Gezira J. of Engineering and applied Sciences
ISSN Linking: 1858-5698
http://journals.uofg.edu.sd/index.php/gjeas/

Approximation algorithms for solving multi-objective optimization problems: Short communication

Gais Alhadi Babikir [1]
University of Gezira, Wad Madani, Sudan.
Gais.alhadi@uofg.edu.sd, Gais.alhadi@hotmail.com.

INFORMATION:

Submission: 29/11/2021

Accepted: 11/09/2022

Publication: 17/10/2022

ABSTRACT
This paper tries to cover the main aspects/properties related to scheduling problems, approximation algorithms, and multi-objective combinatorial optimization. Then, we try to describe the main techniques that can be used to solve such problems. In this paper, the reviews results relate to multi-objective optimization problems, exact and approximation search, with the aim of getting all Pareto optimal solutions for some NP-hard problems.

KEYWORDS
Multi-objective Optimization, Approximation Algorithm, Pareto-optimal solutions

1. INTRODUCTION

Generally, Multi-criteria Decision Making (MCDM) refers to the decision-making process in multiple/conflicting criteria. Indeed, there are two types of MCDM problems: the first type is Multi-objective Decision Making that contains an infinite number of alternative solutions. Thus, the non-dominated Frontier has an infinite number of non-dominated points. The second type of MCDM problems is Multi-attribute Decision Making that contains a finite number of alternative solutions. Thus, the non-dominated frontier has a finite number of non-dominated points. Noteworthy, MCDM problems may not always have a definitive or unique solution. Therefore, different names are given to different solutions depending on the nature of the solutions. Multi-objective Optimization (also known as, multi-criteria optimization, multi-attribute optimization, or Pareto optimization) was highlighted as an area of MCDM that takes into account optimization problems involving more than one objective function. This problem aims in general at finding a special subset of "good" solutions. These solutions are called "Pareto-optimal" and they are defined as those not dominated by other solutions.

The organization of this paper is as follows: the next Section 2 covers the main aspects related to multiple-objective optimization problem. Section 3 provides a description of dominated and non-dominated solutions. In Section 4, we present the main techniques for solving multi-objective problems. Section 5 presents the approximation concepts. The review of multi-objective optimization has been presented in Section 6, and the last Section 7 concludes the paper.

2. MULTIPLE-OBJECTIVE OPTIMIZATION PROBLEM

Multiple-objective optimization problem is one of the challenging areas faced by researchers in decision sciences since the 1950s, with many papers and books (see for example [1-8]). The importance of multiple-objective optimization derives from the proliferation of multicriteria problems in almost all deciding-making areas such as manufacturing, engineering, transportation, biology, economics, business, healthcare systems, and supplies. For instance, the development of a new component may involve reducing the weight to a minimum, while increasing the strength or choosing a portfolio may include maximizing the expected return while minimizing the opportunity. Noteworthy, there are significant differences between the multi-objective optimization problems that allow the preemption, which are called preemptive (often easier in mathematical terms and have an infinite number of schedules), and those that are non-preemptive (having a finite number of schedules). It should be noted that many multi-objective scheduling problems are NP-hard. Thus, it is necessary to find schedules with better performance. The good news is that there are various modifications of traditional approaches that can be exploited to solve multiple objective scheduling problems. These approaches include: Procedures based on dispatching rules; Branch-and-bound and filtered beam search; Perturbation analysis; Local search technique; Approximation algorithms [9]. Multi-objective optimization involves in the most of times a set of conflicting objectives. We can use one objective function by connecting the two objectives with a common cost advantage. Typically, some optimization problems cannot achieve some way to reach the competitive objectives. Therefore, the relative importance of those objectives cannot be measured numerically [10].

In general, a Multi-objective Optimization Problem (MOP) can be defined by a vector function as follows:

$$MOP = \begin{cases} f(x) = f_1(x), f_2(x), \dots, f_n(x) \\ \qquad s.t, \qquad x \in X \end{cases}$$

We assume that: for each decision vector $x \in X$ assigns an objective vector $z \in Z$, where X is the feasible set in the decision space and Z is the feasible set in the objective space. Here, the possible values of the objective function $f(x) = f_1(x), f_2(x), \dots, f_n(x)$ constitute the objective space Z, where $n \geq 2$ (see Fig.1).

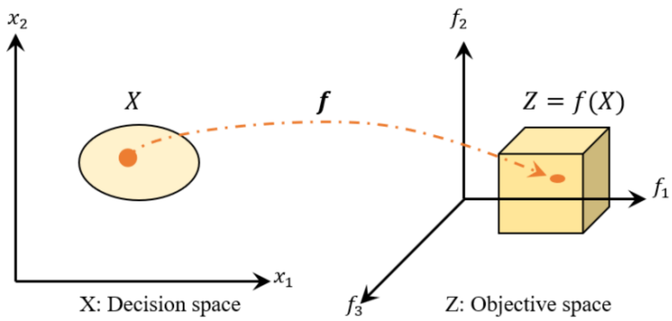

Fig.1 Decision space and objective space

3. DOMINATED AND NON-DOMINATED SOLUTIONS

Suppose we aim to minimize two objectives (f_1, f_2), we can say that the objective vector $z \in Z$ dominates an objective vector $z^* \in Z$, if both of the following conditions are true: $\forall_i \in \{1,2,\dots,n\}, z_i \leq Z_i^*$; $\exists_j \in \{1,2,\dots,n\}, z_j ,< Z_j^*$. *For example, as we can see in Fig. 2, solution A dominates solution B, because A is better than B for all objectives* (f_1, f_2).

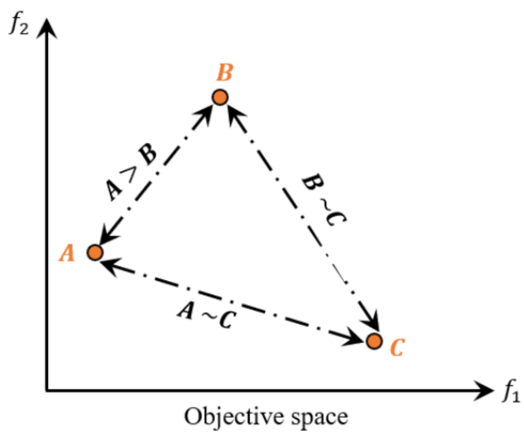

Fig. 2 Solution *A* dominates solution *B*

It should be noted that the set of all non-dominated solutions is called a Pareto-optimal set (or Pareto-optimal solutions) and its mapping in the objective space is called Pareto front (See Fig. 3). Clearly, Pareto solutions are solutions that are not dominated by other solutions [11].

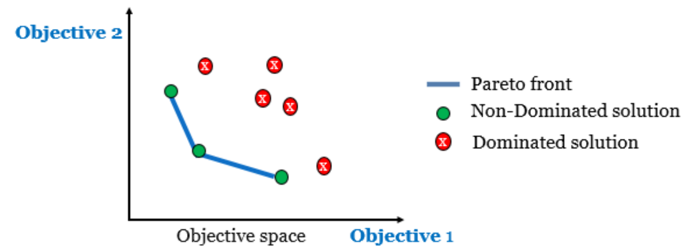

Fig.3 Illustration of Pareto front and the dominated and non-dominated solutions in objective space

It is worth mentioning that multi-objective optimization (MOP) as a part of the decision making process can be classified into three classes. These classes have an interaction between the algorithm and the decision maker. Clearly, the decision maker preferences may be expressed as follows:

1. ***A priori****:* before the results of the optimization process are known *(i.e. before the resolution process).*
2. ***A posteriori****:* which has ***no*** preferences before (i.e. after the resolution process).
3. ***Interactive****:* in this case, the decision maker interaction will be during the resolution process.

In fact, optimization algorithms for solving multi-objective optimization algorithms can be classified into exact and approximate algorithms. Noteworthy, during the last decades, the bi-criteria optimization problems have attracted many researchers from all over the world and have been widely studied in the literature using exact methods such as Branch and Bound and Dynamic Programming algorithms. Indeed, exact search methods are effective for small-size problems. For problems with more than two criteria, there are no accurate and effective measures due to simultaneous difficulties of the NP-hard complexity and the framework of multi criteria optimization problems. However, there are some new techniques proposed in the literature for multi-objective problems. Heuristic methods can be used to solve large-scale or multi-criteria problems to find the Pareto approximation set. Hence, approximate methods can be divided into two classes: The algorithms for a particular problem using some knowledge about the problem; and the metaheuristics (general purpose algorithms) that can be used for a large variety of multi-objective problems. It is noteworthy that the application of metaheuristics to solve multi-objective problems have attracted numerous researchers from all the world and has become an active development area to obtain an approximate set of Pareto optimal solutions. In fact, the approximation theory can provide many tools to carry out such an analysis [8-9].

4. APPROXIMATION CONCEPTS

For self-consistency, we recall some necessary definitions related to the approximation area (See [25]).

Let suppose P an optimization problem, A an approximation algorithm, I an instance of problem P, A(I) is the value for the instance I generated by algorithm A and $A^*(I)$ the optimal value for instance I.

Definition A *is an approximation algorithm for problem P if for any given instance I of that problem, algorithm* A *returns an approximate solution, which is a feasible solution.*

In the rest of this section, we will focus on the used techniques and on the approximation schemes. Two types of schemes can be distinguished in absolute approximation: Polynomial Time Approximation Schemes (PTAS) and Fully Polynomial Time Approximation Schemes (FPTAS). Noteworthy, an approximation scheme is a suboptimal approach based on an input parameter: the accepted error bound. It works quickly, with the aim of getting a solution for an NP-hard problem by respecting the accepted error bound. It is important to mention that these techniques can use linear programming tools or sophisticated algorithmic techniques which lead to difficult implementation problems. Furthermore, some approximation algorithms have impractical running times even though they are polynomial time [26].

Definition Let ε > 0. An algorithm A is called a PTAS for an NP-hard problem, if for any instance I of that problem the algorithm A yields, within a polynomial time for the fixed ε, a feasible solution with an objective value A(I) such that [27]:

$$|A(I) - OPT(I)| \leq \varepsilon . OPT(I),$$

where, OPT(I) is the optimal value of instance I. For the minimization problems, this leads to inequality $A(I) \leq (\mathbf{1} + \varepsilon)OPT(I))$. For the maximization problems, it leads to inequality $A(I) \geq (1 - \varepsilon)OPT(I))$.

When a PTAS runs polynomially in $1/\varepsilon$ and the instance size, it becomes an FPTAS. Such a scheme is the most powerful approximation algorithm that we can expect for an NP-hard problem. Hence, it is clear that approximation schemes are clearly stronger than constant approximation algorithms.

5. A REVIEW OF MULTI-OBJECTIVE OPTIMIZATION

Given the aim of this work, we recall some representative and related existing works to the investigated subject: multiobjective scheduling, makespan and maximum lateness minimization, exact and approximation search. For example, Alhadi et al. [12] considered the scheduling problem on two-parallel machines, with the aim of minimizing the maximum lateness and the makespan. They showed that the problem is NP-hard only in the ordinary sense. Therefore, the authors proposed a dynamic programming algorithm, a PTAS, and two strongly polynomial FPTAS. Furthermore, they did some numerical experiments in order to compare their proposed approaches. The authors showed that the proposed algorithms for the considered problem are very efficient, especially for big instances composed of a lot of jobs. Huo and Zhao [13] considered two bi-criteria scheduling problems on m parallel machines with unavailable intervals, in two cases: In the first case, which is denoted by $P_{m,r_j}| r - a, prmt | \sum C_j / C_{max} \leq T$, each machine has only one unavailable period which starts at time zero. In the second case, which is denoted by $P_{m-1,1} | r - a, prmt | \sum C_j / C_{max} \leq T$, each machine may have multiple unavailable periods, but there is at most one machine unavailable at any time. The authors considered minimizing the makespan and total completion time. They showed that both problems are in P by developing optimal algorithms. A serial-batching machine for minimizing the maximum cost (f_{max}) and makespan (C_{max}) have been considered in [14]. The authors present a polynomial-time algorithm with a time complexity of ($O(n^5)$) to produce all Pareto-optimal solutions. Lin et al. [15] considered the scheduling problem on m unrelated parallel machines, with the aim of minimizing the makespan, total weighted completion time (TWC), and total weighted tardiness (TWT). They proposed two heuristics and a Genetic Algorithm to find non-dominated solutions. The first heuristic is to minimize the makespan and TWT ($R||C_{max}, \sum W_j T_j$), the second heuristic is to minimize the TWC and TWT ($R|| \sum W_j C_j, \sum W_j T_j$), the proposed GA is to minimize the makespan, TWC and TWT ($R||C_{max}, \sum W_j C_j, \sum W_j T_j$). Li et al. [16] studied the scheduling problem on m identical parallel machines with release dates, due dates, and sequence-dependent setup times, to minimize the makespan (C_{max}) and the total tardiness ($\sum T_j$). The authors showed that the problem is NP-hard in the strong sense. Thus, they developed a new mathematical model and approximated methods. A serial-batching machine, to minimize the maximum lateness and the makespan has been considered in [17] ($1|s - batch, b < n|(L_{max}, C_{max})$). The authors present a polynomial-time algorithm with a time complexity of ($O(n^6)$) to produce all Pareto-optimal solutions. In the special case when the processing times and deadlines are agreeable, they present an $O(n^3)$-time algorithm to produce all Pareto optimal solutions. Geng and Yuan [18] studied an unbounded parallel-batching machine to minimize the makespan and the maximum lateness ($1|\beta|C_{max}, L_{max}$). For the strongly NP-hard problem considered, the authors proposed DP with a time complexity of $O(n^{F+1})$. Moreover, they showed that the problem can be solved in $O(n^{2F+1})$ -time algorithm which is of polynomial-time when the number F of families is a constant. A parallel-batching machine to minimize the maximum lateness and the makespan has been considered in [19]. To solve the problem $1|p - batch, b \geq n|F(L_{max}, C_{max})$ (F: an unknown composition objective function), the authors present an algorithm with a time complexity of $O(n^3)$. In [19], the authors studied an unrelated machine with costs, to find a schedule, obtaining a tradeoff between the makespan and the total cost when the number of machines is constant. They present an FPTAS with a time complexity of $O(n(n/\varepsilon)^m)$.

From the presented state-of-the-art, we can conclude that the multi-objective problems have attracted numerous researchers from around the world to achieve various objectives. For more state-of-the-art, the reader is invited to consult the papers by Paschos (2018) [21], Kumar (2008) [22], Amor et al. (2017) [23], Amor and Martel (2014) [24] and Sarkar and Modak (2005) [4].

6. CONCLUSIONS

This paper attempts to review approximation algorithms to solve multi-objective optimization problems. In this paper, we covered the main aspects related to multi-objective combinatorial optimization and the main techniques that can be used to solve such problems. Finally, we have seen some of the most important studies related to multi-objective combinatorial optimization.

References

[1]. A. Allahverdi and T. Aldowaisan, (2004). No-wait flowshops with bicriteria of makespan and maximum lateness. European Journal of Operational Research, 152(1):132–147.

[2]. C. He, Y. Lin and J. Yuan, (2009). A DP algorithm for minimizing makespan and total completion time on a series-batching machine. Information processing letters, 109(12), 603-607.

[3]. G. Sapienza, G. Brestovac, R. Grgurina and T. Seceleanu, (2016). On applying multiple criteria decision analysis in embedded systems design. Design automation for embedded systems, 20(3), 211-238.

[4]. D. Sarkar and J. Modak, (2005). Pareto-optimal solutions for multi-objective optimization of fed-batch bioreactors using nondominated sorting genetic algorithm. Chemical Engineering Science, 60(2):481–492.

[5]. C. Silva and E. Biscaia, (2003). Genetic algorithm development for multi-objective optimization of batch free-radical polymerization reactors. Computers & chemical engineering, 27(8):1329–1344.

[6]. V. T'kindt and J. Billaut, (2006). Multicriteria scheduling: theory, models and algorithms. Springer Science & Business Media.

[7]. B. Lin and A. Jeng, (2004). Parallel-machine batch scheduling to minimize the maximum lateness and the number of tardy jobs. International Journal of Production Economics, 91(2):121–134.

[8]. P. Brucker, (2007). Scheduling Algorithms. Springer, Berlin.

[9]. M. Pinedo, (2016). Scheduling Theory, Algorithms, and Systems. Springer, New York.

[10]. A. V. Lotov, V. A. Bushenkov and G. K. Kamenev, (2013). Interactive decision maps: Approximation and visualization of Pareto frontier (Vol. 89). Springer Science & Business Media.

[11]. E. Talbi, (2009). Metaheuristics: From Design to Implementation. 2009, volume 74. John Wiley & Sons.

[12]. G. Alhadi, I. Kacem, P. Laroche and I. M. Osman, (2020). Approximation algorithms for minimizing the maximum lateness and makespan on parallel machines. Annals of Operations Research, 285(1), 369-395.

[13]. Y. Huo and H. Zhao, (2013). Bi-criteria Scheduling on Multiple Machines Subject to Machine Availability Constraints. In Frontiers in Algorithmics and Algorithmic Aspects in Information and Management (pp. 325-338). Springer, Berlin, Heidelberg.

[14]. C. He, H. Lin, Y. Lin and J. Tian, (2013). Bicriteria scheduling on a series-batching machine to minimize maximum cost and makespan. Central European Journal of Operations Research, 21(1), 177-186.

[15]. Y. K. Lin, J. W. Fowler and M. E. Pfund, (2013). Multiple-objective heuristics for scheduling unrelated parallel machines. European Journal of Operational Research, 227(2), 239-253.

[16]. X. Li, L. Amodeo, F. Yalaoui and H. Chehade, (2010). A multiobjective optimization approach to solve a parallel machines scheduling problem. Advances in Artificial Intelligence, 2010.

[17]. C. He, H. Lin and Y. Lin, (2015). Bounded serial-batching scheduling for minimizing maximum lateness and makespan. Discrete Optimization, 16, 70-75.

[18]. Z. Geng and J. Yuan, (2015). Pareto optimization scheduling of family jobs on a p-batch machine to minimize makespan and maximum lateness. Theoretical Computer Science, 570, 22-29.

[19]. C. He, Y. Lin and J. Yuan, (2007). Bicriteria scheduling on a batching machine to minimize maximum lateness and makespan. Theoretical Computer Science, 381(1-3), 234-240.

[20]. E. Angel, E. Bampis and A. Kononov, (2001). A FPTAS for approximating the unrelated parallel machines scheduling problem with costs. In European Symposium on Algorithms (pp. 194-205). Springer, Berlin, Heidelberg.

[21]. V. Th. Paschos, (2018). Combinatorial approximation of maximum k-vertex cover in bipartite graphs within ratio 0.7. RAIRO - Operations Research, 52:305–314.

[22]. P. Kumar, (2008). A Framework for Multi-objective Optimization and Multi-criteria Decision Making for Design of Electrical Drives. Universiteit Delft.

[23]. S. B Amor, K. Zaras, and E. A. Aguayo, (2017). The value of additional information in multicriteria decision making choice problems with information imperfections. Annals of Operations Research, 253(1):61–76.

[24]. S. B Amor and J. M. Martel, (2014). A new distance measure including the weak preference relation: Application to the multiple criteria aggregation procedure for mixed evaluations. European Journal of Operational Research, 237(3):1165–1169.

[25]. G. Alhadi, I. Kacem, P. Laroche, I. M. Osman, (2018). Maximum Lateness Minimization on Two-Parallel Machine with a Non-availability Interval. In 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 757-762). IEEE.

[26]. *D.* Williamson and D. Shmoys, (2010). The Design of Approximation Algorithms. Cambridge university press, New York.

[27]. P. Schuurman and G. Woeginger, (2011). Lectures on Scheduling. University of Technology,NL-5600 MB Eindhoven, The Netherlands.