Applied Soft Computing 58 (2017) 700–713

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A hybrid project scheduling and material ordering problem: Modeling and solution algorithms

N. Zoraghi[a,*], A. Shahsavar[a], S.T.A. Niaki[b]

[a] Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Islamic Azad University, Qazvin Branch, Qazvin, Iran
[b] Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 8 July 2016
Received in revised form 11 March 2017
Accepted 15 May 2017
Available online 24 May 2017

Keywords:
Material ordering
Multi-mode project scheduling
Multi-objective evolutionary algorithms
Pareto frontier
Total quantity discount

ABSTRACT

A novel combination of a multimode project scheduling problem with material ordering, in which material procurements are exposed to the total quantity discount policy is investigated in this paper. The study aims at finding an optimal Pareto frontier for a triple objective model derived for the problem. While the first objective minimizes the makespan of the project, the second objective maximizes the robustness of the project schedule and finally the third objective minimizes the total costs pertaining to renewable and nonrenewable resources involved in a project. Four well-known multi-objective evolutionary algorithms including non-dominated sorting genetic algorithm II (NSGAII), strength Pareto evolutionary algorithm II (SPEAII), multi objective particle swarm optimization (MOPSO), and multi objective evolutionary algorithm based on decomposition (MOEAD) solve the developed triple-objective problem. The parameters of algorithms are tuned by the response surface methodology. The algorithms are carried out on a set of benchmarks and are compared based on five performance metrics evaluating their efficiencies in terms of closeness to the optimal frontier, diversity, and variance of results. Finally, a statistical assessment is conducted to analyze the results obtained by the algorithms. Results show that the NSGAII considerably outperforms others in 4 out of 5 metrics and the MOPSO performs better in terms of the remaining metric.

1. Introduction

In this study, a combination of a project scheduling problem (PSP) with the material ordering problem (MOP) is investigated where a total quantity discount on material prices is offered based on the size of orders. The activities of the project can be performed in different modes with no preemption. In addition, two types of resources, renewable and nonrenewable resources (material), are required to carry out each activity. Although the renewable resources are assumed to be scarce in the project, their level of scarcity is determined throughout the model optimization in a hope to find economical renewable resource scarcity levels that guarantee the minimum related costs. Such definitions lead to the multimode resource investment problem (MRIP), a class of the PSPs in which a decision variable is defined for the scarcity level of each renewable resource aiming at finding a schedule of activities along with the scarcity level of renewable resources to minimize the costs. In terms of the materials, there are some pecuniary considerations. They are not available at the starting point of the project and should be supplied within the project longevity. Each procurement order of the materials imposes an ordering cost and each remaining unit of materials per period imposes an inventory holding cost on the project. The price of the materials is optional based on a total discount policy in which the bigger orders are taken with higher discounts on the price of materials. Two questions that arise here are how many and when materials should be ordered to result in minimum material costs. This brings up the quantity discount in material ordering (QDPMO) problem into consideration as it is combined with the MRIP leading to an integrated model, MRIPQDPMO. The novelty and practicality of the combined model is the first motivation of this study.

The PSP is inherently a multi-objective problem based on the decision makers' (DM) requirements. For example, both shortest makespan and minimum costs are desired. While the minimum number of tardy activities is also favorable, if tardiness is permitted, the minimum costs and shortest makespan are simultaneously appealing. Besides, having a robust schedule can decrease the impact of uncertainty in the project to gain other goals, and so on. Similar inherent characteristics of the problem under investigation motivated this work to develop a model to be applicable

* Corresponding author.
E-mail addresses: nima.zoraghi@gmail.com (N. Zoraghi), m.shahsavar@ymail.com (A. Shahsavar), Niaki@sharif.edu, Hniaki@gmail.com (S.T.A. Niaki).

http://dx.doi.org/10.1016/j.asoc.2017.05.030
1568-4946/© 2017 Elsevier B.V. All rights reserved.

in a multi-objective environment, for which three different objectives are considered. The most well-known objective considered is the shortest makespan aiming to have a project finished as early as possible while respecting all the constraints. Although this objective plans the project to be finished as early as possible, the final schedule is vulnerable as a myriad source of uncertainties always exist in reality that exposes the project to likely delays, deviations, and disorders. Some of the common uncertainties in a project are unpredictable changes in the duration of activities, delays in the arrival of materials, machine breakdowns, manpower problems, reworks, and the like. Therefore, maximizing the robustness of the project schedule is considered as the second objective to protect the project against possible disruptions during its longevity. The robustness means that in the case of minute increases in the processing time of some activities due to some uncontrollable factors, the makespan of the project does not change at any cost. If the free slack of an activity is defined as the amount of time it can prolong without delaying the start of its immediate successors, then the robustness is usually acquired by the sum of activities' free slacks. In addition to the makespan and robustness, the other objective that really is an appeal in projects is the minimum costs. That is why the third objective considered is the minimization of total costs of the project that consists of employment costs of the renewable resources plus the ordering, holding, and procurement costs of the materials. Modeling this triple-objective problem is another novelty of this study as there exist no similar consideration in the literature.

2. Literature review

Since its introduction in the middle of the twentieth century, the PSP has been investigated several times in terms of activities, resources, and objectives involved. The PSP with activities under single-mode or multimode executions, the PSP under renewable or nonrenewable resources, the PSP under different objectives such as the shortest makespan or minimum resource cost are just a few. Interested readers are referred to [1–10] for further information.

Introduced in Möhring [11], the RIP, as a general branch of PSPs, has been scrutinized in many efforts. Of the exact methods, one can refer to the branch and bound algorithm of Demeulemeester [12] for the single mode RIP, the time-window based branch and bound of Zimmermann & Engelhardt [13], the depth-first branch and bound of Nübel [14] for the RIP under generalized precedence relations, and the algorithm of Sabzehparvar [15] for the multimode RIP inspired by a packing system. Surprisingly, this algorithm has no need for a feasible solution to start up with. In addition, different meta-heuristic algorithms have been proposed in the literature to tackle the RIP or its extensions for which references [16–19] are just a few.

The PSP has also been combined with other issues such as supply chain management or material requirement planning to model closer-to-reality problems. Aquilano & Smith-Daniels [20] and Smith-Daniels & Aquilano [21] brought the combination of PSP and MOP into considerations. Smith-Daniels & Smith-Daniels [22] developed a mixed integer programming for the combined PSO and MOP. Later, Erabsi & Sepil [23] proposed a heuristic algorithm for the problem where the tardiness is permitted. Dodin & Elimam [24] discussed the presence of variable activity duration, the reward and penalty for completion of the activities, and the quantity discount policy. Further works in [25–29] presented some meta-heuristic algorithms to find near-optimum solutions of the problem.

Multi-objective optimization problems are treated in three different approaches based on the DM's viewpoints [30]. In the first approach, all the objectives are transformed into a single-objective function based on the information given by the DM regarding the importance of the objectives before the search. The single-objective problem is then solved by a classical method producing a single solution. In the second approach, multiple objectives are treated concurrently and in a pristine form, for which a multi-objective meta-heuristic algorithm provides a set of non-dominated solutions (usually known as Pareto front), to the DM who finally selects the most favorable solution. In the third approach, the optimization process is divided into several consecutive steps, where in each step; a set of solutions is presented to the DM to take his/her feedback guiding the next step.

In case of transforming the objectives into a single objective, Slowinski [31] was the first work considering the PSP under different objectives including project duration, maximum lateness, and several cost criteria. The author applied a parametric linear programming to solve it. Nudtasomboon & Randhawa [32] introduced a preemptive goal programming for a triple-objective PSP that includes time, cost, and resource leveling criteria. Abbasi et al. [33] investigated the makespan and the robustness as two objectives for a class of PSPs that were transformed into a linear function. They proposed a simulated annealing algorithm to solve the single objective PSP. In terms of multi-objective PSPs, several efforts have been conducted in the literature. For instance, a Pareto-based simulated annealing (PSA) and a multi-objective tabu search (MOTS) were implemented in Viana & De Sousa [34] to solve the multi-objective resource constrained project scheduling problem aiming to minimize the makespan, the weighted lateness of activities, and the violation of resource availabilities. Al-Fawzan & Haouari [35] proposed an MOTS to minimize the makespan and to maximize the robustness of the project simultaneously. Kazemi & Tavakkoli-Moghaddam [36] developed a non-dominated sorting genetic algorithm II (NSGAII) to solve a PSP with the aims of minimizing the makespan and maximizing the net present value (NPV) of project cash flows. In addition, A comparison on the performance of a strength Pareto evolutionary algorithm II (SPEAII) with the ones of NSGAII and PSA applied to solve the PSP under several objectives was conducted in Ballestín & Blanco [30]. Aboutalebi et al. [37] utilized an NSGAII and a multi-objective particle swarm optimization (MOPSO) for the PSP with the aims of minimizing the makespan and maximizing NPV as the two objectives. Yannibelli & Amandi [38] developed a hybrid multi-objective simulated annealing and a multi-objective evolutionary algorithm to minimize the makespan and to assess the most effective set of human resources for each project activity. Khalili et al. [39] devised a two-stage multi-population genetic algorithm (MPGA) and a two-phase subpopulation genetic algorithm (TPSPGA) to minimize the makespan and to maximize the NPV simultaneously. Gomes et al. [40] implemented five multi-objective meta-heuristics to solve the PSP under the makespan and the total weighted starting times of activities. Their algorithms are based on multi-objective GRASP (MOG), multi-objective variable neighborhood search (MOVNS), and Pareto iterated local search (PILS) methods. Shahsavar et al. [41] developed a triple-objective PSP where minimization of makespan, total cost of the renewable resources of the project, and the variation of resource usage were taken into account. They proposed three self-adaptive genetic algorithm based methods including a MPGA, a TPSPGA, and a non-dominated ranked genetic algorithm (NRGA).

For the case of dividing the optimization process into several consecutive steps, one can refer to the works of Hapke et al. [42] and Slowinski et al. [43].

In this paper, the second approach is taken to tackle the multi-objective MRIPQDPMO under investigation, i.e., the search is performed before the DM's decision and provides him/her with a set of non-dominated solutions. Moreover, as NSGAII and SPEAII have previously shown excellent results, they are employed in the next sections to find Pareto solutions of the problem under study.

Besides, the MOPSO as a potent but less-explored algorithm in tackling the multi-objective PSPs is also utilized in order to explore its benefits when applied to solve the multi-objective MRIPQDPMO. In addition to these three population based algorithms, a multi objective evolutionary algorithm based on decomposition (MOEAD) introduced by Zhang and Li [44] which, to the best of our knowledge, has never been utilized on PSPs is also studied to firstly evaluate its efficiency when applying on multimode PSPs and secondly to validate the results obtained using NSGAII, SPEAII, and MOPSO.

The rest of the paper is structured as follows. In Section 3, the problem is defined and the mathematical formulation is developed. The solution methodologies are demonstrated in Section 4. Section 5 is devoted to performance evaluation and parameter calibration of the solution methodologies. Comparison studies come in Section 6. Finally, Section 7 concludes the paper.

3. Problem definition and formulation

Consider a project in an activity-on-node (AON) format sketched with a graph $G(E, A)$, where E shows the events of the graph standing for the activities of the project and A shows the arcs of the graph standing for the precedence relations between activities of the project. The project has N activities where activity i can be executed in M_i modes, each offering a different duration and resource usage. The dummy activities of 1 and N with zero duration and no resource usage present the start and the finish point of the project, respectively, where $M_1 = M_N = 1$. The activities use a different level of renewable resources or materials per period per mode. In addition, there is a set of precedence finish to start relations with zero time lags between activities. The project has to be finished before a due date (DD). Each unit of renewable resources imposes an employment cost on the project. Regarding the materials, an ordering cost is enforced to the project whenever a batch of each material is procured. Remaining materials at the end of each period of the project also charge the project with a holding cost per unit. In addition, the price of materials is determined based on a total discount policy in which the bigger the size of a batch of material procured, the cheaper the price per unit of that material.

The problem is mathematically formulated based on the following indices, parameters, and decision variables.

Set of indices$i$Index of activities $i \in (1, 2, \ldots, N)$mIndex of mode executions where for each activity i, $m \in (1, 2, \ldots, M_i)$$t$Index of time periods $t \in (0, 1, 2, \ldots, DD)$kIndex of renewable resources $k \in (1, 2, \ldots, K)$$l$Index of materials $l \in (1, 2, \ldots, L)$$y$Index of quantity discount ranges where for each material l, $y \in (1, 2, \ldots, Y_l)$

Parametersd_{im}Duration of activity i executed in mode $m$$r_{imk}$Amount required of renewable resource k to process activity i in mode m per periodu_{iml}Amount required of material l to process activity i in mode m per periodDDDue date for project completionP_iSet of immediate predecessors of activity $i$$S_i$Set of immediate successors of activity $i$$C_k$Employment cost of renewable resource k per unitα_{ly}Limit on the quantity discount range y for material $l$$\delta_{ly}$Price of material l purchased in quantity range $y$$A_l$Ordering cost of material $l$$H_l$Inventory holding cost of material l

Decision variablesX_{imt}A zero-one variable that takes 1, if activity i executed in mode m starts at period t; 0 otherwiseλ_{lyt}A zero-one variable that takes 1, if material l is ordered within quantity discount range y at period t; 0 otherwiseQ_{lyt}Quantity ordered of material l within range y at period $t$$I_{lt}$Inventory level of material l at the end of period $t$$R_k$Employment level of renewable resource $k$$EST_{im}$Earliest start time of activity i if it is executed in mode $m$$EFT_{im}$Earliest finish time of activity i if it is executed in mode $m$$FS_{im}$Free slack of activity i if it is executed in mode m

The mixed-integer non-linear mathematical formulation of the problem consists of three objectives. The objective function shown in Eq. (1) minimizes the makespan of the project. In this objective the variable X_{Nmt} shows the period at which the end activity of the project is started.

$$Min \sum_{m=1}^{M_N}\sum_{t=1}^{DD} t \times X_{Nmt} \tag{1}$$

The objective function in Eq. (2) maximizes the robustness of the project defined as the sum of free slacks of activities. In this function, FS_{im} that shows the free slack of activity i in mode m is calculated as the sum of time periods at which activity i can be started without affecting the start of its immediate successors.

$$\text{Max} \sum_{i=1}^{N}\sum_{m=1}^{M_i} FS_{im} \tag{2}$$

The objective function in Eq. (3) minimizes the total costs of the project including; a) the material ordering costs as the first part, b) the material holding costs as the second part, c) the material procurement costs as the third part, and d) the employment cost of renewable resources as the last part.

$$Min \sum_{l=1}^{L}\sum_{t=0}^{DD} A_l \times \sum_{y=1}^{Y_l} \lambda_{lyt} + \sum_{l=1}^{L}\sum_{t=1}^{DD-1} H_l \times I_{lt} + \sum_{l=1}^{L}\sum_{t=0}^{DD}\sum_{y=1}^{Y_l} \delta_{ly} \times Q_{lyt} + \sum_{k=1}^{K} C_k \times R_k; \tag{3}$$

The above three objective functions are subject to the following constraints.

$$\sum_{m=1}^{M_i}\sum_{t=1}^{DD} X_{imt} = 1;\ \forall \text{i} \tag{4}$$

$$\sum_{m=1}^{M_j}\sum_{t=1}^{DD} t \times X_{jmt} - \sum_{m=1}^{M_i}\sum_{t=1}^{DD} t \times X_{imt} \ge d_{im}; \forall (i, j) \in \text{G} \tag{5}$$

$$\sum_{i=1}^{N}\sum_{m=1}^{M_i} r_{imk} \sum_{w=t-d_{im}+1}^{t} X_{imw} \le R_k;\ \forall \text{k}, \forall \text{t} \tag{6}$$

$$\sum_{m=1}^{M_N}\sum_{t=1}^{DD} t \times X_{Nt} \le DD\,; \tag{7}$$

$$I_{lt} = I_{l(t-1)} + \sum_{y=1}^{Y_l} Q_{lyt} - \sum_{i=1}^{N}\sum_{m=1}^{M_i} u_{iml} \sum_{w=t-d_{im}+1}^{t} X_{imw};\ \forall \text{l}, \forall \text{t} \tag{8}$$

$$I_{l0} = 0; \forall \text{l} \tag{9}$$

$$\alpha_{l(y-1)} \times \lambda_{lyt} \le Q_{lyt} \le \alpha_{ly} \times \lambda_{lyt}; \forall \text{l}, \forall \text{y}, \forall \text{t} \tag{10}$$

$$\alpha_{l0} = 0;\ \forall \text{l} \tag{11}$$

$$\sum_{y=1}^{Y_l} \lambda_{lyt} \le 1; \forall \text{l}, \forall \text{t} \tag{12}$$

$$EST_{1m} = EFT_{1m} = 0; \tag{13}$$

$$EST_{im} = \max\left\{EFT_{jm} | j \in P_i\right\} \times \sum_{t=1}^{DD} X_{imt}; \forall \text{i} \in (2, \ldots, \text{N}), \forall \text{m} \tag{14}$$

Table 1
Characteristics of a small example.

Activity	Successor	Mode 1			Mode 2			Mode 3			Parameters	Value
											$(0, \alpha_1]$	(0,10]
		d	K	L	d	K	L	d	K	L	$(\alpha_1, \alpha_2]$	(10,20]
1	{2,3}	0	0	0	0	0	0	0	0	0	$(\alpha_2, \alpha_3]$	(20,50]
2	4	2	5	10	3	6	9	5	3	9	δ_1	7
3	5	3	8	6	5	5	4	4	6	5	δ_2	6
4	{5,6}	1	2	3	4	2	2	3	3	3	δ_3	5
5	7	4	4	2	3	5	3	5	5	4	H	80
6	7	2	10	9	3	9	8	1	9	8	I	10
7	–	0	0	0	0	0	0	0	0	0	C	5

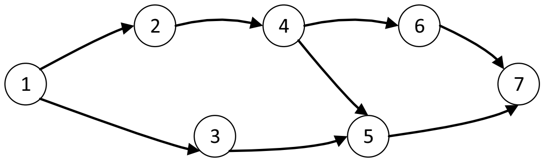


Fig. 1. A network example for the model.

$$EFT_{im} = (EST_{im} + d_{im}) \times \sum_{t=1}^{DD} X_{imt}; \forall \mathrm{i} \in (2, \ldots, \mathrm{N}), \forall \mathrm{m} \tag{15}$$

$$FS_{im} = \left(\min\left\{EST_{jm} | j \in S_i\right\} - EFT_{im}\right) \times \sum_{t=1}^{DD} X_{imt}; \forall \mathrm{i}, \forall \mathrm{m} \tag{16}$$

$$X_{imt} \in \left\{0, 1\right\};\ \lambda_{lyt} \in \left\{0, 1\right\};\ Q_{lyt} \geq 0;\ I_{lt} \geq 0;$$
$$R_k \geq 0;\ \forall i, \forall m, \forall t, \forall l, \forall y, \forall k \tag{17}$$

Eq. (4) enforces each activity to be started only once within the project at a specific mode. Eq. (5) enforces activities to comply with precedence relations. In this formulation, any $(i, j) \in G$ means that activity i is a predecessor of activity j in the graph G. Eq. (6) imposes renewable resource constraints on the model where no violation over the maximum employment (scarcity level) of resources is allowed. Eq. (7) imposes a due date on the completion time of the project. Eq. (8) calculates the inventory level of materials at the end of each period of the project. Eq. (9) asserts that, no inventory level is available in the advent of the project. Eq. (10) restricts batches of materials to be selected among a defined discount range. Eq. (11) sets zero to the lower limits of the first discount range of materials. Eq. (12) ensures that at most one order of a material type is possible per discount range per period. Eq. (13) ts the earliest start and earliest finish time of dummy activity 1 to zero. Eqs. (14) and (15) calculate the earliest start and earliest finish time of other activities, respectively. Eq. (16) calculates the free slack of activities. Noticeably, if activity i is not executed in mode m, its corresponding EST_{im}, EFT_{im}, and $FS_{im}$is set to zero according to Eqs. (14)–(16). Eq. (17) reveal the domains of the variables.

A small example depicted in Fig. 1 gives a better comprehension regarding this model. The example consists of 7 activities which can be performed in three modes. There are only one type of renewable resource and one type of material where the material can be purchased in three different discount ranges. The due date of the network is set to 16. Table 1 summarizes all the information for the simple network under investigation. The symbols in this table are similar to those of model.

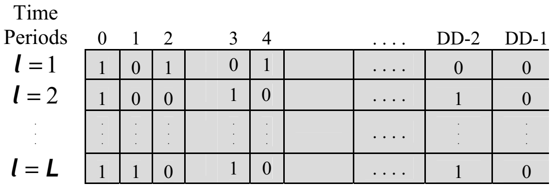
The mathematical formulation of this example is coded in LINGO software, where each objective is solved separately. Table 2 provides the final schedule and objective function values derived from LINGO. The results clearly denote the conflicting nature of the objectives. When the first objective is optimized, the shortest makespan of the project, which is equal to 6, and the corresponding schedule are obtained. By using the obtained final schedule, the second and the third objectives are achieved as follows. The second objective function, which is equal to 2, is calculated based on the final mode of activities as well as the corresponding earliest start and earliest finish time of activities. In addition, the third objective, which is equal to 813, is obtained by solving the mathematical formulation of the QDPMO derived from that final schedule. With regards to the second objective optimization, the best robustness, which is equal to 10, is obtained. Clearly, the first and the third objective can be accessible from the final schedule of this optimization problem. As for the third objective optimization, the minimum cost, which is equal to 798, is revealed and the corresponding first and second objectives are also calculated. It can be observed that when an objective is optimized alone, the other objectives take some distance from the optimal ones.

4. Solution methodologies

As mentioned above, four multi-objective evolutionary algorithms named NSGAII, SPEAII, MOPSO, and MOEAD are utilized in this paper to solve the triple-objective problem modeled in Section 3. These algorithms are able to perform a search within the solution space of the problem to provide the DM with a non-dominated set of solutions before making the decision. All of these algorithms are developed to use similar solution representation in a hybrid structure. In other words, all the algorithms consist of a procedure to search the schedule of activities hybridized with another

Table 2
Optimization results of the example solved for each objective.

Optimization Type	Final Schedule								Objective Values		
	Activities	1	2	3	4	5	6	7	First	Second	Third
First Objective	Mode	1	1	1	1	2	3	1	6	2	813
	Start time	0	0	0	2	3	3	6			
Second Objective	Mode	1	3	1	2	3	3	1	15	10	1599
	Start time	0	0	1	5	10	13	15			
Third Objective	Mode	1	1	1	1	2	3	1	9	2	798
	Start time	0	1	1	3	4	5	9			



Time Periods	0	1	2	3	4		....	DD-2	DD-1
$l = 1$	1	0	1	0	1		....	0	0
$l = 2$	1	0	0	1	0		....	1	0
⋮	⋮	⋮	⋮	⋮	⋮		....	⋮	⋮
$l = L$	1	1	0	1	0		....	1	0

Fig. 2. The solution representation of the common genetic algorithm.

procedure to search the best material ordering policy according to each schedule. The search procedure for the quantity discount problem in material ordering is a genetic algorithm that is common, named CGA, between all the hybrid solution methodologies. The algorithms are described in the following subsections.

4.1. Solution representations

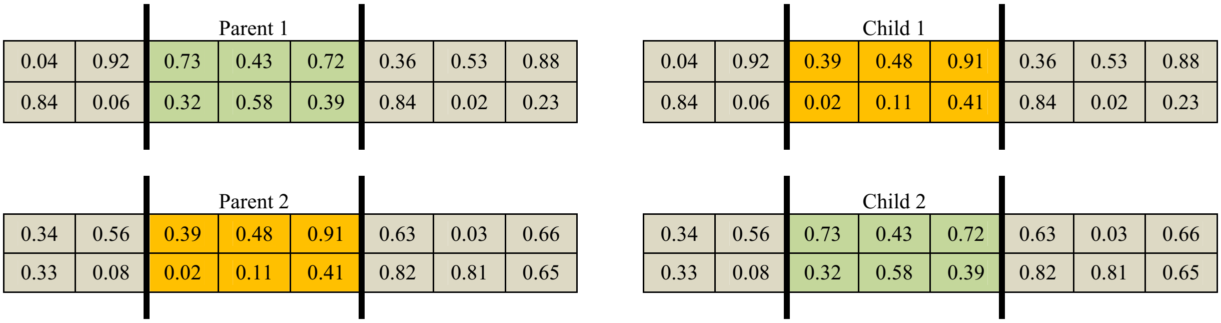
The hybrid procedures tackle the MRIP part and the QDPMO part of the model independently. Therefore, a solution representation is required for the MRIP-based procedure, and another one is needed to present a solution for the QDPMO.

An MRIP solution is represented by a 2-dimentional matrix $Y = \left[Y_{ij}\right]_{2\times N}$. The first row of the matrix relates to the schedule of activities and the second row contributes to the mode of activities. Each cell of the matrix is valued randomly within interval [0,1]. The cell Y_{1j} shows a priority value for activity j, whereas the cell Y_{2j}carries a measure for the execution mode of activity j. The activity priority values in the first row will finally be used to determine the sequence of activities based on the final scheduling is determined, whereas those in the second row will be utilized to determine the mode of the activities. In relation to the first row, at first, the activities qualified to be scheduled are listed. An activity with no unscheduled predecessor has the qualification for scheduling. For a group of qualified activities in the list, the ones with lower priority values are selected to be positioned earlier in the sequence. Then, the list of qualified activities is updated and a new activity with a lower priority is deployed in the sequence. The process continues until all the activities are appeared in the activity sequence. In addition, for each activity j, $m_j = \left[M_j \times Y_{2j}\right] + 1$ determines the mode at which activity j must be executed. With regard to the known activity sequence and execution modes, either the serial schedule generation scheme (SSGS) or parallel schedule generation scheme (PSGS) is employed to generate a schedule [45].

Each QDPMO solution consists of a 2-dimensional matrix $Q = [Q_{lt}]_{L\times(DD-1)}$. The entries of the matrix are randomly filled up with zeros or ones where one shows an order in the corresponding time period for that material and zero shows no order. Noticeably, the first entry of each row of the matrix must be filled with one. Fig. 2 shows how a QDPMO solution is represented. To decode each solution, the entries filled with ones are selected sequentially and an order equal to the sum of materials needed in the periods from the selected entry until all periods ahead with value of zero is ordered.

4.2. Non-dominated sorting genetic algorithm II (NSGAII)

Introduced by Deb et al. [46], NSGAII adds two main characteristics to the genetic algorithm to enable it to solve multi-objective problems. The first characteristic is the non-dominated sorting that ranks the members of a population in some non-dominated front classes. In the first non-dominated front class with rank 1, the members that are not dominated by any other member of the population are positioned. The second front class with rank 2 contains those who are dominated only by the members of the first class. The members of the third class with rank 3 are dominated only by the first and the second class members, and so forth. The second characteristic is the crowding distance factor that determines whether a member is located in a crowded area of known Pareto solutions or not. This factor guides the search toward a uniformly distributed Pareto front. The NSGAII starts with a random population (P_0) of MRIP solutions with the size of Z. To calculate the resulted objective values of a solution, its corresponding schedule is firstly decoded. Consequently, the makespan (first objective) of the project is automatically obtained from the end activity of the project. The robustness (second objective) of the schedule is also easily achieved by summing up the free slack of activities. For the cost function (third objective), the usage profile of renewable resources and materials regarding that schedule is firstly calculated. Three former parts of this objective associated with QDPMO are obtained by running the CGA that finds the best ordering plans of the materials. The last part of this objective is calculated by multiplying the maximum usage of renewable resources over the project duration to their related employment costs. After calculating the objective functions, the members of the population are ranked in some non-dominated fronts and their crowding distance factors are also calculated. Now, for generating the next population (P_i), at first, a population (Q_{i-1}) with the size of Z is generated by joining the members of the previous population (P_{i-1}). For this sake, the tournament selection mechanism, the two-point crossover, and a mutation operator are hired. The tournament selection mechanism selects three members of P_{i-1} at random and then choose one of them based on the rank of classes to which they belong and the crowding distance of members. Actually, the one belonging to lower ranked classes and within a class, the one with the smaller crowding distance has the higher chance of selection. The two-point crossover operator, as represented in Fig. 3, creates two children from two parents selected by the tournament selection mechanism. This operator breaks two selected members (parents) from two randomly chosen points and replaces the columns between two points. Then, the mutation operator depicted in Fig. 4 selects randomly two points of each generated child and reverses the values of cells between these two points in each row of the



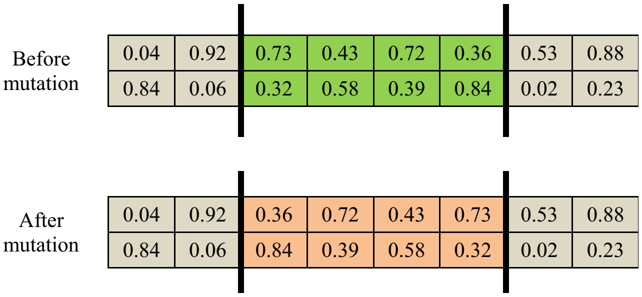
		Parent 1					
0.04	0.92	0.73	0.43	0.72	0.36	0.53	0.88
0.84	0.06	0.32	0.58	0.39	0.84	0.02	0.23

		Child 1					
0.04	0.92	0.39	0.48	0.91	0.36	0.53	0.88
0.84	0.06	0.02	0.11	0.41	0.84	0.02	0.23

		Parent 2					
0.34	0.56	0.39	0.48	0.91	0.63	0.03	0.66
0.33	0.08	0.02	0.11	0.41	0.82	0.81	0.65

		Child 2					
0.34	0.56	0.73	0.43	0.72	0.63	0.03	0.66
0.33	0.08	0.32	0.58	0.39	0.82	0.81	0.65

Fig. 3. A small example for the two-point crossover operator.



Before mutation

0.04	0.92	0.73	0.43	0.72	0.36	0.53	0.88
0.84	0.06	0.32	0.58	0.39	0.84	0.02	0.23

After mutation

0.04	0.92	0.36	0.72	0.43	0.73	0.53	0.88
0.84	0.06	0.84	0.39	0.58	0.32	0.02	0.23

Fig. 4. A small example for the mutation.

child. Now, both populations are combined leading to a population $R_{i-1} = P_{i-1} \cup Q_{i-1}$ with the size of 2*Z*. The members of the combined population are ranked and their crowding distances are calculated. Finally, the next generation (P_i)is filled with the first *Z* members of class 1, then class 2, and so on. The algorithm repeats this procedure for several iterations and then stops.

4.3. Multi-objective particle swarm optimization (MOPSO)

In addition to the features of the original PSO, the MOPSO introduced by Coello & Lechuga [47] possesses a repository for maintaining the non-dominated solutions. The algorithm also uses an adaptive gridding mechanism, which divides the coverage area of the discovered Pareto solutions into some grids to recognize the density of the grids helping the search to proceed towards less-explored areas. The density of a grid is directly related to the number of Pareto solutions that it owns. Similar to the original PSO, the MOPSO starts with a random swarm (P_0) of particles, where each particle which is an MRIP solution is produced as described previously. Each particle is then assessed and its corresponding objectives are calculated. Based on the known objectives, the non-dominated solutions are obtained and saved in the repository. The current position of each particle is also set to its best position (P^*). For the global best position of the population termed in the MOPSO as the leader (L^*), the triple-dimensional solution space is gridded with hyper-cubes to determine the position of non-dominated solutions in the grids. Then, one grid is selected via the roulette-wheeled selection and one of the non-dominated solutions located in that grid is introduced as the leader randomly. In the roulette-wheeled selection, the fitness of the grids is directly contributed to their density calculated as $^1\!/_{n_j}$ for grid *j* where n_j is the number of non-dominated solutions located in the grid *j*. Now, the first step of the algorithm finishes and the next swarm is created. For this purpose, the initial velocities regarding each position of the particles are randomly produced using Eq. (18).

$$V_{ij}^{0} = V_{\min} + (V_{\max} - V_{\min}) \times r \tag{18}$$

In this equation, $V_{\min} = -0.1$, $V_{\max} = 0.1$, and *r* is a uniform random value chosen in interval [0,1]. Then, the velocities and particles are updated to settle into new positions as follows.

$$V_{ij}^{t+1} = \omega \times V_{ij}^{t} + c_1 \times r \times \left(P_{ij}^{*} - X_{ij}^{t}\right) + c_2 \times r \times \left(L_{ij}^{*} - X_{ij}^{t}\right) \tag{19}$$

$$X_{ij}^{t+1} = X_{ij}^{t} + V_{ij}^{t+1} \tag{20}$$

where *X* stands for particles, *t* shows a step of the algorithm, c_1 and c_2 are accelerating constants controlling the movement of the particles toward P^* and L^*, and ω is the inertia weight that controls the influence of previous velocities on the new ones.

The new swarm is then filled with new particles. Typically, each new swarm encompasses some new non-dominated solutions that are compared with the old ones in the repository, where the winners stay alive. In the MOPSO, the repository has a limited capacity. If the number of non-dominated solutions exceeds the capacity, some of them are randomly removed in such a way that those located in the denser grids are removed with higher probability. In addition, the P^* and L^* are also updated. When updating the P^*, if the new position of each particle dominates P^*, it takes the position of P^*, if none of them can dominate each other, one is chosen at random. This procedure proceeds for a predetermined number of steps.

4.4. Strength pareto evolutionary algorithm II (SPEAII)

The SPEAII has been introduced by Zitzler et al. [48]. The algorithm starts with a randomly produced population (P_0) of MRIP solutions with the size of *Z* plus an empty archive $\left(\overline{P_0}\right)$ with the size of $\overline{Z}$. In each step *t* of the algorithm, the fitness of members P_t and $\overline{P_t}$ are calculated using the following three conceptions.

4.4.1. Strength criterion

$S(i)$ is known as the strength of the member *i* calculated as the number of members in $P_t \cup \overline{P_t}$, which are dominated by *i* as represented in Eq. (21), where $i \succ j$ shows *i* dominates *j*.

$$\underset{i\in P_t\cup\overline{P_t}}{S(i)} = \left|\left\{j|j \in P_t \cup \overline{P_t} \wedge i \succ j\right\}\right| \tag{21}$$

4.4.2. Raw fitness criterion

$R(i)$ is known as the raw fitness of the member *i*, which is calculated by the strengths of its dominators according to Eq. (22).

$$\underset{i\in P_t\cup\overline{P_t}}{R(i)} = \sum_{k\in P_t\cup\overline{P_t},\, k\succ i} S(k) \tag{22}$$

4.4.3. Density criterion

$D(i)$ is known as the density of the member *i*, which is a function of the distance to the k^{th} nearest neighbor of *i*, σ_i^k, as shown in Eq. (23) where $k = \sqrt{Z + \overline{Z}}$.

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{23}$$

Finally, the fitness of the member *i* is calculated by adding its raw fitness and density, i.e., $F(i) = R(i) + D(i)$. Afterward, all of the non-dominated solutions in P_t and $\overline{P_t}$ are copied to $\overline{P_{t+1}}$. If the size of $\overline{P_{t+1}}$ exceeds $\overline{Z}$, then some of them are eliminated by the truncation operator. Otherwise, if the size of $\overline{P_{t+1}}$ is less than $\overline{Z}$, then it is filled with the best dominated solutions in P_t and $\overline{P_t}$, which are those with smaller fitness. The truncation operator removes those non-dominated solutions with minimal distance to another member, and in case of several members with the same minimal distance, the second smallest distance is considered and so forth. Now, one step finishes and the other starts by selecting the parents from $\overline{P_{t+1}}$ to be recombined with crossover and mutation operators and create the new population P_{t+1}. This process continually runs until the stopping criterion is met.

The SPEAII developed here uses a population of MRIP solutions (P_0) along with an archive of non-dominated solutions $\left(\overline{P_0}\right)$ with a limited capacity. All the procedures described above are executed stepwise within this SPEAII to create new populations. In this regards, the parents are selected by the tournament selection to be recombined with the crossover operator described in Fig. 3 and the mutation operator depicted in Fig. 4.

4.5. Multi-objective evolutionary algorithm based on decomposition (MOEAD)

The mechanism involved in *MOEAD* is based on the idea that any Pareto optimal solution could be an optimal solution to a scalar optimization problem. In each scalar optimization problem, a single objective that is an integration of all objectives of the original multi-objective problem is solved. This algorithm decomposes a multi-objective optimization problem into Z scalar optimization sub-problems and then solves each sub-problem simultaneously based on a neighborhood search. There are different methods such as the weighted sum approach and the Tchebycheff approach [49] for converting a multi-objective problem into scalar optimization sub-problems. For example, let $\lambda^1, \lambda^2, ..., \lambda^Z$ be a set of even spread weight vectors, each for one sub-problem, and $f^* = \left(f_1^*, f_2^*, \ldots, f_N^*\right)$ be the reference point including optimal values of N objective functions. In the Tchebycheff approach, the single objective function of sub-population z shown in Eq. (24) is optimized subject to original constraints of the multi-objective problem at hand.

$$\textit{Minimize} \quad \max_{1 \le i \le N} \left\{ \lambda_i^z | f_i^z(x) - f_i^* | \right\} \tag{24}$$

In fact, for each Pareto solution x, there exist a weight vector λ such that x is an optimal solution to (24) and each optimal solution of (24) is a Pareto solution for the original multi-objective problem. Therefore, different Pareto solutions are obtained by changing the weight vectors.

The algorithm produces Z even spread weight vectors and accordingly generates a random population of Z problem solutions, $x^1, x^2, ..., x^Z$. Then, the Euclidean distances between all the weight vectors are calculated and the T closest weight vectors to each weight vector $i = 1, ..., Z$are selected. Let $\lambda^{i_1}, ..., \lambda^{i_T}$be the T closest weight vectors to λ^i and $B(i) = \left\{i_1, \ldots, i_T\right\}$ be their corresponding solutions. Then, for each member i of the population, two indices k, $l \in B(i)$ are selected randomly and a new solution y is produced by combining the two selected solutions, x^k and x^l. The new solution y is improved by a heuristic method to produce y'. The reference point is obtained and updated through the algorithm run. When a y' gives birth to, if one of its objective functions or more outperforms those of reference point, an update in the reference point is done. In addition, if y' can reach to a better Tchebycheff result compared to $j \in B(i)$, then it will take the position of j in $B(i)$. This procedure is executed in each iteration of the algorithm for all of the Z weight vectors. The algorithm stops when a predefined number of iterations is executed. Moreover, MOEAD maintains all the non-dominated Pareto solutions and compare them with any new y' to update the non-dominated set.

In the MOEAD utilized in this paper, after creating Z numbers of even spread weight vectors and Z numbers of MRIP solutions, each y is produced by the two-point crossover shown in Fig. 3 that is applied on the corresponding x^k and x^l. In addition, an improvement operator is designed to create y' in the following way. In this operator, an activity of the schedule y is selected at random and its execution modes are changed until all of the modes are examined. In case of a change on the mode, the new schedule and the new material ordering are obtained and the corresponding objectives are calculated. Finally, if the solution obtained from one mode can dominate the others, it is set to y', otherwise the solution resulted from the mode with better Tchebycheff objective function is selected as y'.

4.6. Common genetic algorithm (CGA)

As stated before, all of the hybrid methodologies share a common genetic algorithm to tackle the QDPMO. After a random creation of a population of QDPMO solutions as explained earlier, the parents are selected by the roulette-wheel selection operator. Then, two new QDPMO solutions are produced via an arithmetic crossover that is performed on a pair of parents. In this crossover, two parents Q^1 and Q^2 are selected. Then, each gene of two child q^1 and q^2 is produced via Eqs. (25) and (26) in which γ is a random value drawn from interval [0,1].

$$q_{ft}^1 = (\gamma)Q_{lt}^1 + (1-\gamma)Q_{lt}^2 \tag{25}$$

$$q_{ft}^2 = (1-\gamma)Q_{lt}^1 + (\gamma)Q_{lt}^2 \tag{26}$$

In equations above, l shows the l^{th} material and t shows the time period. To mutate each new solution, a material and one of its entries is randomly selected. Then, its value changes from one to zero or from zero to one. This process should be iterated several times to stop the algorithm.

5. Experimentation

In this section, the way the test problems are generated is explained first. Then, some well-known metrics in the literature are taken to compare the multi-objective solution algorithms. Afterward, the parameter calibration process of the algorithms is detailed. Finally, the computational results of the calibrated algorithms on the test problems are analyzed using the performance metrics.

5.1. Test problems

A set of 480 multimode resource constrained project scheduling problem benchmarks available in the project scheduling problem library (PSPLIB) is selected to be used. As the benchmarks lack renewable resource costs, ordering and holding costs, and price of materials, they are generated as follows. The employment cost of each renewable resource is drawn from [100,200], the holding costs are selected from interval [1,10], the ordering costs are selected form [100,500]. In addition, 2–4 quantity discount ranges are defined for each material where the upper limits on the discount ranges 1, 2, 3, and 4 are randomly generated within intervals (80,85], (75,80], (70,75], and [65,70], respectively. The lower limit of discount range 1 is set to zero and the lower limits of the other ranges are set to one unit beyond the upper limit of their previous range. For example, the lower limit of discount range 2 is set to the upper limit of discount range 1 plus 1. The prices of the materials are also selected at random according to discount ranges in such a way that the smaller the label (1,2,3, or 4) of a discount range within which the order is made, the higher the price of materials purchased in that range.

5.2. Performance metrics

A myriad of metrics has been introduced in the literature to compare non-dominated set of solutions, each evaluating different features of the Pareto fronts. Zitzler et al. [50] state that metrics should be able to assess Pareto frontier sets in three different terms including the closeness or convergence of the Pareto solutions to the Pareto-optimal front, the distribution uniformity of the Pareto frontier within the solution space, and the diversity or coverage of the Pareto frontier within the solution space. In this paper, five metrics are chosen to evaluate the non-dominated sets obtained by the utilized algorithms.

The first metric is a convergence metric defined as the error ratio by $ER = \sum_{i=1}^{|S|} E_i / |S|$, where $|S|$ is the number of vectors in the Pareto frontier S found by an algorithm, $E_i = 0$if vector i is in the Pareto-

Table 3
Variables' Descriptions.

Algorithms/Factors	A	B	C	D	E
CGA	Maximum iteration	Population size	Crossover rate	Mutation rate	–
NSGAII	Maximum iteration	Population size	Crossover rate	Mutation rate	–
SPEAII	Maximum iteration	Population size	Archive size	Crossover rate	Mutation rate
MOPSO	Maximum iteration	Swarm size	Repository size	No. of hypercubes	–
MOEAD	Maximum iteration	No. of sub-problems	No. of closest weight vectors (T)	–	–

optimal front S^*, and $E_i = 1$ otherwise. The lesser the value of ER, the better the Pareto-frontier.

The second metric employed is the general distance (GD) that estimates how far a Pareto frontier S is away from the Pareto-optimal front S^* using Eq. (27). Lesser values of GD are desired.

$$GD(S) = \frac{1}{|S|}\sum_{x\in S} Min\left\{D_{xy}|y \in S^*\right\} \tag{27}$$

In Eq. (27), $D_{xy}$is the Euclidean distance from vector x to vector y defined in Eq. (28).

$$D_{xy} = \sqrt{\sum_{i=1}^{U}(f_i(x) - f_i(y))^2} \tag{28}$$

Moreover, $U$is the number of objectives and $f_i(x)$shows the value of the i^{th} objective in vector x. Interested readers are referred to David & Van Veldhuizen [51] for more details about ER and GD.

The third metric known as the spread is usually denoted by Δ and determines the diversity or extent of S [52]. Eq. (29) shows how Δ is calculated.

$$\Delta = \frac{\sum_{i=1}^{U} D(e_i, S) + \sum_{x\in S}|D(x, S) - \overline{D}|}{\sum_{i=1}^{U} D(e_i, S) + |S|\left(\overline{D}\right)} \tag{29}$$

Such that

$$D(x, S) = \underset{y\in S, y\neq x}{Min} \|f(x) - f(y)\|^2;\ \overline{D} = \frac{1}{|S^*|}\sum_{x\in S^*} D(x, S) \tag{30}$$

Where $D(x, S)$ is the Euclidean distance between consecutive vectors in S and $(e_1, e_2, \ldots, e_N)$are the extreme solutions of U objectives in S^*. The closer the value of Δ to zero, the more ideal the distribution of S.

The fourth metric, $D1_R$, defined in Eq. (31), considers the extent and the convergence of a Pareto frontier simultaneously [53]. Lesser values of $D1_R$ are also desired.

$$D1_R(S) = \frac{1}{|S^*|}\sum_{y\in S^*} Min\left\{D_{xy}|x \in S\right\} \tag{31}$$

The fifth metric known as spacing (SP) introduced by Schott [54] measures how uniformly the vectors are distributed. Eq. (32) defines this metric. The SP values closer to zero indicate higher uniformity.

$$SP = \sqrt{\frac{1}{|S| - 1}\sum_{i=1}^{|S|}\left(D_i - \overline{D}\right)^2} \tag{32}$$

In Eq. (32), $D_i$is calculated according to Eq. (33) and $\overline{D}$ is the mean of all D_is.

$$D_i = \underset{j}{Min}\sum_{i=1}^{U}|f_i(x) - f_j(x)|;\ \ i, j = 1, 2, \ldots, |S| \tag{33}$$

5.3. Parameter calibration

The efficiency of meta-heuristic algorithms depends highly on the values chosen for their parameters. That is why the response surface methodology (RSM) is used in this paper to calibrate the parameters of all algorithms. As a combination of statistical and mathematical techniques, RSM is employed for optimization especially in situations where the quality of response(s) is highly influenced by some input variables [55]. RSM optimizes the statistically estimated response function(s) by experimenting different input variables. For this sake, a full or fractional factorial design is first required to introduce different combinations of input variables for experimentations. Then, and after the experimentations, the response function(s) is estimated to be solved mathematically divulging the best level of input variables leading to the best response(s).

All the four utilized algorithms incorporate two approaches where the CGA is responsible in all for the QDPMO and works independent of the complement approach liable for the MRIP. Therefore, the parameters of the CGA are tuned independently. This makes a total of five algorithms, NSGAII, SPEAII, MOPSO, MOEAD and CGA, for which the calibration is taken place. The parameters of these algorithms that are subject to calibration are introduced in Table 3 and their corresponding domains are shown in Table 4.

Then, a central composite face centered (CCF) design including a full or fractional factorial design with some central and axial points as shown in Tables 5–7 is used for each algorithm. Note that the CGA is calibrated using one response, while the others are calibrated to optimize five responses defined as the performance metrics explained before. The only response chosen for the CGA is the material-related costs of the project calculated by the third objective of the model while its last part that relates to the renewable resource costs is omitted. Noticeably, the response of the CGA is normalized according to the best response known among the

Table 4
Variables' Domain.

Factors/Algorithms	CGA	NSGAII	SPEAII	MOPSO	MOEAD
A	[50,250]	[20,40]	[20,40]	[20,40]	[20,40]
B	[25,50]	[15,30]	[15,30]	[15,30]	[15,30]
C	[0.6,0.99]	[0.6,0.99]	[30,60]	[40,80]	[4,6]
D	[0.025,0.075]	[0.025,0.075]	[0.6,0.99]	[5,15]	–
E	–	–	[0.025,0.075]	–	–

Table 5
Results of CCF Design for CGA, NSGAII, and MOPSO.

A	B	C	D	CGA	NSGAII					MOPSO				
					ER	*GD*	$D1_R$	Δ	*SP*	*ER*	*GD*	$D1_R$	Δ	*SP*
−1	1	1	−1	0.22	0.98	2556	384	0.98	694	1.00	3985	930	0.96	679
0	−1	0	0	0.09	0.96	3015	399	0.97	856	0.96	2690	426	0.98	1889
0	0	0	0	0.08	0.93	1050	420	0.97	628	0.84	1417	485	0.98	745
−1	−1	1	1	0.14	0.92	1942	329	0.98	932	0.86	2526	286	0.97	554
1	−1	1	−1	0.04	0.92	1397	342	0.97	531	0.80	2286	214	0.97	417
−1	0	0	0	0.18	0.90	2429	793	0.97	964	0.96	1731	301	0.98	1318
1	−1	−1	1	0.02	0.87	1210	325	0.99	1017	0.94	1604	367	1.00	2084
0	0	−1	0	0.09	0.89	1058	336	0.96	623	0.97	1930	482	0.98	907
1	1	−1	−1	0.10	0.69	907	231	0.99	778	1.00	2989	346	0.98	1002
−1	1	−1	1	0.17	1.00	1718	362	0.98	1267	0.93	2336	495	0.97	1051
0	0	0	0	0.08	0.97	1044	384	0.97	794	0.93	2502	342	0.98	1200
1	1	1	1	0.01	0.96	1273	260	0.99	1049	0.79	1519	292	1.00	1911
1	0	0	0	0.03	0.70	1090	356	0.98	777	0.89	2086	580	0.97	1125
0	1	0	0	0.08	0.82	1950	586	0.95	883	0.79	1726	318	1.01	2059
0	0	0	1	0.06	0.79	1119	323	0.96	593	0.86	1177	350	1.00	1894
−1	−1	−1	−1	0.19	0.96	1263	566	0.99	956	0.92	1516	341	1.00	1950
0	0	0	0	0.08	0.80	1193	287	0.97	649	0.54	1500	167	1.00	1167
0	0	1	0	0.07	0.54	1721	152	0.95	559	0.95	698	434	0.98	698
0	0	0	−1	0.11	0.78	2442	258	0.96	579	0.64	1623	199	0.99	1003

Table 6
Results of CCF Design for SPEAII.

A	B	C	D	E	*ER*	*GD*	$D1_R$	Δ	*SP*
−1	1	−1	−1	1	0.99	1298	317	1.01	1971
−1	−1	1	1	−1	0.93	879	528	0.97	899
0	0	0	0	0	0.99	2428	597	0.97	1167
1	−1	−1	−1	−1	0.78	1645	278	0.98	1406
0	0	0	0	1	1.00	3963	1117	0.98	2007
1	−1	1	−1	1	0.99	1434	340	0.96	771
0	0	0	0	0	0.98	1509	394	0.97	728
1	1	−1	1	−1	0.95	1819	455	0.97	518
−1	0	0	0	0	0.95	1350	510	0.97	815
1	0	0	0	0	0.98	1021	341	0.97	488
0	−1	0	0	0	0.80	919	237	1.00	1363
0	1	0	0	0	0.96	970	628	0.98	806
0	0	0	1	0	0.84	1162	229	0.96	463
−1	−1	−1	1	1	0.94	734	480	1.01	1434
0	0	0	0	0	0.87	1611	426	1.01	1377
0	0	−1	0	0	0.77	1795	216	0.97	617
−1	1	1	−1	−1	0.92	709	324	1.02	1492
0	0	1	0	0	0.82	929	248	0.98	1565
0	0	0	−1	0	0.95	1730	532	0.98	1178
1	1	1	1	1	0.89	1203	341	0.97	660
0	0	0	0	−1	0.76	953	382	0.97	625

Table 7
Results of CCF design for MOEAD.

A	B	C	*ER*	*GD*	$D1_R$	Δ	*SP*
−1	−1	−1	0.92	2404	728	1.04	2149
−1	−1	1	0.93	2127	1235	0.97	1902
−1	1	−1	0.82	2897	1654	0.97	1420
−1	1	1	0.85	2288	1955	1.00	1570
1	−1	−1	0.58	4505	860	0.99	857
1	−1	1	0.81	2400	1504	1.03	4141
1	1	−1	0.96	2536	1802	1.01	3644
1	1	1	0.47	2128	976	1.03	4076
0	0	0	1.00	3661	1921	0.95	2065
0	0	0	0.88	1719	894	1.02	2377
0	0	0	0.80	1702	2102	1.07	3495
−1	0	0	0.96	2887	1453	1.00	2210
1	0	0	0.89	2638	1105	0.96	1618
0	−1	0	0.67	2691	742	0.99	2139
0	1	0	0.67	2645	1072	0.97	1522
0	0	−1	0.49	2692	815	1.01	1245
0	0	1	0.71	2051	624	0.99	2117

Table 8
Favorable Level of Parameters.

Algorithms/Factors	A	B	C	D	E
CGA	250	25	0.99	0.075	–
NSGAII	40	29	0.845	0.075	–
SPEAII	40	26	30	0.76	0.075
MOPSO	27	24	80	15	–
MOEAD	40	30	6	–	–

whole levels of the design for each problem examined. As for the other algorithms, the five previously introduced metrics are first used as the responses. Then, a multi-objective optimization problem is taken into consideration. This is another novelty in this study since, to the best of our knowledge; no research has considered such calibration for multi-objective evolutionary algorithms yet.

Tables 5–7 show the experimental results of 25 instances with different levels of complexity, i.e., those with different number of activities, renewable resources, and materials. The values of the response columns in these tables are calculated as the mean of the response values in 25 tested instances in each level of the CCF. Meanwhile, in order to calculate the only response of the CGA, at first a schedule is known with one of the algorithms designed for MRIP. Then, the CGA is executed to find the best corresponding ordering plan that engenders the response.

Using the results in Tables 5–7 we conduct five analyses of variance, each relating to one algorithm to estimate a second order function for each response of the algorithms. Afterward, there are a single objective mathematical model for the CGA and four quintuple-objective mathematical models for the other four algorithms to be optimized. While the single objective model is easy-to-deal-with, the multi-objective models are onerous for which the fuzzy goal programming (FGP) method is employed. The FGP, introduced by Tiwari [56], transforms all the objectives into a single one defined as the weighted sum of achievement degrees of the objectives with respect to their target values. An achievement degree is defined via a fuzzy single utility function for each objective, where the target of 1 is acquired only if the objective reaches to as same level as when it is optimized solely. The favorable levels of the parameters of the algorithms obtained by solving the corresponding single objective or FGP model are presented in Table 8.

6. Computational results and comparisons

In this section, the parameter-tuned algorithms are first implemented on a set of 480 test problems and then are compared in terms of the five performance metrics defined. Afterward, a statistical analysis that aims to recognize significant differences among the algorithms is conducted. Finally, a novel evaluation and comparison based on the ideal solutions of each objective is conducted.

6.1. Comparisons based on pareto solutions

The test problems considered here are of different complexity levels in terms of the number of activities and resources. They range from problems with 12–32 activities (N), 1–5 modes (M) for each activity execution, 1–3 renewable resources (K), and 3–5 nonrenewable resources (L). Table 9 summarizes the results of the algorithms implemented on 480 test problems. In this table, there are 16 different problem sets each with 30 different instances. The data in each cell of the table is the average of the corresponding metric obtained from these instances.

Table 9
Performances of the algorithms in terms of five metrics.

No. of Problem	N	M	L	K	ER				GD				D1R				Δ				SP			
					SPEAII	NSGAII	MOPSO	MOEAD	SPEAII	NSGAII	MOPSO	MOEAD	SPEAII	NSGAII	MOPSO	MOEAD	SPEAII	NSGAII	MOPSO	MOEAD	SPEAII	NSGAII	MOPSO	MOEAD
30	18	3	2	2	0.88	0.03	0.93	0.97	1364	92	2348	3395	869	10	1389	3628	1.00	1.07	0.97	0.98	1230	578	854	2542
30	12	3	2	2	0.80	0.08	0.98	0.96	527	148	1083	2062	404	54	643	1834	1.00	1.05	0.98	1.00	837	421	743	2169
30	14	3	2	2	0.83	0.06	0.92	0.95	919	482	1757	2907	505	32	794	2642	0.99	1.06	0.97	0.97	826	490	840	1785
30	16	3	2	2	0.88	0.02	0.94	0.98	1128	150	2298	4238	958	3	1700	3465	0.98	1.06	0.97	0.96	1011	424	1087	2065
30	18	3	2	2	0.82	0.07	0.93	0.97	1107	123	2107	3956	643	39	1067	3143	0.99	1.06	0.97	0.97	972	533	1197	2027
30	20	3	2	2	0.84	0.04	0.93	0.94	1263	209	2025	3455	595	13	864	3003	0.98	1.06	0.97	0.98	1072	670	839	2289
30	22	3	2	2	0.89	0.05	0.94	0.95	1574	302	2876	4304	825	28	1248	3401	0.98	1.05	0.96	0.97	1366	607	1067	2812
30	32	3	2	2	0.83	0.02	0.95	0.94	2561	395	3710	6010	1042	8	1840	3447	0.97	1.05	0.96	0.98	1439	753	1299	2910
30	18	2	2	2	0.78	0.10	0.84	0.96	579	239	627	1798	496	39	601	2419	1.00	1.04	0.99	1.00	897	475	751	2107
30	18	4	2	2	0.89	0.03	0.93	0.90	1821	666	3062	2904	1033	14	1478	2869	0.97	1.03	0.96	0.98	1157	454	1052	2775
30	18	5	2	2	0.91	0.02	0.96	0.91	2009	183	3354	3766	1040	9	1922	3341	0.97	1.06	0.96	0.98	906	455	1245	3302
30	18	3	1	2	0.76	0.04	0.89	0.95	523	61	822	1676	442	10	719	2202	0.99	1.13	0.93	0.94	603	238	435	1363
30	18	3	3	2	0.86	0.02	0.94	0.97	1288	217	3143	5062	1463	8	1998	5356	0.98	1.07	0.98	0.97	1308	646	2204	3656
30	18	3	2	3	0.86	0.03	0.96	0.96	1055	107	2270	4027	864	14	937	3688	0.98	1.06	0.97	0.98	1072	476	917	2843
30	18	3	2	4	0.88	0.02	0.97	0.96	1249	201	2486	3538	993	5	1514	3201	0.96	1.50	0.91	0.98	1018	561	1033	2451
30	18	3	2	5	0.91	0.01	0.96	0.97	1353	74	2279	3556	775	3	1575	3446	0.98	1.06	0.97	0.98	1098	521	1047	2312
Average					0.85	0.04	0.94	0.95	1270	228	2266	3541	809	18	1268	3193	0.98	1.09	0.96	0.98	1051	519	1038	2463

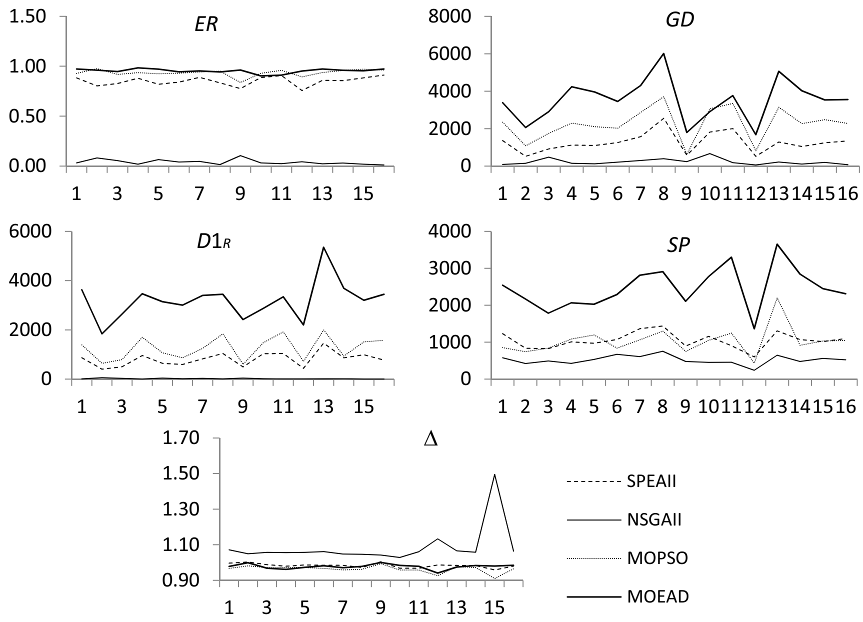


Fig. 5. The visual comparison of algorithms based on 5 metrics.

The results in Table 9 clearly show that the NSGAII outperforms the other algorithms in 4 out of 5 metrics including *ER*, *GD*, $D1_R$, and *SP* where a considerable distinction is observed between the results of NSGAII and the others.

In terms of *ER*, the results obtained by NSGAII never exceed the value of 0.1 stating that, at the worst situation only 10% of the members of Pareto frontier produced by the NSGAII do not belong to the optimal Pareto frontier. An optimal Pareto frontier is obtained by combining the Pareto frontiers of four algorithms and maintaining those that are not dominated at all. While the *ER* of NSGAII for all of problem sets is less than 0.1, those of the SPEAII, MOPSO, and MOEAD are never less than 0.7.

In terms of *GD*, the average *GD* provided by the NSGAII, 228, is far better than that of SPEAII, 1270, MOPSO, 2266, and MOEAD, 3541. While the *GD* of NSGAII only exceeds 400 in two problem sets, the *GD* of SPEAII and MOPSO never falls below 500, 600, and 1600, respectively, in all of the 16 problem sets.

The same conclusion with a more outstanding difference can be made in terms of $D1_R$ where the ones for SPEAII, MOPSO, and MOEAD are more than 44, 70, and 177 times, respectively, greater than the one of NSGAII while a smaller measure is desired.

In terms of *SP*, similar results are reported where the average measure of the NSGAII is less than 520 while the similar measure for the other three algorithms jumps over 1000.

For the remaining metric, Δ, surprisingly the MOPSO has denoted the better performance with a negligible distinction from SPEAII and MOEAD, both performing better in terms of this metric compared to NSGAII.

It should be mentioned that the SPEAII produces better results than the MOPSO in *ER*, *GD*, and $D1_R$ metrics and for the *SP*, the MOPSO has performed a little better. In addition, the results of MOPSO have been better than MOEAD in terms of all five metrics.

For a better understanding of these comparisons, a visual comparison can be made in Fig. 5. According to Fig. 5, the egregious distinction between the results of NSGAII and the other algorithms in terms of the *ER*, *GD*, and $D1_R$ is obviously observed. The superiority of the NSGAII in relation to the *SP* is also apparent with a slighter distinction. In addition, Fig. 5 shows how MOPSO functions better in terms of the Δ metric.

6.2. Statistical comparison

The differences among the four proposed algorithms in terms of the means of the employed metrics are statistically investigated in this section. For each metric, an analysis of variance (ANOVA) is conducted to test the equality of the metric means obtained by the algorithms against the inequality of the means. In each test, the two hypotheses are:

$$\begin{cases} H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4 \\ H_1 : \mu_i \neq \mu_j;\ i, j \in \{1, 2, 3, 4\};\ i \neq j \end{cases} \tag{34}$$

On one hand, the null hypothesis in (34) presents the equality of the means of the algorithms in terms of a specific metric emphasizing no significant differences among them. On the other hand, the alternative hypothesis in (34) conjectures the opposite situation. Five ANOVA tests are conducted based on the data located in Table 9, each for one metric. The p-values resulted from the ANOVAs denote that all of the null hypotheses are rejected at 95% confidence level, indicating that algorithms produce significantly different results. To discover how the algorithms differ from each other when a significant difference exists, a post-hoc analysis based on the least significant difference (LSD) is performed [57]. Fig. 6 depicts the lower and the upper limits of the metrics produced by each algorithm shown in box plots. Principally, if the boxes do not cross each other, a significant difference between the algorithms is concluded. Looking at the *ER* section of Fig. 6, a significant difference is observed among all algorithms as no overlap is observed between the boxes. In this metric, the NSGAII leads the others followed by the SPEAII, MOPSO, and MOEAD. In terms of *GD*, no overlap is shown between all algorithms confirming again a statistical significant difference between the algorithms. In this case, a similar arrangement is established with the flagship of the NSGAII followed by the SPEAII, MOPSO, and MOEAD. In terms of $D1_R$, similar consequence and arrangement is witnessed. As for the *SP*, the significance is acknowledged between the NSGAII and other algorithms. Since no overlap is observed between the SPEAII and MOPSO, any significant disparity between their results is rejected. Moreover, the results of MOEAD have been significantly different from the ones obtained by the other methodologies. For the remaining metric,

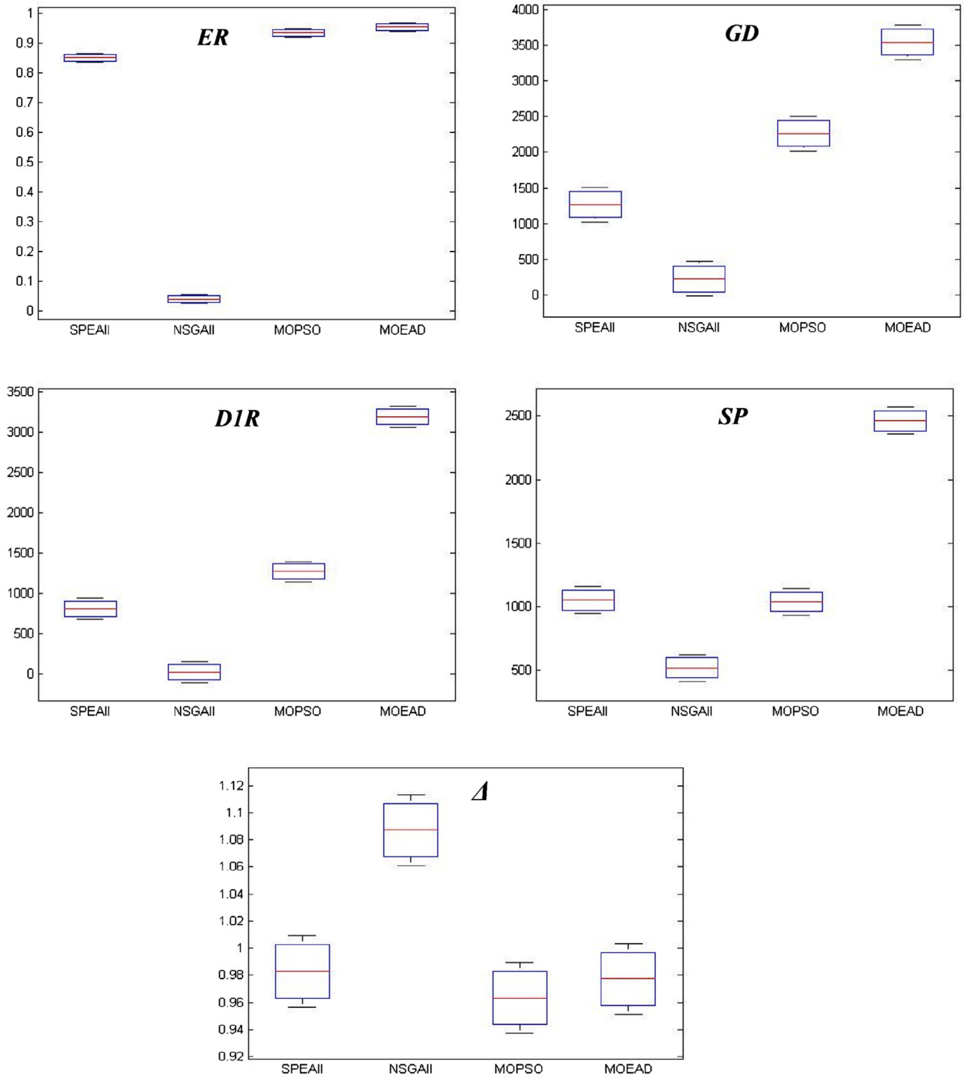


Fig. 6. Upper and lower limits of metrics via LSD 95% confidence level.

Δ, the story is completely different. In this metric, the significance is only confirmed between the NSGAII with the other algorithms where the NSGAII is positioned at the end of ranking and MOPSO is introduced as the leader with a negligible disparity with SPEAII and MOEAD.

6.3. Comparisons based on single objectives

An ideal solution for a multi-objective problem is acquired when all the objectives settle into their optimal points. The ideal solutions, if known, can be used as a criterion to evaluate how far or close the Pareto solutions are away from the ideal ones. Due to the conflicting nature of multi-objective problems, such ideal solution is inaccessible. For the problem under investigation, the ideal solutions are hard-to-access even if the objectives are considered separately because the model encompasses two NP-hard problems. On the one hand, Möhring [11] proved the NP-hardness of the simple RIP leading subsequently to the NP-hardness of the MRIP as a more complex extension of that problem. On the other hand, Cheng et al. [58] discussed the NP-hardness of the QDPMO. Therefore, meta-heuristic algorithms are utilized here to access a semi-ideal solution for each objective of the model. A GA and a PSO are designed by hiring the features of the NSGAII and MOPSO. The GA uses the previously introduced MRIP solution representation, roulette-wheel parent selection, and the crossover and mutation operators described in Section 4. The PSO also utilizes the MRIP solution representation together with the procedure explained in Section 4. Both of the algorithms are executed to find the semi-ideal solutions for all of the three objectives. Finally, the best solution discovered for each objective by the two algorithms is introduced as its semi-ideal solution.

After finding the semi-ideal solutions for each objective of the 480 test problems, two new metrics are defined upon which the quality of the Pareto solutions reported by four proposed methodologies are evaluated. The first metric is a general distance (GD) similar to Eq. (27) in which S^* contains only a vector including the semi-ideal solutions of the three objectives. Moreover, since three objectives differ from each other typically and quantitatively, a separate general distance is also calculated for each one to discover the share of each objective in the total GD calculated by the first metric. In fact, GD_i calculates the general distance of the i^{th} objective of the Pareto fronts from its semi-ideal solution while ignoring the other objectives. Table 10 summarizes the performance results of the algorithms in terms of these metrics. As anticipated, Table 10 substantiates the results of Table 9 wherein the superiority of NSGAII is still indefeasible. The NSGAII has produced better results in terms of GD, GD_2, and GD_3 with noticeable distinctions. Regarding GD_1, the results of SPEAII are closer to the ideal ones. A noticeable point

Table 10
Computational results of the algorithms based on the semi-ideal solutions of the objectives.

No. of Problem	N	M	L	K	GD				GD1				GD2				GD3			
					SPEAII	NSGAII	MOPSO	MOEAD	SPEAII	NSGAII	MOPSO	MOEAD	SPEAII	NSGAII	MOPSO	MOEAD	SPEAII	NSGAII	MOPSO	MOEAD
30	18	3	2	2	13,943	11,079	17,112	22,871	12	16	14	15	84	59	80	88	13,943	11,076	17,112	22,870
30	12	3	2	2	8034	5983	9600	13,087	9	9	9	10	25	22	27	30	8034	5981	9600	13,087
30	14	3	2	2	10,192	8318	12,995	17,890	10	11	11	12	38	30	39	42	10,192	8317	12,995	17,889
30	16	3	2	2	13,465	9387	16,130	21,751	10	13	12	13	51	37	51	57	13,465	9385	16,130	21,751
30	18	3	2	2	13,198	11,185	17,404	22,924	11	15	13	15	61	45	62	67	13,198	11,182	17,404	22,923
30	20	3	2	2	15,419	12377	18,691	24,594	13	17	15	16	75	55	73	82	15,418	12,374	18,690	24,593
30	22	3	2	2	17,493	14,408	23,176	27,940	15	19	17	17	100	74	100	112	17,492	14,405	23,175	27,939
30	32	3	2	2	27,151	22,824	35,027	40,529	16	27	22	23	229	150	209	229	27,149	22,816	35,026	40,528
30	18	2	2	2	10,329	9476	10,802	17,088	10	13	10	11	58	44	59	65	10,328	9474	10,802	17,088
30	18	4	2	2	15,197	10,354	19,232	22,388	13	14	15	17	65	51	63	68	15,197	10,350	19,232	22,388
30	18	5	2	2	15,699	11,657	20,912	24,086	13	17	16	18	75	52	72	75	15,699	11,654	20,912	24,086
30	18	3	1	2	7879	6212	9483	13,611	12	15	12	15	62	45	65	68	7879	6210	9482	13,610
30	18	3	3	2	18,656	15,879	24,259	33,229	13	16	16	16	66	49	61	71	18,656	15,877	24,259	33,229
30	18	3	2	3	15,348	12,079	18,736	25,578	13	17	14	16	86	68	90	94	15,347	12,076	18,736	25,578
30	18	3	2	4	14,033	11,742	18,729	25,231	13	17	16	16	78	60	81	81	14,033	11,743	18,729	25,231
30	18	3	2	5	14,420	11,523	18,718	25,088	14	17	15	17	75	53	76	81	14,420	11,520	18,718	25,088
Average					14,404	11,530	18,188	23,618	12	16	14	15	77	56	75	82	14,403	11,527	18,188	23,617

is that, the measures reported by algorithms in metric GD_3 are very much close to the similar measure in *GD*. This shows that, the total *GD* is more affected by the third objective than the others.

7. Conclusion

The present study investigates a combination of the multimode resource investment project scheduling problem and the quantity discount problem in material ordering in a multi-objective circumstance. Some of the main features of the combined model are as follows. The activities of the project can be executed in different modes. They require renewable as well as nonrenewable resources. The renewable resources are always available during the project life span but in limited levels and impose some costs on the project based on their availability levels. The nonrenewable ones are purchased during the project longevity such that in each purchase an ordering cost together with the price costs of materials calculated based on a total discount are imposed on the project. The remaining units of materials per period also charge the project with holding costs. The goal of the model was to minimize the makespan, maximize the robustness of the project, and minimize the total costs simultaneously. Four powerful multi-objective evolutionary algorithms (NSGAII, SPEAII, MOPSO, MOEAD) were developed to search through the solution space of the problem and find the best Pareto fronts. The algorithms hybridize two approaches; one searching among the schedules of the activities of the project scheduling compartment of the model and another navigating through the potential material ordering plans for the schedules. The former approach is unique for each methodology while the latter is common between all of them. The parameters of the algorithms were statistically tuned using the response surface methodology. Then, the parameter-tuned algorithms were experimented on a set of 480 test problems with different levels of complexity. The results were compared in two different viewpoints. In the first viewpoint, they were compared based on five metrics evaluating the Pareto fronts in terms of closeness to the optimal front, diversity, and variance they possess. The comparisons in this viewpoint showed that, NSGAII has brought about better performances in 4 out of 5 metrics while the MOPSO has functioned better in the other metric. In addition, an analysis of variance conducted to test the equality of the means of each metric provided by the algorithms statistically confirmed significant differences between the performances of the algorithms. In the second viewpoint, the methodologies were compared based on the semi-ideal solutions of the objectives acquired by solving the problems under each objective separately. The results of comparisons in this viewpoint showed that the Pareto fronts produced by the NSGAII were not only more close to the semi-ideal solutions of the objectives, but also a large number of semi-ideal solutions were observed among the Pareto solutions reported by the NSGAII. As future research, adding cash inflows (payments) to the project and considering the objective function (3) under net present value maximization is suggested. In addition, considering the earliness with bonus and tardiness with penalty in the project finish time can make the model more applicable to real-world projects. Developing other multi-objective evolutionary algorithms are also another possible future research.

References

[1] O. Icmeli, S.S. Erengüç, C.J. Zappe, Project scheduling problems: a survey, Int. J. Oper. Prod. Manage. 13 (1993) 80–91.

[2] S.E. Elmaghraby, Activity nets: a guided tour through some recent developments, Eur. J. Oper. Res. 64 (1995) 199–215.

[3] W.S. Herroelen, B. De Reyck, E.L. Demeulemeester, Resource-constrained project scheduling: a survey of recent developments, Comput. Oper. Res. 25 (1999) 279–302.

[4] P. Brucker, A. Drexl, R. Möhring, K. Neumann, E. Pesch, Resource-constrained project scheduling: notation, classification, models and methods, Euro J. Oper. Res. 112 (1999) 3–41.

[5] L.V. Tavares, A review of the contribution of operation research to project management, Euro. J. Oper. Res. 136 (2002) 1–18.

[6] E.L. Demeulemeester, W. Herrolen, Project Scheduling: a Research Handbook, Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

[7] R. Kolisch, S. Hartmann, Experimental investigation of heuristics for resource-constrained project scheduling: an update, Euro. J. Oper. Res. 174 (2006) 23–37.

[8] S. Hartmann, D. Briskorn, A survey of variants and extensions of the resource-constrained project scheduling problem, Euro. J. Oper. Res. 207 (2010) 1–14.

[9] V. Van Peteghema, M. Vanhoucke, An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling on new dataset instances, Euro. J. Oper. Res. 235 (2014) 62–72.

[10] C.H. Schwindt, J. Zimmermann, Handbook on Project Management and Scheduling, Springer, 2015.

[11] R.H. Möhring, Minimizing costs of resource requirements in project networks subject to a fix completion time, Oper. Res. 32 (1984) 89–120.

[12] E.L. Demeulemeester, Minimizing resource availability costs in time limited project networks, Manage. Sci. 41 (1995) 1590–1598.

[13] J. Zimmermann, H. Engelhardt, Lower bounds and exact algorithms for resource leveling problems, University of Karlsruhe, Germany, Tech. Rep. 517 (1998).

[14] H. Nübel, The resource renting problem subject to temporal constraints, OR Spektrum 23 (2001) 574–586.

[15] M. Sabzehparvar, A mathematical model for the multi-mode resource investment problem, J. Ind. Eng. Int. 4 (2008) (pp. 3).

[16] S. Shadrokh, F. Kianfar, A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty, Euro. J. Oper. Res. 181 (2007) 86–101.

[17] M. Ranjbar, F. Kianfar, S. Shadrokh, Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm, Appl. Math. Comput. 196 (2008) 879–888.

[18] A.A. Najafi, S.T.A. Niaki, M. Shahsavar, A tuned genetic algorithm for resource investment problem with discounted cash flows and generalized precedence relations, Comput. Oper. Res. 36 (2009) 2994–3001.

[19] M. Shahsavar, A.A. Najafi, S.T.A. Niaki, An efficient genetic algorithm to maximize net present value of project payments under inflation and bonus–penalty policy in resource investment problem, Adv. Eng. Softw. 41 (2010) 1021–1030.

[20] N.J. Aquilano, D.E. Smith-Daniels, A formal set of algorithms for project scheduling with critical path method material requirements planning, J. Oper. Manage. 1 (1980) 57–67.

[21] D.E. Smith-Daniels, N.J. Aquilano, Constrained resource project scheduling subject to material constraints, J. Oper. Manage. 4 (1984) 369–387.

[22] D.E. Smith-Daniels, V.L. Smith-Daniels, Optimal project scheduling with materials ordering, IIE Trans. 19 (1987) 122–129.

[23] A. Erabsi, C. Sepil, A modified heuristic procedure for materials management in project networks, Int. J. Ind. Eng.-Theory 6 (1999) 132–140.

[24] B. Aodin, A.A. Elimam, Integrated project scheduling and material planning with variable activity duration and rewards, IIE Trans. 33 (2001) 1005–1018.

[25] T. Schmitt, B. Faaland, Scheduling recurrent construction, Nav. Res. Log. 51 (2004) 1102–1128.

[26] M.S. Sajadieh, S. Shadrokh, F. Hassanzadeh, Concurrent project scheduling and material planning: a genetic algorithm approach, Sci. Iran. 16 (2009) 91–99.

[27] N. Zoraghi, A.A. Najafi, S.T.A. Niaki, An integrated model of project scheduling and material ordering: a hybrid simulated annealing and genetic algorithm, J. Optim. Ind. Eng. 28 (2012) 19–27.

[28] N. Zoraghi, A.A. Najafi, S.T.A. Niaki, Resource constrained project scheduling with material ordering: two hybridized metaheuristic approaches, International Journal of Engineering (IJE), Trans. C: Asp. 10 (2015) 896–902.

[29] N. Zoraghi, A. Shahsavar, B. Abbasi, V. Van Peteghem, Multi-mode resource-constrained project scheduling problem with material ordering under bonus–penalty policies, TOP (2016).

[30] F. Ballestín, R. Blanco, Theoretical and practical fundamentals for multi-Objective optimization in resource-constrained project scheduling problems, Comput. Oper. Res. 38 (2011) 51–62.

[31] R. Slowinski, Multiobjective network scheduling with efficient use of renewable and nonrenewable resources, Euro. J. Oper. Res. 7 (1981) 265–273.

[32] N. Nudtasomboon, S. Randhawa, Resource-constrained project scheduling with renewable and non-renewable resources and time-resource trade-offs, Comput. Ind. Eng. 32 (1997) 227–242.

[33] B. Abbasi, S. Shadrokh, J. Arkat, Bi-objective resource-constrained project scheduling with robustness and makespan criteria, Appl. Math. Comput. 180 (2006) 146–152.

[34] A. Viana, J.P. de Sousa, Using metaheuristics in multiobjective resource constraints project scheduling, Euro. J. Oper. Res. 120 (2000) 359–374.

[35] M.A. Al-Fawzan, M. Haouari, A bi-objective model for robust resource-constrained project scheduling, Int. J. Prod. Econ. 96 (2005) 175–187.

[36] F.S. Kazemi, R. Tavakkoli-Moghaddam, Solving a multi-objective multi-mode resource-constrained project scheduling with discounted cash flows, in: Proc. 6th Int Management Conf., Tehran, Iran, 2008.

[37] R.S. Aboutalebi, A.A. Najafi, B. Ghorashi, Solving multi-mode resource-constrained project scheduling problem using two multi-objective evolutionary algorithms, Afr. J. Bus. Manage. 6 (2012) 4057–4065.

[38] V. Yannibelli, A. Amandi, Integrating a multi-objective simulated annealing algorithm and a multi-objective evolutionary algorithm to solve a multi-objective project scheduling problem, Expert Syst. Appl. 40 (2012) 2421–2434.

[39] S. Khalili, A.A. Najafi, S.T.A. Niaki, Bi-objective resource constrained project scheduling problem with makespan and net present value criteria: two meta-heuristic algorithms, Int. J. Adv. Manuf. Tech. 69 (2013) 617–626.

[40] H.S. Gomes, F.D.A.D. Neves, M.J.F. Souz, Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations, Comput. Oper. Res. 44 (2014) 92–104.

[41] A. Shahsavar, A.A. Najafi, S.T.A. Niaki, Three self-adaptive multi-objective evolutionary algorithms for a triple-objective project scheduling problem, Comput. Ind. Eng. 87 (2015) 4–15.

[42] M. Hapke, A. Jaszkiewicz, R. Slowinski, Interactive analysis of multiple-criteria project scheduling problems, Eur. J. Oper. Res. 107 (1998) 315–324.

[43] R. Slowinski, B.W. Soniewiclu, J. Eglarz, DSS for multiobjective project scheduling, Eur. J. Oper. Res. 79 (1994) 220–229.

[44] Q. Zhang, H. Li, MOEA/D, a multiobjective evolutionary algorithm based on decomposition, IEEE T, Evol. Comput. 11 (6) (2008) 712–731.

[45] R. Kolisch, S. Hartmann, Heuristic algorithms for the resource-constrained project scheduling problem: classification and computational analysis, in: J. Weglarz (Ed.), Project Scheduling: Recent Models, Algorithms, and Applications, Kluwer Academic Publishers, Boston, 1999, pp. 147–178.

[46] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi objective genetic algorithm-NSGA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197.

[47] C.A. Coello Coello, M.S. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in: Proc. 2002 Congr. Evolutionary Computation, Honolulu, Hi, 2010, pp. 1051–1056.

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

[49] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer, Norwell: MA, 1999.

[50] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2000) 173–195.

[51] A. David, D.A. Van Veldhuizen, Multiobjective Evolutionary Algorithm: Classification, Analyses, and New Innovations, PhD. Thesis, Dept. Elect. Comp. Eng., Graduate School of Engineering, Force Institute of Technology, Wright-Patterson AFB, OH, USA, 1999.

[52] A.J. Nebro, F. Luna, E. Alba, B. Doronsoro, J.J. Durillo, A. Beham, AbYSS: Adapting scatter search to multiobjective optimization, IEEE Trans. Evol. Comput. 12 (2008) 439–457.

[53] J.D. Knowles, D.W. Corne, On metrics for comparing non-dominated sets, in: Proc. Int. Congr. Evolutionary Computation (CEC02), IEEE Press, New York, 2002.

[54] J.R. Schott, Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization, M.S., Thesis, Dept. Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1995.

[55] R.H. Myers, D.C. Montgomery, Response Surface Methodology: Process and Product Optimization Using Designed Experiments, John Wiley & Sons Inc, New York, USA, 1995.

[56] R.N. Tiwari, S. Dharmar, J.R. Rao, Fuzzy goal programming –an additive model, Fuzzy Set Syst. 24 (1987) 27–34.

[57] D.C. Montgomery, Design and Analysis of Experiments, John Wiley & Sons, Inc, New York, USA, 2009.

[58] C. Cheng, D.T. Chiang, C. Lu, An optimal algorithm for the quantity discount problem, J. Oper. Manage. 7 (1987) 165–177.