, "Indicator-based selection in multiobjective search. 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)," Springer-Verlag, Berlin, Germany, Birmingham, UK, pp.832–842.

[82]. Kollat, J. B., Reed, P. M., 2005. The value of online adaptive search: A performance comparison of nsgaii, ε−nsgaii and ε−moea. Carlos A. Coello Coello, A. Hernández Aguirre, E. Zitzler,eds., Evolutionary Multi-Criterion Optimization, Third International Conference, EMO 2005, Guanajuato, Mexico, Proceedings, Lecture Notes in Computer Science, vol. 34(10). Springer, 386–398.

[83]. Wang, J. B., Xia, Z. Q., 2005 "Flow-shop scheduling with a learning effect" Journal of the Operational Research Society, 56, pp. 1325-1330.

[84]. Tavakkoli-Moghaddam, R., Rahimi-Vahed, A., Mirzaei, A. H., 2007. "A hybrid multi-objective immune algorithm for a flow-shop scheduling problem with bi-objectives: Weighted mean completion time and weighted mean tardiness" Information Sciences, 177(22), pp. 5072-5090.

[85]. Yagmahan, B., Yenisey, M. M., 2008, **"**Ant colony optimization for multi-objective flow-shop scheduling problem**"** Computers & Industrial Engineering, 54(3), pp. 411-420

[86]. Allouche, M. A., Martel, B. A., Jean-Marc, Loukil, T., and Rebaï, A., 2009. "Solving multi-criteria scheduling flow-shop problem through compromise programming and satisfaction functions" European Journal of Operational Research, 192(2), pp. 460-467.

[87]. Yagmahan, B., Yenisey, M. M., 2010, **"**A multi-objective ant colony system algorithm for flow-shop scheduling problem **"** Expert Systems with Applications, 37(2), pp. 1361-1368.

[88]. Dugardin, F., Yalaoui, F., Amodeo, L., 2010. "New multi-objective method to solve reentrant hybrid flow-shop scheduling problem" European Journal of Operational Research, 203(1), pp. 22-31.

[89]. Mellor, P., 1966, "A Review of Job-shop Scheduling" Journal of the Operational Research Society, 17, pp. 161-171.

[90]. Coello C.A., 1999, "A comprehensive survey of evolutionary-based multioboptimization techniques", Knowledge and Information Systems, 1(3), pp. 269-308.

[91]. Dimopoulos C. and Zalzala, A.M.S., 2000, "Recent developments in evolutionary computation for manufacturing optimization: problems, solutions and comparisons", IEEE Transactions in Evolutionary Computation, 4(2), pp.93-113.

[92]. Udo, G. J., 1994, "A Simulation Study of Due-date Assignment Rules in a Dynamic Job-shop" Journal of the Operational Research Society, 45, pp. 1425-1435.

[93]. Toker, A., Kondakci, S., Erkip, N., 1994. "Job-shop Scheduling under a Non-Renewable Resource Constraint" Journal of the Operational Research Society, 45, pp. 942-947.

[94]. Mesghouni, K., Pesin,P., Trentesaux, D., Hammadi, S., Tahon, C., and Borne, P., 1999, "Hybrid approach to decision making for job-shop scheduling", Production Planning & Control,10(7), pp. 690 706.

[95]. Esquivel S., Ferrero, S., Gallard, R., Salto, C., Alfonso, H., and Schótz, M.,2002, "Enhanced evolutionary algorithms for single and multiobjective optimization in the job scheduling problem", Knowledge-Based Systems, 15, pp. 13-25.

[96]. Thiagarajan, S., Rajendran, C., 2005, "Scheduling in dynamic assembly job-shops to

minimize the sum of weighted earliness, weighted tardiness and weighted flowtime of jobs "Computers & Industrial Engineering, 49(4), pp. 463-503.

[97]. Gao, J., Gen, M., Sun, L., Zhao, X., 2007, "A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job-shop scheduling problems" Computers & Industrial Engineering, 53(1), pp. 149-162.

[98]. Cheng, T.C.E., Kang, L. Y., Ng C. T. 2007. "Due-date assignment and parallel-machine scheduling with deteriorating jobs" Journal of the Operational Research Society, 58, pp. 1103–1108.

[99]. Vilcot, G., and Jean, B., 2008, "A tabu search and a genetic algorithm for solving a bicriteria general job-shop scheduling problem," European Journal of Operational Research, 190(2), pp. 398-411

[100]. Tavakkoli-Moghaddam, R., Safaei, N., Kah, M. M. O., 2008, "Accessing feasible space in a generalized job-shop scheduling problem with the fuzzy processing times: a fuzzy-neural approach" Journal of the Operational Research Society, 59, pp. 431-442.

[101]. Tay, J. C., Ho, N. B., 2008, "Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems" Computers & Industrial Engineering, 54(3), pp. 453-473.

[102]. Lei, D., 2008, "A Pareto archive particle swarm optimization for multi-objective job-shop scheduling" Computers & Industrial Engineering, 54(4), pp. 960-971.

[103]. Manikas, A., Chang, Y., 2009, "Multi-criteria sequence-dependent job-shop scheduling using genetic algorithms" Computers and Industrial Engineering, 56(1), pp. 179-185.

[104]. Xing, L., Chen, Y., Yang, K., 2009, "Multi-objective flexible job-shop schedule: Design and evaluation by simulation modeling" Applied Soft Computing, 9(1), pp. 362-376

[105]. Zhang, G., Shao, X., Li, P., Gao, L., 2009, "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem" Computers & Industrial Engineering, 56(4), pp. 1309-1318.

[106]. Zobolas, G. I., Tarantilis, C. D., Ioan, G., 2009, "A hybrid evolutionary algorithm for the job-shop scheduling problem" Journal of the Operational Research Society, 60, pp. 221-235.

[107]. Zhu, Z.C., Ng, K. M., Ong, H. L., 2010, "A modified tabu search algorithm for cost-based job-shop problem" Journal of the Operational Research Society, 61, pp. 611-619.

[108]. Huang, R., 2010, "Multi-objective job-shop scheduling with lot-splitting production", International Journal of Production Economics, 124(1), pp. 206-213.

[109]. Adibi, M.A., Zandieh, M., Amiri, M., 2010, "Multi-objective scheduling of dynamic job-shop using variable neighborhood search " Expert Systems with Applications, 37(1), pp. 282-287.

[110]. Arumugam, V., Ramani, S., 1980. "Evaluation of Value Time Sequencing Rules in a Real World Job-shop" Journal of the Operational Research Society, 31, pp. 895-904.

[111]. Shaw J. and Fleming, P. J., 1996, "An initial study of practical multi-objective production scheduling using genetic algorithms", in Proceedings of the UKACC International Conference on Control, IEE Conference Publication no. 427, pp.479-484.

[112]. Fonseca C.M. and Fleming, P.J., 1993, "Genetic algorithms for multi objective optimization: formulation, discussion and generalization", in Proceedings of the 5th International Conference on Genetic Algorithms and their Applications, S. Forrest, Ed. San Mateo, CA: Morgan Kaufmann, pp.416-423.

[113]. Tamaki, H., Mukai,T., Kawakami, K., and Araki, M., 1996, "Genetic algorithm approach to multi-objective scheduling problem in plastics forming plant," Advances in Production Management Systems, N. Okino, H. Tamura, S. Fujii, Eds. Chapman & Hall, pp.553-556.

[114]. Khoo, L.P., Lee, S. G., and Yin, X.F., 2000, "A prototype genetic algorithm enhanced multi-objective scheduler for manufacturing systems", International Journal of Advanced Manufacturing Technology, 16, pp.131- 138.