RAIRO-Oper. Res. 55 (2021) 2093–2128
https://doi.org/10.1051/ro/2021087

RAIRO Operations Research
www.rairo-ro.org

A MULTI-OBJECTIVE MULTI-AGENT OPTIMIZATION ALGORITHM FOR THE MULTI-SKILL RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM WITH TRANSFER TIMES

Amir Hossein Hosseinian and Vahid Baradaran*

Abstract. This paper addresses the Multi-Skill Resource-Constrained Project Scheduling Problem with Transfer Times (MSRCPSP-TT). A new model has been developed that incorporates the presence of transfer times within the multi-skill RCPSP. The proposed model aims to minimize project's duration and cost, concurrently. The MSRCPSP-TT is an NP-hard problem; therefore, a Multi-Objective Multi-Agent Optimization Algorithm (MOMAOA) is proposed to acquire feasible schedules. In the proposed algorithm, each agent represents a feasible solution that works with other agents in a grouped environment. The agents evolve due to their social, autonomous, and self-learning behaviors. Moreover, the adjustment of environment helps the evolution of agents as well. Since the MSRCPSP-TT is a multi-objective optimization problem, the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is used in different procedures of the MOMAOA. Another novelty of this paper is the application of TOPSIS in different procedures of the MOMAOA. These procedures are utilized for: (1) detecting the leader agent in each group, (2) detecting the global best leader agent, and (3) the global social behavior of the MOMAOA. The performance of the MOMAOA has been analyzed by solving several benchmark problems. The results of the MOMAOA have been validated through comparisons with three other meta-heuristics. The parameters of algorithms are determined by the Response Surface Methodology (RSM). The Kruskal–Wallis test is implemented to statistically analyze the efficiency of methods. Computational results reveal that the MOMAOA can beat the other three methods according to several testing metrics. Furthermore, the impact of transfer times on project's duration and cost has been assessed. The investigations indicate that resource transfer times have significant impact on both objectives of the proposed model.

Mathematics Subject Classification. 90B35, 91B32, 68M20.

Received September 5, 2020. Accepted May 29, 2021.

1. Introduction

In the Resource-Constrained Project Scheduling Problem (RCPSP), a group of activities which are bound together based on precedence relations are scheduled so that the project's duration is minimized. The RCPSP considers resource limitations and the scheduling process is conducted subject to finiteness of resources. In the standard RCPSP, the activities are connected by the Finish-to-Start (FS) precedence relations with zero

Keywords. Multi-agent systems, multi-objective optimization, multi-skill RCPSP, resource transfer times, TOPSIS.

Department of Industrial Engineering, Faculty of Engineering, Islamic Azad University, Tehran North Branch, Tehran, Iran.
*Corresponding author: V_baradaran@iau-tnb.ac.ir

Table 1. Comparison of the MSRCPSP-TT and the standard RCPSP.

MSRCPSP-TT (proposed model)	Standard RCPSP
The MSRCPSP-TT is a bi-objective model to minimize the make-span and total cost of a project, concurrently.	The standard RCPSP is a single-objective model to only minimize the make-span of a project.
Each worker is able to perform at least one skill.	Each worker is able to perform only one skill.
Activities have standard levels.	Activities have no standard levels.
Resource transfer times are considered.	Resource transfer times are not considered.

time-lags and resources have finite amounts in each period [68]. In the standard RCPSP, it is assumed that no transfer times are required to transfer resources between the execution sites of different activities. However, in real-world situations, a significant amount of time is needed to transfer resources between these sites. As an example, suppose that in a manufacturing project, several work pieces are needed to be manufactured. The work pieces are manufactured in separate workshops located in different sites. In this case, workers as project resources might be requested at different manufacturing sites. Hence, several hours or days might be required to transfer the workers from one site to another. As another example, a remarkable amount of time is required when there is a necessity to transfer a requested hefty equipment between different construction sites [38]. Thus, it is required to consider resource transfer times in modeling of the RCPSP so as to produce feasible schedules for real-life situations. Multi-Skill Resource-Constrained Project Scheduling Problem (MSRCPSP) is a variant of the standard RCPSP which has been studied widely in the literature [26]. In this problem, each resource possesses several skills with known familiarity levels. To complete an activity, one or more skills with predefined standard levels are needed [59,69]. The main goal of this paper is to propose a bi-objective model for the multi-skill RCPSP with resource transfer times (MSRCPSP-TT). The MSRCPSP-TT aims to schedule activities of a project with respect to precedence relations, resource finiteness, and transfer times such that the duration and total cost of the project are minimized, simultaneously. To clarify the distinctions between the MSRCPSP-TT and the standard RCPSP, their properties have been compared in Table 1.

The literature of the RCPSP shows that this problem is classified as an NP-hard problem [4]. Therefore, it is unlikely that exact methods can offer optimal solutions for large-scale instances of this problem in polynomial time [38]. This issue motivated a great number of researchers to develop methods that are capable of detecting optimal or near optimal solutions in a rational computational time. In this respect, soft computing methods, mostly heuristics and meta-heuristics are used [25,26]. One of the active topics in the field of expert systems and artificial intelligence is Multi-Agent Systems (MAS). A multi-agent system is interpreted as a loosely-coupled network of agents that cooperate with each other to obtain feasible results for complex problems [5]. Multi-agent based methods have been widely used for optimization problems and showed good performance. Hence, as another contribution of this paper, we design a Multi-Objective Multi-Agent Optimization Algorithm (MOMAOA) to solve the MSRCPSP-TT. The collaboration between agents is carried out in a grouped environment. This collaboration is modeled *via* social behavior, autonomous behavior, self-learning, and adjustment of environment. Since there are two objective functions for the MSRCPSP-TT, the MOMAOA employs the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) as a Multi-Attribute Decision Making (MADM) method in its procedures that enables the algorithm to tackle the bi-objective scope of the problem. The efficiency and reliability of the MOMAOA is tested by solving the benchmark problems known as the iMOPSE dataset. The performance of the proposed algorithm is also compared with the outputs of three well-known meta-heuristics, namely the Non-dominated sorting genetic algorithm II (NSGA-II), the Pareto Envelope-based Selection Algorithm II (PESA-II), and the Multi-Objective Particle Swarm Optimization (MOPSO) method. All algorithms are calibrated *via* the Resource Surface Methodology (RSM). This research faces the following questions that are thoroughly answered:

(1) What is the background of considering resource transfer times for the multi-skill RCPSP?
(2) What is the background of using the MADM methods in different procedures of multi-agent systems?
(3) How can the TOPSIS approach, as an MADM method, be applied to different procedures of a multi-agent optimization algorithm?
(4) Are the proposed model and algorithm capable of producing appropriate schedules for projects?
(5) How efficient is the proposed MOMAOA algorithm in solving test problems of the MSRCPSP-TT comparing to other well-known optimizers?
(6) What is the impact of transfer times on the objectives of the proposed model?

The main contributions of this research are threefold: First, a bi-objective model is mathematically formulated for the MSRCPSP, where multi-skill workers must travel between different sites to perform their assigned tasks. Therefore, resource transfer times have been considered for the proposed model. Moreover, the budget considered for completion of a project is limited. Second, to solve this model, a multi-objective multi-agent optimization method has been proposed. In this study, the agents are structured in a group organization. Similar to many other optimizers, the proposed method consists of several procedures. This research developed several TOPSIS-based methods for the procedures of the proposed algorithm. These developed TOPSIS-based methods are used for: (1) finding the leader agent in each group, (2) finding the global best leader agent, and (3) the global social behavior of the MOMAOA. Third, a great number of computational experiments have been conducted to evaluate the performance of the proposed algorithm in comparison with other efficient methods. The algorithms were tuned by means of the RSM. The rest of the paper is structured as follows: Section 2 surveys the literature of RCPSP with resource transfer times, the multi-skill RCPSP, and the multi-agent systems. Section 3 presents the mathematical formulation of the MSRCPSP-TT. The proposed algorithm is described in Section 4. Section 5 provides the numerical test problems and analysis of the outputs obtained by the algorithms. Ultimately, Section 6 concludes the paper and offers some suggestions for future studies.

2. Literature review

In this section, the most related studies of current research are briefly reviewed. Thus, the previous studies on the RCPSP with transfer times, the multi-skill RCPSP, and ultimately the multi-agent systems are surveyed.

2.1. Previous studies on the RCPSP with transfer times (RCPSP-TT)

The number of studies on the RCPSP with transfer times is very scarce. Krüger and Scholl [42] proposed some methods to solve the RCPSP with transfer times for both single-project and multi-project cases. They modeled both cases as integer linear models and developed a priority-rule based heuristic as a solution approach. In another research, Krüger and Scholl [43] developed a framework for the Resource-Constrained Multi-Project Scheduling Problem (RCMPSP) including transfer times and costs. The proposed framework includes managerial approaches to tackle resource transfers, different resource transfer types, and new roles that can be assigned to resources during these transfers. Poppenborg and Knust [61] developed a Tabu Search (TS) algorithm based on a resource flow representation for the RCPSP-TT.

2.2. Previous studies on the MSRCPSP

Different variants of the MSRCPSP have been studied in the literature. Bellenguez and Neron [3] proposed a model, where the competency of resources is different. A disjunctive-constrained multi-skill formulation has been developed by Pessan *et al.* [60] for scheduling maintenance tasks. Gutjahr *et al.* [22] integrated the learning phenomenon into the project portfolio selection problem and developed an Ant Colony Optimization (ACO) method and a Genetic Algorithm (GA) to find its solutions. Li and Womer [47] combined mixed-integer linear model with Constraint Programming (CP) and presented a Hybrid Benders Decomposition (HBD) for the multi-skill RCPSP. To eliminate resource conflicts, a cut-generating scheme has been developed. Heimerl and Kolisch [28] proposed a model for the multi-skill multi-project scheduling problem. Kazemipoor *et al.* [39] formulated the multi-skill project portfolio scheduling problem as a Goal Programming (GP) model. For each task, an infinite set of modes has been considered. Duration of projects is minimized *via* a Differential Evolution (DE) algorithm. To find appropriate solutions to the MSRCPSP, Liu and Wang [50] utilized the constraint programming method along with multiple heuristics. Mehmanchi and Shadrokh [56] examined the learning and forgetting phenomena on competency of multi-skill manpower. Tabrizi *et al.* [65] developed a bi-stage approach based on genetic procedures and path relinking methods to optimize the Net Present Value (NPV). Correia and Saldanha-da-Gama [13] focused on the cost perspective of the MSRCPSP. Montoya *et al.* [57] examined utilizing a column generation approach in the Branch and Price (B&P) method. Myszkowski *et al.* [58] embedded priority-rule based methods in the ACO. A Teaching-Learning-Based Optimization (TLBO) algorithm was developed by Zheng *et al.* [74] to schedule activities in a multi-skill environment. Javanmard *et al.* [36] combined the MSRCPSP and Resource Investment Problem (RIP) and developed a model for it. A genetic-based and a particle-swarm-based (PSO) algorithm were developed to solve large-scale instances. Maghsoudlou *et al.* [51] formulated a multi-mode based model and suggested a Multi-Objective Invasive Weeds Optimization (MOIWO) algorithm. For the multi-skill RCPSP, Maghsoudlou *et al.* [52] suggested multiple multi-objective cuckoo-search-based approaches. Chen *et al.* [11] focused on impacts of learning and forgetting phenomena on competency of manpower in a multi-project MSRCPSP. The step-deteriorating phenomenon has been embedded in the MSRCPSP by Dai *et al.* [17] and a Tabu Search method with four neighborhood structures and two mutation operators was presented to solve it. Myszkowski *et al.* [59] merged the Differential Evolution (DE) method and a greedy algorithm to detect feasible solutions of the model. Wang and Zheng [69] proposed a Fruit-fly Optimization Algorithm (FOA) that applies the TOPSIS method during the optimization. Zhu *et al.* [76] developed a Discrete Oppositional Multi-Verse Optimization (DOMVO) method for the MSRCPSP. The researchers used the path relinking method to model the black-white phase in their proposed algorithm. To enhance the quality of outputs, they utilized the opposition-based learning (OBL) approach as well. Moreover, a repairing procedure has been devised to produce feasible solutions. Laszczyk and Myszkowski [44] developed the NSGA-II with a new selection operator. They used a clone prevention approach to acquire more diverse Pareto fronts. Lin *et al.* [49] proposed a hyper-heuristic based on the genetic programming for solving test problems of iMOPSE. Hosseinian *et al.* [35] utilized the Linear Threshold Model (LTM), which is usually used in the Influence Maximization (IM) problem, to model the learning phenomenon of workers in the MSRCPSP. An improved version of the NSGA-II was suggested to optimize make-span and total costs of projects. Hosseinian and Baradaran [31] found communities of workforces that can appropriately cooperate with each other by maximizing modularity. They used a greedy algorithm to find the communities, while a Dandelion Algorithm (DA) was developed to solve the MSRCPSP. Hosseinian and Baradaran [32] focused on the multi-mode MSRCPSP and a genetic algorithm has been developed for it. For this algorithm, two new procedures have been devised to find better solutions. Furthermore, the VIKOR method has been embedded in the GA for selecting candidate solutions in order to generate new offspring. Hosseinian and Baradaran [33] proposed two new algorithms, namely the Pareto-based Grey Wolf Optimizer (P-GWO) and the Multi-Objective Fibonacci-based Algorithm (MOFA) for the MSRCPSP with deterioration effect and financial limitations. They used the Data Envelopment Analysis (DEA) in the P-GWO to update the archive of non-dominated solutions. In another research, Hosseinian and Baradaran [34] studied the MSRCPSP with Generalized Precedence Relations (GPR). In their proposed formulation, the learning phenomenon has been considered for the workforces which means that they can become more efficient by repeating their skills. The researchers have modified the Pareto Archived Evolution Strategy (PAES) to solve this problem. Dai *et al.* [18] investigated the MSRCPSP with step deterioration and proposed a Variable Neighborhood Search (VNS) method for it. Cai *et al.* [7] studied the MSRCPSP with transfer times and uncertainty skills. A robust genetic algorithm was developed for the problem. Dang Quoc *et al.* [14, 15] developed an algorithm known as the CSM (inspired by the Cuckoo Search method) and a Differential Evolution Method (DEM) for the MSRCPSP. In another research, Dang Quoc *et al.* [16] offers another version of the cuckoo search algorithm called the R-CSM for the Real-RCPSP. Tian *et al.* [67] proposed a resource-leveling operator along with a schedule-compress operator for the NTGA and MOFOA methods to improve their solutions. The former operator levels workload of employed resources, while the latter operator tries to omit idle times of resources. Table 2 summarizes different

Table 2. Summary of studies on the multi-skill RCPSP.

References	Objective function			Transfer time		Solution approach
	Make-span	Cost	Other	TT	WTT	
[3]	✓				✓	A heuristic
[60]	✓				✓	B&B
[22]			✓		✓	ACO, GA
[47]		✓			✓	HBD
[28]		✓			✓	CPLEX
[39]	✓				✓	DE
[50]	✓				✓	CP
[56]	✓				✓	CPLEX
[65]			✓		✓	GA
[13]		✓			✓	CPLEX
[57]	✓				✓	B&P
[58]	✓	✓			✓	ACO
[74]	✓				✓	TLBO
[36]		✓			✓	GA, PSO
[51]	✓	✓	✓		✓	MOIWO
[52]		✓	✓		✓	Cuckoo search
[11]	✓	✓	✓		✓	NSGA-II
[17]	✓	✓			✓	TS
[59]	✓	✓			✓	DE and Greedy algorithm
[69]	✓	✓			✓	FOA
[76]	✓				✓	DOMVO
[44]	✓	✓			✓	NSGA-II
[49]	✓				✓	GA
[35]	✓	✓			✓	NSGA-II
[31]	✓				✓	DA
[32]	✓				✓	GA
[33]	✓	✓			✓	P-GWO, MOFA
[34]	✓	✓	✓		✓	PAES
[18]	✓				✓	VNS
[7]	✓			✓		GA
[15]	✓				✓	CSM
[14]	✓				✓	DEM
[16]	✓				✓	R-CSM
[67]	✓	✓			✓	NTGA, MOFOA
This research	✓	✓		✓		MOMAOA

characteristics of previous studies. In Table 2, "TT[1]" infers that resource transfer times have been considered in proposed models, while "WTT[2]" implies that the researchers have not considered resource transfer times in their proposed formulations. Table 2 also includes the objective functions and solution approaches of previous studies.

2.3. Previous studies on the multi-agent systems

Multi-agent methods have been widely used to solve complex problems that are intractable for other methods. Brandolese *et al.* [5] proposed a multi-agent paradigm to allocate production capacity to multiple requirements. Yan *et al.* [71] utilized the MAS to schedule activities and eliminate resource conflicts through transferring

[1]Transfer times (TT).

[2]Without transfer times (WTT).

message and negotiation among agents. They introduced mobile agents so as to reduce communication cost and to increase the communication speed. Knotts *et al.* [41] developed eight agent-based algorithms to solve the multi-mode RCPSP. Each algorithm uses a priority rule to control the access of agents to resources. Böcker *et al.* [6] developed a multi-agent based scheduling model for a railway transportation system. Lee *et al.* [45] developed an MAS for short-term scheduling of resources, which are shared by several projects. The researchers developed a market mechanism called precedence cost tâtonnement (P-TâTO) for resource scheduling. The P-TâTO was also used to find precedence conflict-free schedules. In another research, Knotts and Dror [40] investigated the implementation of agent technology for large-scale multi-mode resource-constrained project scheduling problems. They introduced reactive and deliberative agents. These agents use different procedures to select execution modes of activities. A multi-agent system based on general equilibrium market mechanism was designed by He *et al.* [27] to solve large-scale instances of the RCPSP. Homberger [29] integrated a Restart Evolution Strategy (RES) with a multi-agent system for the decentralized Resource-Constrained Multi-Project Scheduling Problem (RCMPSP). Confessore *et al.* [12] proposed a market-based multi-agent system for the RCMPSP. In this study, each project represents an agent. They used a market-based method to resolve conflicts between projects and respective agents. This method is an iterative combinatorial auction process. A multi-agent system was proposed by Chen and Wang [9] for dynamic scheduling of a project. Adhau *et al.* [1] developed a multi-agent system based on an auction-based negotiation approach. This system aims at resolving resource conflicts as well as allocating different resources to multiple competing projects. Tao *et al.* [66] developed a Quantum Multi-Agent Evolutionary Algorithm (QMAEA) for multi-objective combinatorial optimization problems in large-scale service-oriented distributed simulation systems. Zheng and Wang [73] proposed a Multi-Agent Optimization Algorithm (MAOA) for solving the RCPSP. In the MAOA, the agents cooperate in a grouped environment. The agents evolve by means of social behavior, autonomous behavior, self-learning, and adjustment of environment. Martin *et al.* [54] developed a multi-agent-based distributed framework for tackling different problem domains. In their multi-agent system, each agent represents a different combination of meta-heuristic and local search algorithms. They evaluated their proposed framework on permutation flow-shop scheduling problem and capacitated vehicle routing problem. Han *et al.* [24] proposed a multi-agent system for offshore project scheduling. The multi-agent system was designed to facilitate the integration of offshore project scheduling. Fu *et al.* [20] addressed a two-agent stochastic flow shop deteriorating problem to minimize the make-span and total tardiness. Hosseinian and Baradaran [30] developed a multi-objective multi-agent optimization algorithm to optimize modularity and community score in the community detection problem. Their proposed algorithm uses the Weighted Sum Method (WSM) for finding the best and leader agents. In the previous studies on the multi-agent systems, the agents represent different concepts such as solutions, algorithms, activities, resources, etc. Table 3 shows the concepts represented by agents in previous studies. Table 3 also indicates that whether multi-criteria decision making techniques have been used in multi-agent systems or not. "MAS-WM[3]" implies that the proposed multi-agent system uses a Multi-Attribute Decision Making technique (MADM), while "MAS-WOM[4]" represents that the MADM techniques have not been utilized in the proposed MAS.

2.4. Significance of this research

Due to the previous studies reviewed in this section, there is a research gap for the multi-skill RCPSP with transfer times. Hence, in this paper, a bi-objective mathematical formulation is proposed for the multi-skill RCPSP with transfer times (MSRCPSP-TT). The objectives of the proposed model are minimization of make-span and total cost of project. Moreover, it can be inferred from Table 2 that evaluating the performance of a multi-agent optimization algorithm can be an interesting topic to investigate. Thus, we develop a multi-objective multi-agent optimization algorithm (MOMAOA) to solve the MSRCPSP-TT to evaluate its effectiveness. Besides, it can be concluded from Table 3 that the application of MADM techniques in multi-agent

[3]The MAS with multi-attribute decision making technique (MAS-WM).
[4]The MAS without multi-attribute decision making technique (MAS-WOM).

TABLE 3. Characteristics of multi-agent systems proposed in previous studies.

Authors	Objective		Agent					MADM technique		Field
	Single	Multi	Activity	Algorithm	Resource	Solution	Other	MAS-WM	MAS-WOM	
[5]	✓						✓		✓	Capacity allocation problem
[71]	✓		✓		✓		✓		✓	Project scheduling problem
[41]	✓		✓						✓	Project scheduling problem
[6]	✓						✓		✓	Train coupling and sharing problem
[45]	✓						✓		✓	Resource scheduling problem
[40]	✓		✓						✓	Project scheduling problem
[27]	✓						✓		✓	Project scheduling problem
[29]	✓			✓					✓	Project scheduling problem
[12]	✓						✓		✓	Project scheduling problem
[9]	✓		✓		✓		✓		✓	Project scheduling problem
[1]	✓				✓		✓		✓	Project scheduling problem
[66]		✓					✓		✓	Multi-objective optimization problems
[73]	✓					✓			✓	Project scheduling problem
[54]	✓					✓			✓	Flow-shop scheduling problem and Capacitated vehicle routing problem
[24]	✓				✓		✓		✓	Project scheduling problem
[20]		✓					✓		✓	Flow-shop scheduling problem
[30]		✓				✓		✓		Community detection problem
This research		✓				✓		✓		Project scheduling problem

systems deserves more attention. Hence, we used the TOPSIS method in various procedures of the MOMAOA to investigate the effect of an MADM technique in a multi-agent system.

3. Problem description and mathematical formulation

This paper studies the multi-skill resource-constrained project scheduling problem with transfer times (MSRCPSP-TT). The assumptions of the proposed problem are as follows:

- Let G $(J,\ E)$ be an activity-on-node (AON) network to depict the structure of the project. J is a set of interrelated and non-preemptive activities and E is a set of edges representing Finish-to-Start (FS) precedence relations among activities with zero-time lags. The precedence relations define that which activities should be completed before other activities could be started.
- Activities have known and predefined durations.
- Activities have merely one execution mode.
- To accomplish a project, a set of multi-skill and unrelated workers is required. Each worker is able to perform a subset of skills from the skill pool (*e.g.* electrician, machinist, analyst, tester, etc.).
- Workers have different use-costs.
- Expenditures of a project is bound to a limited budget.
- The workers are assigned to activities based on their required skills. For each skill of a worker, there is a certain familiarity level. The worker s is capable of performing the activity j, if and only if the worker s possesses the required skill and his/her familiarity level is not less than the standard level [69].
- Each worker is allowed to process at most one activity at a time.
- Workers have to be transferred between the execution sites of activities to perform required skills.
- Transfer times are known and deterministic.
- Transfer times of workers impose additional costs to the project.

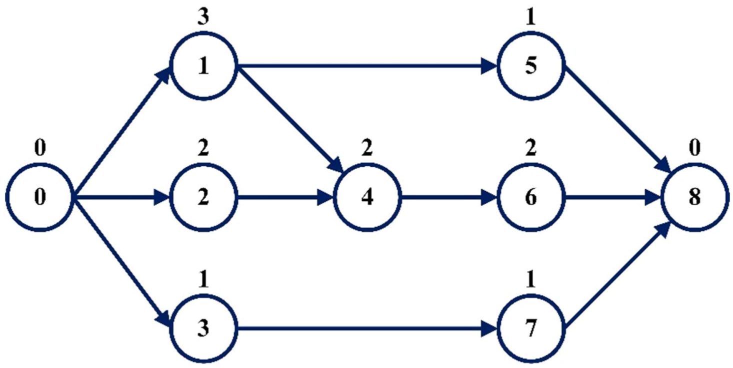

Figure 1. The AON network of the example.

Table 4. Required skills and standard levels of activities.

Activity	Skill	Standard level
1	3	2
2	2	1
3	2	3
4	1	2
5	3	3
6	2	3
7	1	3

– A planning time horizon of discrete time periods has been considered to schedule activities. The project includes a dummy start activity 0 and a dummy finish activity $N+1$, which mark the start time and finish time of the project, respectively. These dummy activities have no duration and they need no workers.

Consider a project comprising seven non-dummy activities to be performed by five workers. The project requires three skills. There are three familiarity levels for each skill. The activities "0" and "8" are dummy start and finish activities, respectively. The AON network of the project is illustrated in Figure 1. Each activity is represented as a node. The nodes are weighted with processing times.

Table 4 details the required skills by activities. Moreover, the standard level of each skill is reported in Table 4 as well. Table 5 shows the skills which can be performed by each worker. Besides, Table 5 provides the familiarity levels of workers.

Based on the information provided in Tables 4 and 5, a skill matrix $\mathrm{SK} = [sk_{js}]_{7\times 5}\,(j = 1,\ldots,7|s = 1,\ldots,5)$ is created that shows which workers can be assigned to each activity. Table 6 illustrates the skill matrix SK for the example. According to Table 6, the workers 1, 2, and 3 are eligible to perform activity "1". The workers 3, 4, and 5 can be assigned to activity "2". The worker 5 is the only eligible worker to execute activity "3". The workers 1, 4, and 5 can be allocated to activity "4", while the second worker is the only qualified human resource to perform the activity "5". The worker 5 can be assigned to activity "6" and the workers 4 and 5 can accomplish activity "7".

To transfer worker s from the operation site of activity j to the operation site of activity j', a transfer time denoted as $\tau_{jj's}$ is needed. The triangular inequality is satisfied for all transfer times ($\tau_{jj's} \le \tau_{jj''s} + \tau_{j''j's}$). For the project described above, the transfer time matrix ($\tau_{jj'}$) is as follows. Table 7 shows the resources assigned

TABLE 5. Skills and familiarity levels of workers.

Worker	Skill	Familiarity level
1	1	2
	3	1
2	3	3
3	2	1
	3	2
4	1	3
	2	2
	3	1
5	1	3
	2	3

TABLE 6. The skill matrix for the example.

Activities	Workers				
	1	2	3	4	5
1	✓	✓	✓	✗	✗
2	✗	✗	✓	✓	✓
3	✗	✗	✗	✗	✓
4	✓	✗	✗	✓	✓
5	✗	✓	✗	✗	✗
6	✗	✗	✗	✗	✓
7	✗	✗	✗	✓	✓

to activities based on the information given in Tables 4–6.

$$\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 2 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{9\times 9}.$$

Figure 2 illustrates a feasible schedule for the example. There are R_1, R_2, and R_3 available workers to perform the first, the second, and the third skills, respectively. The make-span of the project is equal to 8 time periods which has been obtained with respect to precedence relations and resource limitations. The arrows in Figure 2 indicate transfers of workers. The worker 4 has been assigned to activities "2" and "7". The worker 4 has to be transferred to the operation site of activity "7" after the completion of activity "2". Besides, the worker 5 has been assigned to activities "3" and "6". This worker needs to be transferred from the operation site of activity "3" to the execution site of activity "6". As shown in Figure 2, the transfer times have delayed the completion of project for two periods.

The objectives of the MSRCPSP-TT are minimization of make-span and total cost of project, simultaneously. In the following, Section 3.1 describes the notations of sets, parameters, and decision variables used in the

TABLE 7. The resources assigned to activities.

Activity	Assigned worker
1	3
2	4
3	5
4	1
5	2
6	5
7	4

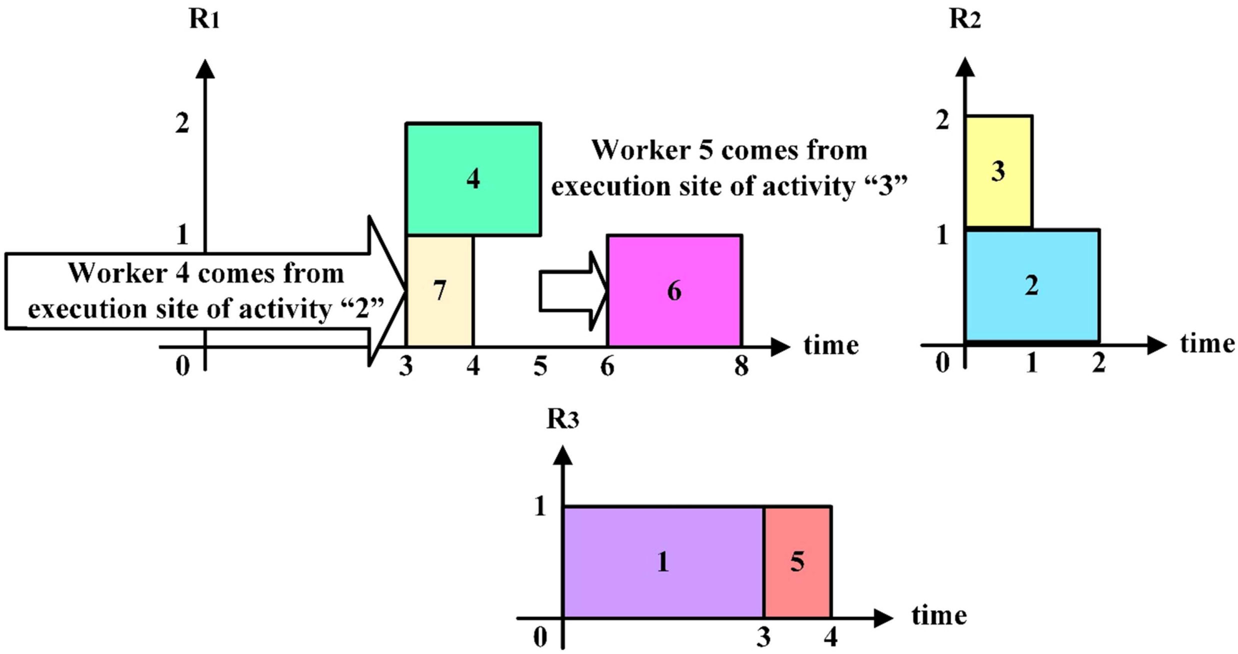

FIGURE 2. A feasible schedule for the MSRCPSP-TT.

proposed model. Section 3.2 presents the mathematical formulation, and Section 3.3 describes the presented model.

3.1. Notations

The following notations are defined to formulate the MSRCPSP-TT:

Sets

J	Set of activities ($j, j' = 0, \ldots, N+1$)
Γ	Set of required skills ($k = 1, \ldots, K$)
Π	Set of time periods ($t = 1, \ldots, T$)
E	Set of Finish-to-Start precedence relations
Λ	Set of multi-skill workers ($s = 1, \ldots, S$)
P_j	Set of predecessors of activity j

Parameters

d_j	Duration of activity j
c_s	The fixed unit salary of worker s
γ_{jk}	The required standard level of skill k for activity j
L_{sk}	The level that worker s masters skill k
$\tau_{jj's}$	The required time to transfer worker s from execution site of activity j to execution site of activity j'
β	The amount of budget considered for the whole project
ϑ_{sk}	Equals 1 if worker s has skill k, otherwise it equals 0

Variables

C_{js}	The cost of performing activity j by worker s
θ_j	The processing time required by activity j considering transfer time
FT_j	Finish time of activity j
Z_1	Make-span of the project
Z_2	Total cost of the project
X_{jt}	Equals 1 if activity j starts at the beginning of period t, otherwise it equals 0
$\lambda_{jj's}$	Equals 1 if worker s is transferred from execution site of activity j to execution site of activity j', otherwise it equals 0
Y_{js}	Equals 1 if worker s is assigned to activity j, otherwise it equals 0
ω_{jk}	Equals 1 if activity j requires skill k, otherwise it equals 0
η_{jst}	Equals 1 if worker s is performing activity j in period t, otherwise it equals 0

3.2. Mathematical formulation

$$\text{Min } Z_1 = \sum_{t=1}^{T} t \cdot X_{(N+1)t} \tag{3.1}$$

$$\text{Min } Z_2 = \sum_{j=0}^{N+1} \sum_{s=1}^{S} Y_{js} \cdot C_{js} \tag{3.2}$$

Subject to:

$$\sum_{t=1}^{T} X_{jt} = 1; \qquad \forall j \in J \tag{3.3}$$

$$\theta_j = d_j + \sum_{j'=0}^{N+1} \sum_{s=1}^{S} \tau_{j'js} \cdot \lambda_{j'js} \cdot Y_{js}; \qquad \forall j \in J \tag{3.4}$$

$$\mathrm{FT}_j \leq \mathrm{FT}_{j'} - \theta_{j'}; \qquad \forall j, j' \in J, j \in P_{j'} \tag{3.5}$$

$$\sum_{s=1}^{S} Y_{js} \cdot L_{sk} \geq \omega_{jk} \cdot \gamma_{jk}; \qquad \forall j \in J, \forall k \in \Gamma \tag{3.6}$$

$$\vartheta_{sk} \geq \omega_{jk}; \qquad \forall j \in J, \forall s \in \Lambda \tag{3.7}$$

$$\sum_{j=0}^{N+1} \eta_{jst} \leq 1; \qquad \forall s \in \Lambda, \forall t \in \Pi \tag{3.8}$$

$$C_{js} = c_s \cdot \theta_j; \qquad \forall j \in J, \forall s \in \Lambda \tag{3.9}$$

$$\sum_{t=1}^{T} t \cdot X_{j't} - \sum_{t=1}^{T} (t + d_j) \cdot X_{jt} - (T + \tau_{jj's}) \cdot \lambda_{jj's} \geq -T; \qquad \forall j, j' \in J, \forall s \in \Lambda \tag{3.10}$$

$$\sum_{j=0}^{N+1}\sum_{s=1}^{S} Y_{js}\cdot C_{js} \leq \beta \tag{3.11}$$

$$C_{js}, \theta_j, \mathrm{FT}_j, Z_1, Z_2 \geq 0 \tag{3.12}$$

$$X_{jt}, \lambda_{jj's}, Y_{js}, \omega_{jk}, \eta_{jst} \in \{0,1\}. \tag{3.13}$$

3.3. Model description

The objective functions (3.1) and (3.2) are to minimize the make-span and total cost of project, respectively. Constraint (3.3) secures that each activity starts exactly once. In a project, activities cannot be started more than once when preemption is not allowed. Therefore, Constraint (3.3) is required for the formulation. If a worker is supposed to perform activity j at site "B" when he/she has just completed another task in site "A", he/she should be transferred to site "B" in order to carry out the aforementioned activity. Therefore, this transferring time must be considered in the overall required duration of activity j. Equation (3.4) calculates the processing time required by activity j considering resource transfer time. When two activities are bound together regarding precedence relations, the successor must be completed after all its predecessors. In other words, the finish time of a successor cannot be larger than the finish times of its predecessors. Constraint (3.5) secures the precedence relations between activities. Based on the assumptions of the model, for performing each skill of an activity, there is a standard level. Hence, even though a worker may have the required skill of an activity, he/she may not be efficient enough to perform that specific skill. The eligibility of a worker is determined by his/her familiarity level. Therefore, the familiarity level of the workers assigned to skill k of activity j must be equal or more than the standard level requested by that specific skill of activity j. Constraints (3.6) and (3.7) guarantee that each activity can only be performed by eligible workers. Workers cannot be present at two different locations at the same time; therefore, Constraint (3.8) ensures that each worker can only perform one activity in each period. Salary of workers is different due to their familiarity levels. Thus, the cost of an activity depend on the workers assigned to it. Equation (3.9) computes the cost of performing activity j by worker s. Constraint (3.10) secures that transfer time of worker s is taken into consideration. The budget of projects are limited in real-world scenarios, therefore this limitation should be considered in the formulation. Constraint (3.11) secures that total cost of project cannot exceed the amount of budget considered for the whole project. Constraints (3.12) and (3.13) defines the feasible scope of decision variables.

4. Solution approach

4.1. Multi-agent system (MAS)

Agent is known as a notion in artificial intelligence [73]. Each agent can be interpreted as a computer system existing in a particular environment. Agents receive information from the environment by means of sensors. They can take appropriate actions to independently comprehend the target of the system without interventions from humans or other agents. For each agent, there is a set of possible action, namely social behaviors, pro-activeness, and responsiveness [70, 73]. The agents analyze the information received from the environment and take immediate actions to influence the environment or to adapt to its changes. Social behavior enables the agents to interact with other ones or to interact with external entities. In a multi-agent system (MAS), there is a group of independent agents that interact with one another and perform their tasks in a specific environment to accomplish predefined targets [2, 37, 73]. For each MAS, there are three main factors: (1) a set of available agents denoted as $A = \{A_1, A_2, \ldots, A_n\}$, (2) the environment where the agents perform their tasks and interact with each other, and (3) a set of rules that control the interactions between agents and environment [48, 73]. The agents can be arranged in different organizations, for more details on various organizations of agents please visit the reference [2]. This study considers group organization for agents which has been illustrated in Figure 3.

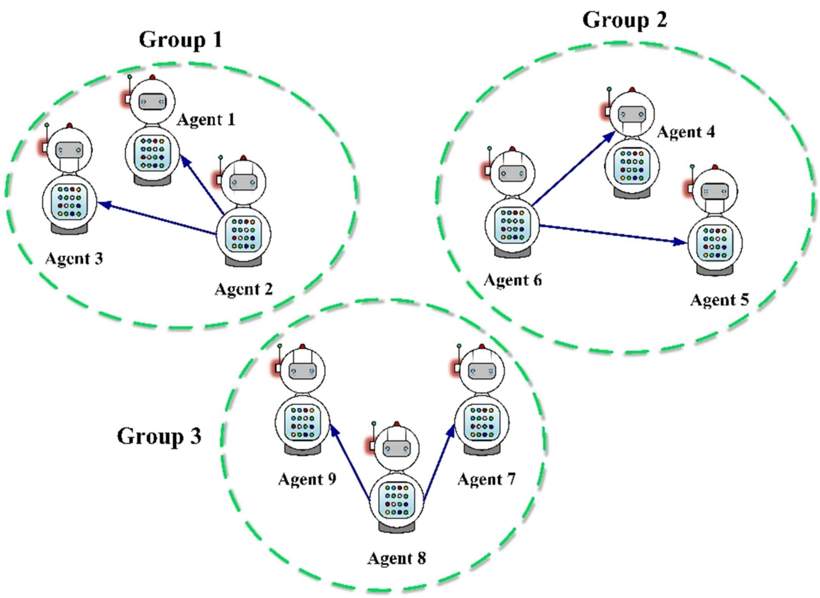

FIGURE 3. Group organization of agents [2].

4.2. Multi-agent optimization algorithm (MAOA)

For each multi-agent system, there are three major features: (1) environment, (2) behaviors of agents, and (3) interactions between agents [73]. These features are elaborated as follows:

4.2.1. Adjustment of environment

In the multi-agent system used in this study, each agent is represented as a solution. The environment is structured by the agents and their relations. This study utilized the grouped structure introduced by Zheng and Wang [73] that consists of $G\,(g = 1, \ldots, G)$ groups. Each group is constituted by NA_g agents, where NA_g denotes the number of agents in gth group. The agent that has the best fitness is considered as the "leader". Leader agents of existing groups are compared with each other. The group that has the best leader agent among all leader agents is known as the elite group. The second best agent in each group is known as the "active" agent. Figure 4 illustrates a leader-group organization.

The agents can explore the solution space accurately through adjusting the environment. In a multi-agent optimization algorithm (MAOA), all agents are re-grouped so as to update the environment. In this respect, the active agent of each group is substituted with the worst agent of the elite group. This adjustment will share the information among groups and it helps to improve the exploring procedure [73].

4.2.2. Social, autonomous, and self-learning behaviors of agents:

The MAOA has two sorts of social behaviors, namely local social behavior and global social behavior which are explained as follows [73]:

- Local social behavior, which indicates the collaborative interaction within a group. The interaction between the leader agent of a group with other agents in that specific group is defined as the local social behavior. This type of social behavior helps to exploit the neighborhoods of existing agents in a group. Local social behavior of agents is shown in Figure 5a.
- Global social behavior, which is the collaborative interaction in the entire environment. For this type of social behavior, the leader agent of the elite group works with leader agents of all groups. This type of

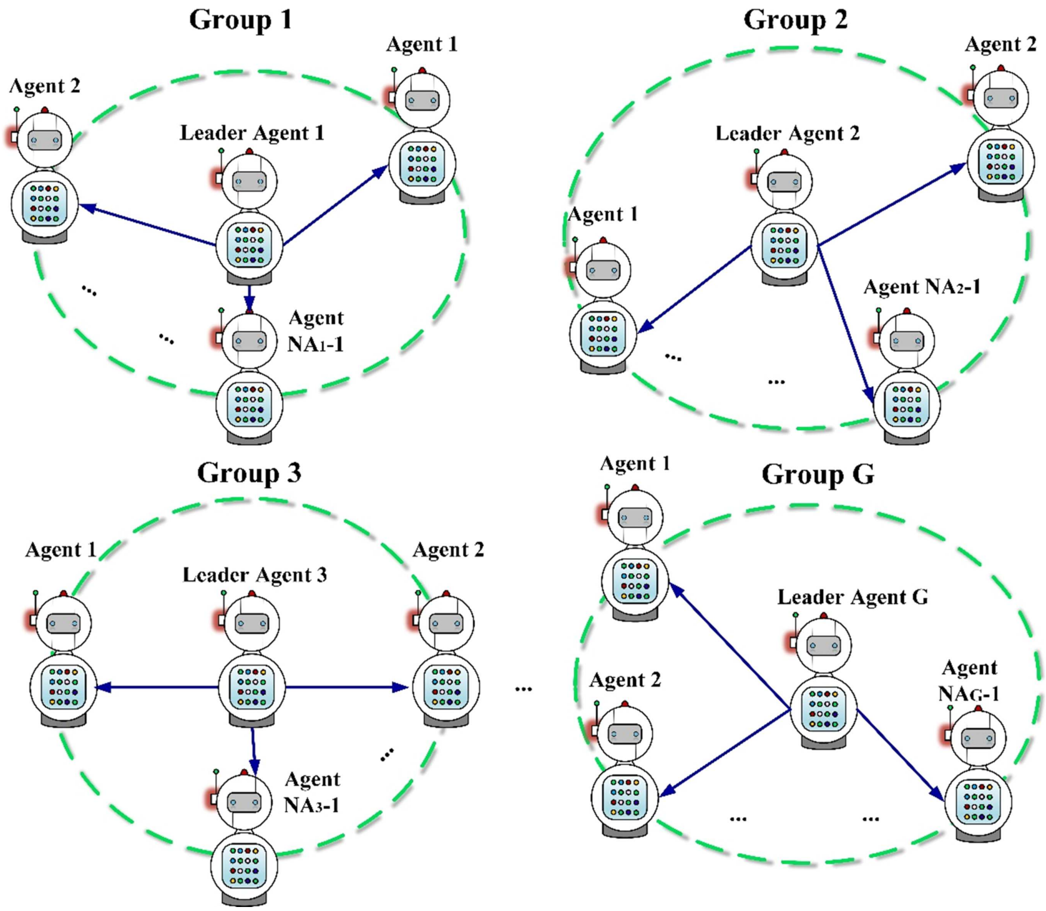

Figure 4. Leader-group organization of agents [73].

cooperation leads to profound exploration of the entire solution space. Figure 5b shows the global social behavior of agents.

For each agent, there is another behavior called the autonomy. Based on this behavior, each agent can act independently without external interference. Due to this behavior, each agent exploits its neighborhood in a randomly manner to find better solutions [73]. Self-learning is another behavior considered for each agent in a MAS. Since agents receive information throughout the solving process, they can improve themselves *via* learning from the obtained knowledge [75]. The structure of the classical MAOA is depicted in Figure 6.

4.3. Multi-Objective Multi-Agent Optimization Algorithm (MOMAOA)

Based on the social, autonomous, and self-learning behaviors of a multi-agent system and due to the multi-objective optimization problem tackled in this study, we propose a multi-objective multi-agent optimization algorithm (MOMAOA). In the MOMAOA, the environment is initially formed by dispersing agents into multiple groups. The agents are evolved *via* social, autonomous, and self-learning behaviors. The social behavior is considered as the global exploration, while the autonomous and self-learning behaviors are considered as local exploitation. To adjust the environment, the agents are transferred among groups to deepen the exploration process. Features of the MOMAOA for solving the MSRCPSP-TT are described in the following sections.

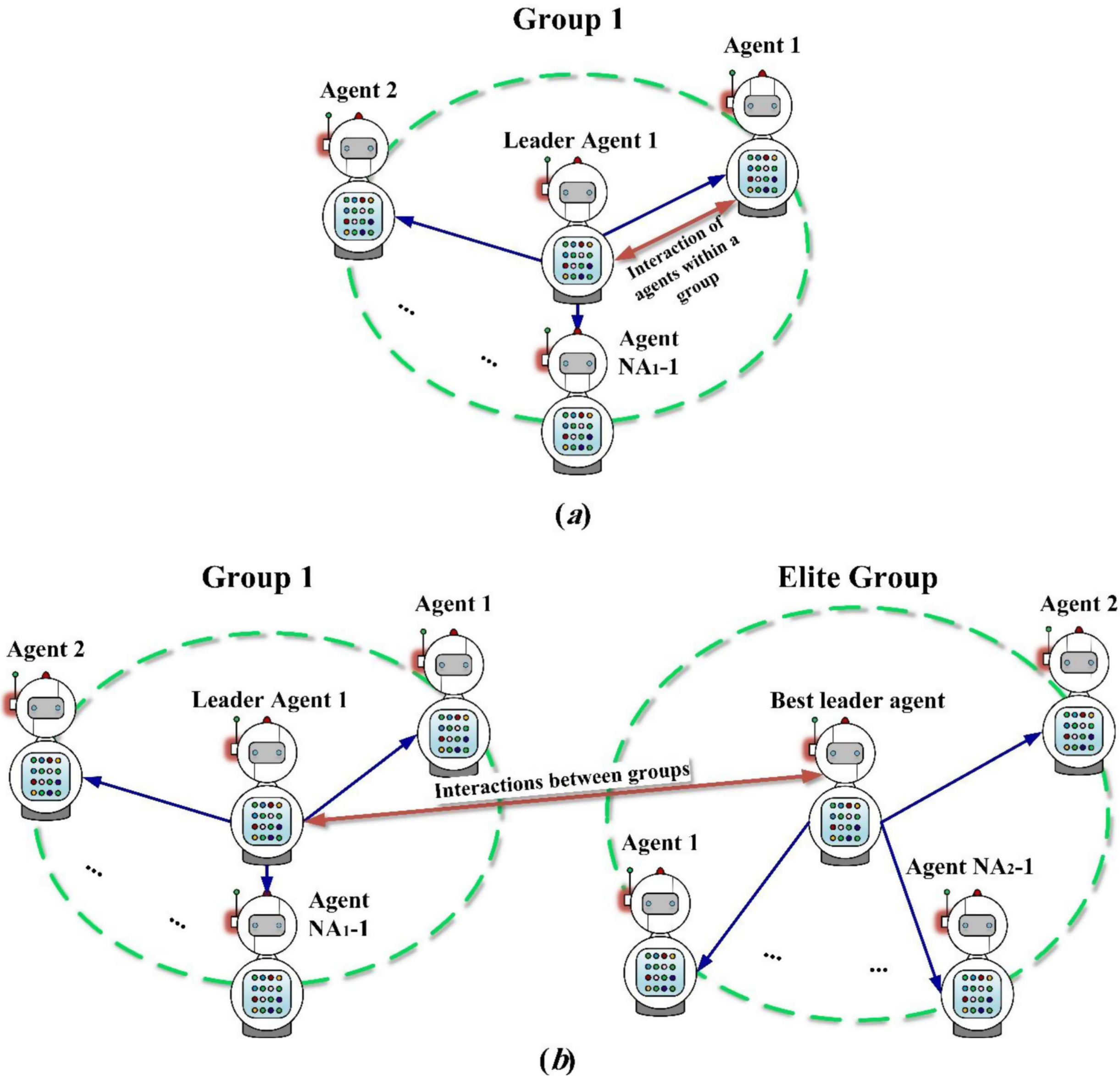

FIGURE 5. (a) Local social behavior of agents. (b) Global social behavior of agents [73].

4.3.1. Solution representation and decoding scheme

In this paper, each agent denotes a solution of the MSRCPSP-TT. Each solution is represented as a $(2 \times N)$ matrix as shown in Figure 7, where N is the number of project activities. The first row of each solution is a precedence-feasible activity list. Each activity j_a $(a = 1, 2, \ldots, N)$ should be positioned on the list after all its predecessors [25]. The second row is a resource list which indicates the resources assigned to each activity. π_a $(a = 1, 2, \ldots, N)$ indicates the worker assigned to the activity j_a.

Having produced the agents (solutions), they are randomly dispersed into G groups. Each group comprises GS number of agents (GS denotes the group size). Hence, the population size is equal to $(G \times \text{GS})$. The decoding

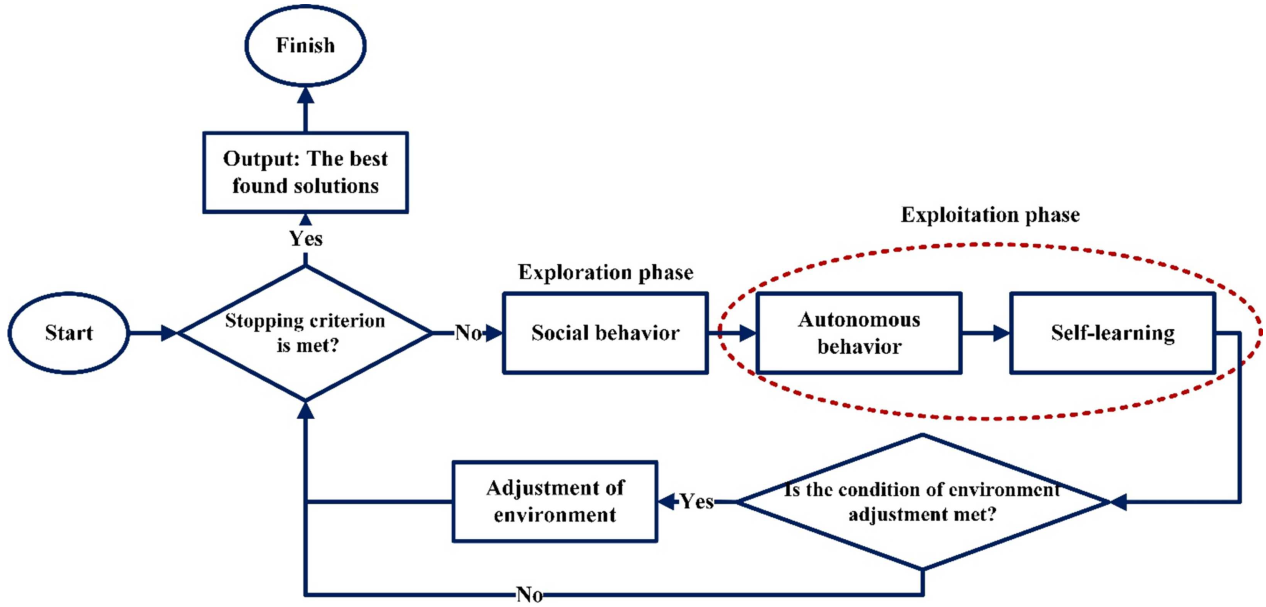

Figure 6. Flowchart of the basic MAOA [73].

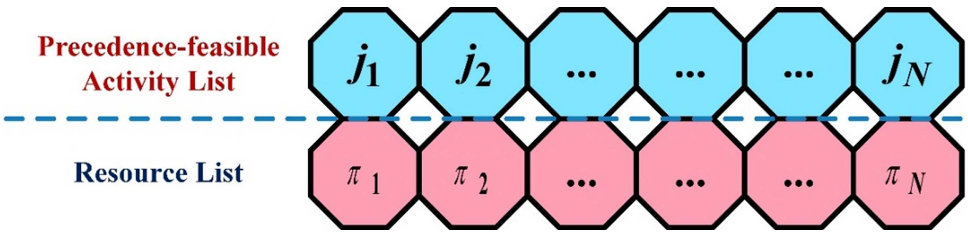

Figure 7. Solution representation.

procedure determines the start times of activities according to the sequence of the activity list and the resource assignment plan. In this study, we apply the serial schedule generation scheme (S-SGS) to construct schedules for the MSRCPSP-TT. The S-SGS is an iterative procedure that consecutively adds an activity to a schedule until a feasible schedule is achieved. In each iteration, the first un-scheduled activity on the precedence-feasible activity list is selected to determine its earliest possible start time. This process continues until no un-scheduled activity is left [38].

4.3.2. Procedure of finding the leader agent in each group

To find the leader agent in each group LA_g $(g = 1, \ldots, G)$, we firstly utilize the non-dominated sorting method in the NSGA-II proposed by Deb *et al.* [19] to determine the non-dominated agents (solutions). This mechanism can be embedded in most of multi-objective evolutionary algorithms to approximate the Pareto front. If there is one single non-dominated agent among all agents existing in a group, this agent is considered as the leader agent of the group. However, if there are multiple non-dominated agents in a group, we use the TOPSIS method which is a multi-attribute decision making technique to rank these agents. The concept of the TOPSIS method is that the selected alternative has the least distance from the positive ideal solution, while it should be away from the negative ideal solution [8]. To use the TOPSIS method, a decision matrix is created. The rows and

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)

Notations:

DM	The decision matrix, where rows and columns are agents and criteria, respectively.
Q	The number of agents (rows of decision matrix, alternatives, or candidate solutions)
O	The number of criteria (columns of decision matrix or objective functions)
$q_{ii'}$	The element in intersection of i^{th} row and i'^{th} column ($i = 1, \ldots, Q \mid i' = 1, \ldots, O$) in the DM
NM	The normalized decision matrix
$\upsilon_{ii'}$	The element in intersection of i^{th} row and i'^{th} column in the NM
WNM	The weighted normalized matrix
$u_{ii'}$	The element in intersection of i^{th} row and i'^{th} column ($i = 1, \ldots, Q \mid i' = 1, \ldots, O$) in WNM
$w_{i'}$	The weight of i'^{th} criterion ($i' = 1, \ldots, O$)
ψ_b	The best alternative (agent)
ψ_w	The worst alternative (agent)
D_i^+	The distance between i^{th} agent and the best agent
D_i^-	The distance between i^{th} agent and the worst agent

Step 1. Normalize the decision matrix (DM). The normalized matrix (NM) is obtained using the Eq. (4.1)

$$\upsilon_{ii'} = \frac{q_{ii'}}{\sqrt{\sum_{i'=1}^{O} q_{ii'}^2}}, (i = 1, \ldots, Q) \tag{4.1}$$

Step 2. Compute the weighted normalized decision matrix (WNM) using the Eq. (4.2)

$$u_{ii'} = \upsilon_{ii'} . w_{i'}, \quad (i = 1, \ldots, Q), \quad (i' = 1, \ldots, O) \tag{4.2}$$

Step 3. Find the best alternative agent (ψ_b) and the worst alternative agent (ψ_w) by the following equations:

$$\psi_b = \left\{ \left\langle \min\left(u_{ii'} \mid i = 1, \ldots, Q\right) \mid i' \in CR^- \right\rangle, \left\langle \max\left(u_{ii'} \mid i = 1, \ldots, Q\right) \mid i' \in CR^+ \right\rangle \right\} \equiv \left\{ u_{bi'} \mid i' = 1, \ldots, O \right\} \tag{4.3}$$

$$\psi_w = \left\{ \left\langle \max\left(u_{ii'} \mid i = 1, \ldots, Q\right) \mid i' \in CR^- \right\rangle, \left\langle \min\left(u_{ii'} \mid i = 1, \ldots, Q\right) \mid i' \in CR^+ \right\rangle \right\} \equiv \left\{ u_{wi'} \mid i' = 1, \ldots, O \right\} \tag{4.4}$$

Where, $CR^+ = \{i' = 1, \ldots, O | i'\}$ denotes the positive criteria, while $CR^- = \{i' = 1, \ldots, O | i'\}$ represents negative criteria.

Step 4. Calculate the distance between the best agent and each agent (D_i^+) and the distance between the worst agent and each agent (D_i^-):

$$D_i^+ = \sqrt{\sum_{i'=1}^{O} \left(u_{ii'} - u_{bi'}\right)^2}, (i = 1, \ldots, Q) \tag{4.5}$$

$$D_i^- = \sqrt{\sum_{i'=1}^{O} \left(u_{ii'} - u_{wi'}\right)^2}, (i = 1, \ldots, Q) \tag{4.6}$$

Step 5. Calculate the relative closeness to the best agent for each alternative (CL_i). The higher the value of CL_i, the better the rank of i^{th} agent:

$$Cl_i = \frac{D_i^-}{D_i^- + D_i^+}, \quad \left(0 \le CL_i \le 1\right), \quad \left(i = 1, \ldots, Q\right) \tag{4.7}$$

Output: Ranking of agents.

Figure 8. Procedure of the TOPSIS method used to find leader agents.

columns of this matrix represent the non-dominated agents and criteria, respectively. These criteria include the make-span and total cost of the project. Both make-span and total cost of project are negative criteria. The criteria are equally important. The procedure of the TOPSIS method is illustrated in Figure 8. The procedure of finding the leader agent of each group is depicted in Figure 9.

4.3.3. Procedure of finding the global best leader agent

To find the global best leader agent, the non-dominated sorting method is hired once again to determine the non-dominated agents among leader agents of all groups. If there is one single non-dominated agent among all leader agents, this agent is chosen as the global best leader agent. Otherwise, the TOPSIS method described in Section 4.3.2 is used to rank the leader agents so as to find the best one. Figure 10 shows the procedure of finding the global best leader agent.

Procedure of finding the leader agent of the group	
Notations:	
NOS	The number of non-dominated solutions (agents)
DM	The decision matrix (rows and columns represent the non-dominated agents and objective functions, respectively)

```
Use the non-dominated sorting method to determine the non-dominated solutions (agents);
    IF NOS > 1 do
        Create a decision matrix (DM);
        Apply the TOPSIS method to rank the agents;
        Select the best ranked agent as the leader agent of the group;
    ELSE
        Select the only non-dominated agent as the leader agent of the group;
    ENDIF
Output: Leader agent of the group
```

Figure 9. Procedure of finding leader agents.

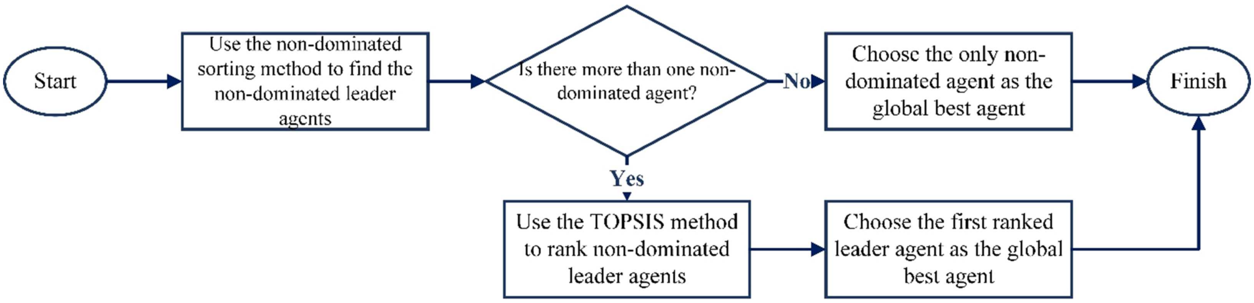

Figure 10. Procedure of finding the global best leader agent.

4.3.4. Social behavior in the MOMAOA

A crossover operator has been applied for the MOAOA that the offspring agents can inherit characteristics of both parents. This crossover operator is used as the global social behavior to perform the interaction between the global best leader agent and the leader agent of each group. The best offspring agent will take the place of corresponding leader agent if any of the following conditions is met:

(1) The best offspring agent dominates the corresponding leader agent.
(2) A real random number RND (RND $\in (0,1)$) is larger than a predefined probability of replacement (ρ).

Since the solution representation used in this paper consists of two parts, the crossover operator produces offspring agents in two phases. The first phase is dedicated to generating a feasible activity list, while the second phase is considered to generate a feasible resource list. To generate feasible activity lists, the Magnet-Based Crossover Operator (MBCO) introduced by Zamani [72] has been hired. To determine the workers assigned to activities on offspring agents, a simple procedure is used. In this procedure, an integer random number (Rand) is generated on the interval $[1,2]$ for each activity. If Rand is equal to 1 for activity j, the worker assigned to activity j on the global best agent will be allocated to this activity on the offspring agent. On the other hand, if Rand is equal to 2 for activity j, the worker assigned to activity j on the leader agent will be allocated to this activity on the offspring agent. Figure 11 shows the procedure of generating resource lists for the offspring agents.

For the local social behavior, a procedure is used to enhance the quality of agents with the help of their corresponding leader agent. Since transfer times of resources increase the make-span and total cost of project, the procedure of assigning resources to activities can be improved for each agent so as to minimize both objective functions. In this respect, a random binary string $(\text{RBS})_{1\times N}$ is generated to let resource lists of agents inherit

Procedure of generating resource lists for offspring agents

Notations:

NOF	Number of offspring agents ($i = 1, \ldots, NOF$)
N	Number of activities ($j = 1, \ldots, N$)
$Rand$	An integer random number on the interval [1,2]
WRL_{ij}	The worker assigned to activity j on resource list of offspring i
WGB_j	The worker assigned to activity j on the global best agent
WLA_j	The worker assigned to activity j on the leader agent

```
FOR i = 1 to NOF do
    FOR j = 1 to N do
        Generate Rand on the interval [1,2];
        IF Rand = 1 do
            WRL_ij ← WGB_j;
        ELSE
            WRL_ij ← WLA_j;
        ENDIF
    ENDFOR
ENDFOR
```

Output: Resource lists of offspring agents

FIGURE 11. Generating resource lists of offspring agents.

Procedure of improving resource lists of agents

Notations:

G	The number of groups ($g = 1, \ldots, G$)
NA_g	The number of agents in group g ($i = 1, \ldots, NA_g$)
N	The number of activities ($j = 1, \ldots, N$)
WAA_{ij}	The worker assigned to activity j on the resource list of agent i
WLA_j	The worker assigned to activity j on the leader agent
$WRLA_{ij}$	The worker assigned to activity j on the newly generated resource list of agent i
RBS_j	The value of j^{th} position on random binary string

```
Generate a random binary string (RBS)_{1×N} ;
FOR g = 1 to G do
    FOR i = 1 to NA_g do
        FOR j = 1 to N do
            IF RBS_j = 1 do
                WRLAj ← WLAj
            ELSE
                WRLAj ← WAAj
            ENDIF
        ENDFOR
    ENDFOR
ENDFOR
```

FIGURE 12. Improving resource lists of agents.

from the resource list of the leader agent. If RBS_j is equal to 1, the worker assigned to activity j on the leader agent will be assigned to this activity on the newly generated resource list. If $\text{RBS}_j$is equal to 0, the worker assigned to activity j will not change on the newly generated resource list. The proposed operator used as the local social behavior is shown in Figure 12. The whole procedure of social behavior in the MOMAOA including global and local behaviors is depicted in Figure 13.

4.3.5. Autonomous behavior in the MOMAOA

In this study, we utilize the Permutation-Based Swap (PBS) operator proposed by Chen *et al.* [10] for autonomous behavior of each agent. The PBS operator randomly chooses two adjacent activities with no precedence

Procedure of social behavior

Notations:

LA_g	The leader agent in group g ($g = 1, \ldots, G$)
GBA	The global best agent which belongs to group l
NA_g	The number of agents in group g
BOF	The best generated offspring agent
RND	A real random number on the interval (0,1)
ρ	The probability of replacing LA_g with BOF

Global social behavior

FOR $g = 1$ **to** G ($g \neq l$) **do**
  Apply the crossover operator between LA_g and GBA to generate new offspring;
  Use the non-dominated sorting method to determine the non-dominated agents;
  Employ the TOPSIS method to select the best offspring agent (BOF);
  IF BOF dominates LA_g **do**
    Change the place of LA_g with BOF;
  ELSE produce a random number (RND)
    IF $RND \geq \rho$
      Change the place of LA_g with BOF;
    ENDIF
  ENDIF
ENDFOR

Local social behavior

FOR $g = 1$ **to** G **do**
  FOR agent A_i ($i = 1$ to NA_g and $A_i \neq LA_g$) **do**
    Apply the operator proposed for the local social behavior between A_i and LA_g;
    IF the generated agent dominates A_i **do**
      Change the place of A_i with the produced agent;
    ENDIF
  ENDFOR
ENDFOR

FIGURE 13. Procedure of social behavior.

relations. A new activity list is generated by swapping these two activities. Given that these two activities have no precedence relations, the newly produced activity list is feasible [73]. Since each agent includes a resource list, the PBS operator needs to be developed to generate feasible resource lists as well. For this purpose, the PBS operator swaps the assigned workers of the selected activities in order to generate a feasible resource list. Considering the project depicted in Figure 1, the procedure of the PBS operator is illustrated in Figure 14.

4.3.6. Self-learning in the MOMAOA

Li and Willis [46] proposed a Forward-Backward Improvement (FBI) procedure to reduce the project completion time. The FBI procedure adjusts a solution, iteratively. In each iteration, the backward and forward scheduling method is used to minimize the make-span. Similar to the multi-agent system developed in [73], the MOMAOA employs the FBI method as the self-learning behavior of the best leader agent which enables the algorithm to deepen its exploitation phase.

4.3.7. Adjustment of environment in the MOMAOA

The adjustment of environment is required to share information among existing groups. The MOMAOA adjusts its environment every fifteen iterations. Suppose that the global best agent belongs to group l. For each group g ($g \neq l$), the TOPSIS method is employed to determine the active agent. If the active agent of group g

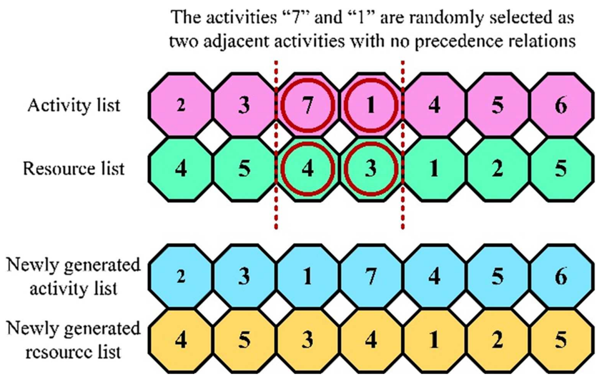

Figure 14. Using the PBS operator to generate a new feasible agent.

Procedure of adjusting the environment

Notations:

G	The number of groups ($g = 1, \ldots, G$)
WA	The worst agent of group l
AA_g	The active agent of group g (The second ranked agent of group g)

FOR $g = 1$ **to** $G\,(g \neq l)$ **do**
  Use the TOPSIS method to rank the agents of group g and specify the active agent (AA_g);
  Determine the worst agent of group l (WA);
  IF AA_g dominates WA **do**
    AA_g is transferred to group l and WA is transferred to group g;
  ENDIF
ENDFOR

Figure 15. Procedure of adjusting the environment.

(AA_g) dominates the worst agent of group l (WA), the AA_g moves to group l and the WA takes the position of AA_g in group g. Figure 15 illustrates the procedure of adjusting the environment.

4.3.8. Elitism in the MOMAOA

For the MOMAOA, there is an archive of non-dominated agents. In each iteration, the non-dominated offspring agents generated by social behavior, autonomous behavior, self-learning, and adjustment of environment are merged. Each offspring agent is compared to the agents existing in the archive. If an offspring agent succeeds to dominate any of the agents existing in the archive, it will take the place of the dominated agent. The maximum number of iterations (MaxIt) has been considered as the stopping criterion for the MOMAOA. The structure of the MOMAOA is depicted in Figure 16.

5. Computational study

In this section, we evaluate the performance of the MOMAOA comparing to three state-of-the-art multi-objective evolutionary algorithms, *i.e.* NSGA-II, PESA-II, and MOPSO. The algorithms are coded in the Matlab R2017b software. The codes are run on a PC with Intel Core 2 Quad processor Q8200 (4M Cache, 2.33 GHz, 1333 MHz FSB) and 4GB memory. The results obtained by the algorithms are described by project duration (hours) and project cost (currency unit).

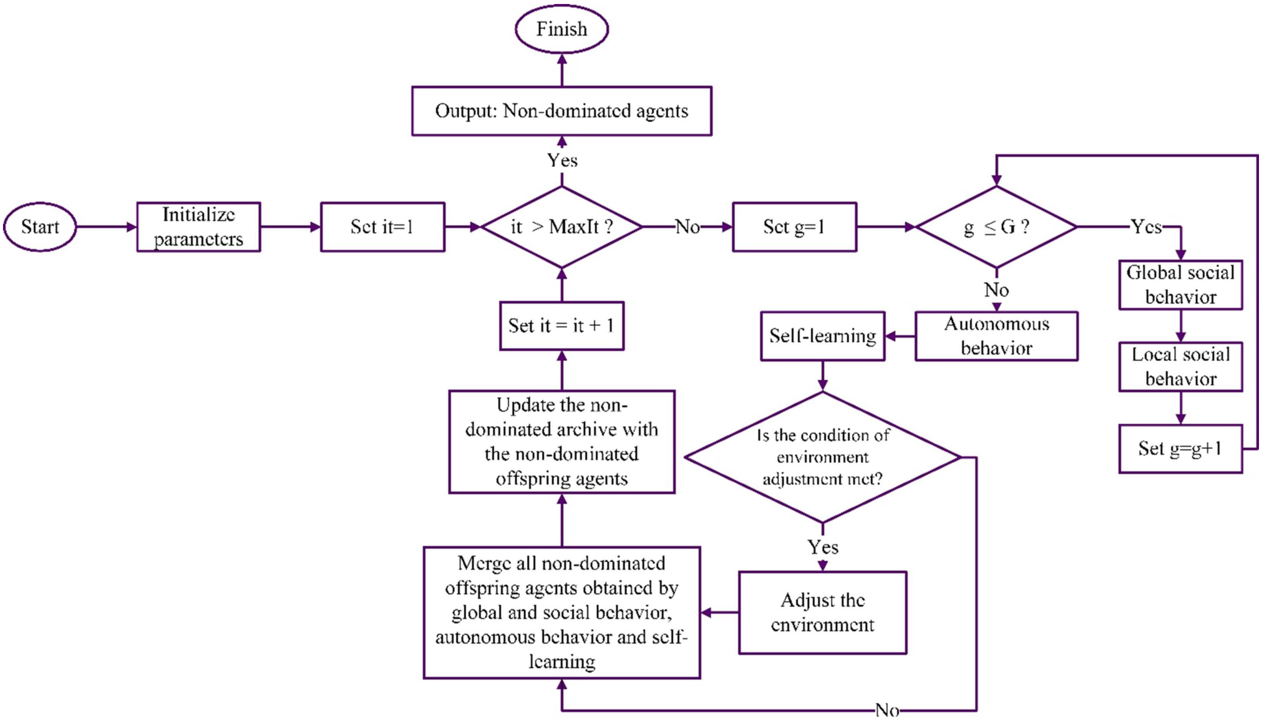

Figure 16. Flowchart of the MOMAOA.

5.1. Test problems

We use the iMOPSE dataset proposed by Myszkowski *et al.* [59], which have been generated based on real-world projects. The iMOPSE dataset consists of 36 project instances which has been produced based on the most general features of projects. These features include the number of activities (N), the number of available workforces (S), the number of precedence relations between activities (NPR), and the number of required skills (K). Table 8 summarizes the features of the iMOPSE dataset. As shown in Table 8, there are two groups of test problems that consist of 100 and 200 activities. Test instances 1–18 are considered as small size problems while test instances 19–36 are considered as large size problems In the iMOPSE dataset, each worker masters six different skills. The scheduling complexity is different for each project due to various features. For detailed description of the iMOPSE dataset, see [59].

5.2. Performance measures

Since the objectives of the proposed model conflict with each other, it is challenging to evaluate a multi-objective evolutionary algorithm (MOEA). For multi-objective optimization problems, it is required to provide multiple but evenly distributed solutions to form a Pareto front. These solutions enable the decision maker to choose from different alternatives [55]. We use five well-known multi-objective metrics to evaluate the performances of the algorithms. These metrics are as follows:

– **Mean ideal distance (MID)**

The metric measures the closeness between the solutions of the approximation front and the ideal point [77]:

$$\mathrm{MID} = \frac{\sum_{i=1}^{\mathrm{NOS}} \sqrt{\left(\mathrm{OFV}_1^i - \mathrm{OFV}_1^*\right)^2 + \left(\mathrm{OFV}_2^i - \mathrm{OFV}_2^*\right)^2}}{\mathrm{NOS}} \tag{5.1}$$

TABLE 8. Summary of the iMOPSE dataset [59].

Problem No.	Dataset instance	N	S	NPR	K
1	100_20_23_9_D1	100	20	23	9
2	100_20_22_15	100	20	22	15
3	100_20_47_9	100	20	47	9
4	100_20_46_15	100	20	46	15
5	100_20_65_9	100	20	65	9
6	100_20_65_15	100	20	65	15
7	100_10_27_9_D2	100	10	27	9
8	100_10_26_15	100	10	26	15
9	100_10_47_9	100	10	47	9
10	100_10_48_15	100	10	48	15
11	100_10_64_9	100	10	64	9
12	100_10_65_15	100	10	65	15
13	100_5_20_9_D3	100	5	20	9
14	100_5_20_15	100	5	22	15
15	100_5_48_9	100	5	48	9
16	100_5_48_15	100	5	46	15
17	100_5_64_9	100	5	64	9
18	100_5_64_15	100	5	64	15
19	200_40_45_9	200	40	45	9
20	200_40_45_15	200	40	45	15
21	200_40_90_9	200	40	90	9
22	200_40_91_9	200	40	91	15
23	200_40_130_9_D4	200	40	130	9
24	200_40_144_15	200	40	133	15
25	200_20_55_9	200	20	55	9
26	200_20_54_15	200	20	54	15
27	200_20_97_9	200	20	97	9
28	200_20_97_15	200	20	97	15
29	200_20_150_9_D5	200	20	150	9
30	200_20_145_15	200	20	145	15
31	200_10_50_9	200	10	50	9
32	200_10_50_15	200	10	50	15
33	200_10_84_9	200	10	84	9
34	200_10_85_15	200	10	85	15
35	200_10_135_9_D6	200	10	135	9
36	200_10_128_15	200	10	128	15

where, NOS is the number of non-dominated solutions on the approximation front. For the ith non-dominated individual existing on the approximation front, OFV_1^i and OFV_2^i are the make-span and total cost values, respectively. OFV_1^* and OFV_2^* denote the ideal points regarding each objective. In comparison of multiple algorithms, the algorithm with the least MID value has the best performance.

– **Spacing metric (SM)**

This metric measures the distribution of non-dominated solutions throughout the approximation front. This metric can be computed using the equation (5.2) [64]:

$$\mathrm{SM} = \sqrt{\frac{1}{\mathrm{NOS}-1}\sum_{i=1}^{\mathrm{NOS}}\left(\mathrm{dist}_i - \overline{\mathrm{dist}}\right)^2} \tag{5.2}$$

where, dist_i is the Euclidean distance in the phenotype space between consecutive solutions on the approximation front. dist_i is computed using the following formula:

$$\mathrm{dist}_i = \min_{i'} \left\{ \left|\mathrm{OFV}_1^i - \mathrm{OFV}_1^{i'}\right| + \left|\mathrm{OFV}_2^i - \mathrm{OFV}_2^{i'}\right| \right\}; \qquad i, i' = 1, 2, \ldots, \mathrm{NOS} \tag{5.3}$$

$\overline{\mathrm{dist}}$ is the average of all dist_i, computed as $\overline{\mathrm{dist}} = \sum_{i=1}^{\mathrm{NOS}} \mathrm{dist}_i/\mathrm{NOS}$. The closer the values of SM to zero, the more uniformly the distribution of solutions.

– **Diversification metric (DM)**

This metric is used to measure the extension of the Pareto front. Higher extension of a Pareto front indicates better diversity of results [77]:

$$\mathrm{DM} = \sqrt{\left(\max_{i=1:\mathrm{NOS}} \mathrm{OFV}_1^i - \min_{i=1:\mathrm{NOS}} \mathrm{OFV}_1^i\right)^2 + \left(\max_{i=1:\mathrm{NOS}} \mathrm{OFV}_2^i - \min_{i=1:\mathrm{NOS}} \mathrm{OFV}_2^i\right)^2}. \tag{5.4}$$

– **Computational time (CPU time)**

The computational time required by each algorithm to find optimal or near-optimal solution is another criterion to evaluate the performance of an optimizer [23].

– **Set coverage (C-*metric*)**

Consider two Pareto fronts denoted as PF_1 and PF_2. $C\left(\mathrm{PF}_1, \mathrm{PF}_2\right)$ indicates the percentage of solutions on the PF_2 dominated by at least one solution of PF_1 [77]:

$$C(\mathrm{PF}_1, \mathrm{PF}_2) = \frac{|i' \in \mathrm{PF}_2 | \exists i \in \mathrm{PF}_1 : i \text{ dominates } i'|}{|\mathrm{PF}_2|} \tag{5.5}$$

where, i and i' are the solutions on the PF_1 and PF_2, respectively. $|\mathrm{PF}_2|$ is the number of solutions on the PF_2.

5.3. Calibrating parameters of algorithms

Proper adjustment of parameters accelerates the convergence of algorithms and it enhances the quality of solutions. In this study, we use the Response Surface Methodology (RSM) as an effective statistical approach to detect promising parameters' values. The aim of the RSM is to optimize a response (output variable) which is influenced by several independent input variables (factors). Lower and upper levels of each parameter is determined in the initial step. Then, optimal levels of parameters are obtained *via* the RSM. Equation (3.8) formulates the generalized model of the RSM [62]:

$$y = f(\delta_1, \delta_2, \ldots, \delta_n) + \varepsilon \tag{5.6}$$

where, y represents a response variable and n denotes the number of independent input variables $(\delta_1, \ldots, \delta_n)$. ε represents an error, while f is a response function. To realize the condition of the response surface, the RSM detects minimum and maximum points; therefore, the region of optimal response is obtained. In this research, we have used the Box–Behnken design (BBD) as one of the renowned response surface methodology design which is often used to tune full quadratic models [62]. It requires only three levels to run an experiment. Three levels (-1), (0), and $(+1)$ have been considered to indicate low, zero, and high levels of variables, respectively [53]. The most effective factors of the MOMAOA are reported in Table 9.

This study used the response variable (y) introduced by Rahmati *et al.* [63] which has been called the Multi-Objective Coefficient of Variation (MOCV) for the Pareto-based algorithms. The RSM is conducted on large-size test problems with 200 activities. Each combination of different levels obtained by the Box–Behnken designs is implemented five times. To compute the MOCV as the response variable for each experiment, the results are turned into the Relative Percentage Difference (RPD) [21]. Then, MOCV is computed for all experiments. The final tuned values of the MOMAOA are $G = 4$, $\mathrm{NA} = 75$, $\rho = 0.90$ and $\mathrm{MaxIt} = 300$.

TABLE 9. Factors and levels for the RSM.

Factor	Symbol	Coded level		
		-1	0	1
Number of groups	G	2	3	4
Number of agents in each group	NA	25	50	75
Acceptance probability	ρ	0.80	0.85	0.90
Maximum number of iterations	MaxIt	100	200	300

5.4. Comparative analysis

In this section, we compare the performances of algorithms in solving the test problems. Table 10 reports the average values of the MID, SM, DM, and CPU time that have been obtained by 10 runs of algorithms for each test problem. Based on the MID metric, the MOMAOA has strongly prevailed other methods. This means that the MOMAOA had better convergence in comparison with the NSGA-II, PESA-II, and MOPSO. In terms of the SM metric, the proposed method outperformed other algorithms. This implies that the MOMAOA has succeeded to find more uniformly distributed solutions. The outputs of the algorithms in terms of the DM metric show that the solution set found by the MOMAOA covers a wider space comparing to the NSGA-II, PESA-II, and MOPSO. It can be inferred from Table 10 that the MOPSO has the best performance. More investigations of Table 10 reveal that by increasing the size of problems, the values of performance measures also increased. Table 11 shows the standard deviations of values acquired by the MOMAOA, NSGA-II, PESA-II, and MOPSO. As shown in this table, the MOMAOA has obtained more consistent outputs than the NSGA-II, PESA-II, and MOPSO.

To examine if the performances of the algorithms are significantly different or not, the algorithms are statistically compared *via* the Kruskal–Wallis test. To compare these four methods statistically, the following hypothesis test is considered. The Matlab 2017b is used to conduct the Kruskal–Wallis test at a 95% confidence interval. A null hypothesis (H_0) is rejected in favor of the alternative hypothesis if the *P-Value* is less than or equal to 5%.

$$\begin{aligned}&H_0\text{; There is no significant difference between algorithms in terms of a performance measure}\\&H_1\text{; There is significant difference between algorithms in terms of a performance measure.}\end{aligned}\tag{5.7}$$

Tables A.1–A.4 report the outputs of the Kruskal–Wallis test. To make this paper as succinct as possible, we reported the outputs of the Kruskal–Wallis tests (Tabs. A.1–A.4) in Appendix A. However, the conclusions that can be used from these tests have been summarized as follows: The results in Table A.1 indicate that there are significant differences between these four algorithms in terms of the MID metric (P-Value $= 0.0173 < 0.05$). Based on the outputs in terms of the SM metric in Table A.2, the performances of algorithms are not significantly different at a 95% confidence interval (P-Value $= 0.9679 > 0.05$). Table A.3 shows that the algorithms do not perform statistically equal in terms of the DM metric (P-Value $= 0.0437 < 0.05$). Ultimately, the outputs in Table A.4 imply that the performances of algorithms are not statistically different in terms of CPU time (P-Value $= 0.8361 > 0.05$). Figure 17 presents interval plots to clarify the statistical results better. The upper left plot in Figure 17 indicates that in terms of the MID metric, the MOMAOA is superior to other methods. The MOPSO is ranked the second, while the PESA-II takes the third place. The upper right plot in Figure 17 indicates the interval plots in terms of the SM metric. Based on this plot, the MOMAOA is ranked the best, the PESA-II is the second, and the MOPSO is the third. The lower left plot in Figure 17 shows that in terms of the DM metric, the MOMAOA is the best ranked method, the MOPSO is the second, and the NSGA-II is the third. The lower right plot in Figure 17 implies that the MOPSO is the fastest method. Then, the MOMAOA has taken the second place, and the NSGA-II is the slowest algorithm.

TABLE 10. Comparison of algorithms in terms of the MID, SM, DM, and CPU time.

Problem No.	MID				SM				DM				CPU time			
	MOMAOA	NSGA-II	PESA-II	MOPSO	MOMAOA	NSGA-II	PESA-II	MOPSO	MOMAOA	NSGA-II	PESA-II	MOPSO	MOMAOA	NSGA-II	PESA-II	MOPSO
1	54.07	337.16	332.78	97.85	1.03	1.84	1.27	1.16	4885.26	2153.31	2625.07	3084.74	43.57	49.72	46.21	39.15
2	50.96	331.82	328.73	95.92	0.96	1.70	1.39	1.10	4839.66	3099.35	3111.74	3914.87	42.39	45.14	43.40	38.66
3	68.21	346.11	341.15	107.22	2.18	3.58	3.34	2.76	4538.64	2679.61	2189.36	4365.72	48.34	53.18	51.25	45.79
4	65.38	342.48	339.58	97.06	2.11	3.24	3.08	2.84	4007.22	4300.21	2295.92	2691.43	44.27	49.88	46.51	41.55
5	83.27	371.04	366.36	117.85	3.46	4.17	4.12	3.98	4873.16	3194.24	4726.15	4872.53	50.82	57.26	55.93	46.16
6	88.64	376.21	368.98	136.91	3.25	3.98	3.65	3.61	4315.65	4631.94	4768.40	4418.13	45.24	49.04	47.33	43.15
7	134.92	443.78	439.83	167.97	5.83	6.53	6.28	6.02	3951.45	2445.54	3625.63	3812.20	60.35	62.44	61.65	56.88
8	137.65	448.34	442.05	184.17	5.75	6.46	6.15	5.97	5128.00	2691.41	2079.34	2955.31	67.26	71.03	69.84	64.76
9	159.02	462.39	456.37	199.79	7.31	7.79	7.60	7.55	5001.89	2336.62	2604.34	4339.58	79.33	80.67	79.17	74.39
10	164.69	468.11	461.44	214.61	7.74	8.68	8.22	8.16	4350.31	2308.21	2959.48	3201.22	83.56	89.27	86.70	79.42
11	179.91	501.17	496.23	211.47	9.08	9.76	9.54	9.81	4494.95	4507.88	4363.58	4468.04	101.62	105.09	104.31	98.12
12	183.08	526.97	514.55	221.93	9.71	9.83	9.62	9.84	4424.09	3639.11	1946.21	3252.34	72.68	77.38	74.49	69.27
13	256.83	594.26	587.10	288.96	11.35	11.90	11.78	11.67	3665.48	3549.58	2029.07	2520.12	154.34	159.69	155.16	152.16
14	265.30	607.28	591.09	314.54	12.07	13.49	13.11	12.75	3852.49	2334.86	2406.97	2557.97	142.64	148.64	145.75	138.59
15	297.54	646.01	629.75	327.63	14.78	16.94	16.07	15.52	4191.85	4459.09	3847.35	4130.54	160.18	171.27	168.16	158.05
16	291.15	641.76	623.06	336.65	14.04	16.51	16.23	16.19	3710.98	3766.17	4095.17	3891.05	170.21	177.34	173.25	168.00
17	369.63	729.19	706.70	415.98	16.92	18.30	17.62	17.64	4938.62	2952.86	3843.24	4394.28	109.26	113.81	110.61	105.70
18	371.86	740.82	713.91	419.23	17.05	18.24	17.88	18.15	3639.53	3439.75	3252.77	3517.96	121.36	126.93	124.50	119.07
19	652.58	1199.14	1165.28	684.27	23.13	24.45	23.86	25.87	7586.99	6320.29	8454.45	8697.13	105.37	111.69	106.01	100.92
20	661.04	1203.08	1170.33	699.04	23.65	24.62	23.97	24.79	7169.67	7561.27	8052.64	7213.45	86.19	91.16	90.68	81.74
21	768.53	1292.11	1273.41	803.73	24.77	25.46	25.27	25.33	7886.70	7082.52	7767.32	7879.31	77.62	79.07	77.83	73.45
22	772.15	1308.35	1281.67	818.15	25.80	26.67	26.35	26.05	7975.11	7937.16	6049.60	7953.10	79.04	82.52	80.38	76.59
23	871.81	1392.03	1364.19	910.44	29.26	33.82	32.57	31.61	8396.67	7745.11	7578.82	8312.11	71.37	77.07	73.86	67.39
24	879.99	1396.85	1371.72	928.20	31.67	35.78	34.06	37.11	8731.43	8311.16	7155.78	8434.03	76.29	82.39	80.91	72.73
25	1310.79	2037.61	1936.84	1344.43	43.72	44.02	43.89	44.65	7042.06	8272.77	8518.92	7116.21	135.47	144.80	141.26	130.55
26	1315.35	2051.23	1963.41	1350.63	42.86	43.35	43.21	43.48	7749.93	6602.49	7546.97	7714.84	219.21	230.54	226.06	215.26
27	1723.78	2678.55	2555.90	1756.69	46.36	49.44	48.23	47.13	9904.06	7696.24	8000.99	9397.36	131.20	135.73	132.55	126.80
28	1735.40	2688.67	2568.34	1768.12	47.15	51.16	50.68	49.85	6816.88	6193.43	6961.39	6753.46	157.91	163.80	159.24	154.55
29	2597.68	3442.30	3186.17	2645.07	59.60	63.33	61.89	61.66	10 415.42	5691.62	6897.17	9660.11	214.36	223.12	218.11	211.06
30	2410.22	3256.28	3228.29	2451.81	57.08	61.64	58.16	60.12	9621.32	7832.22	8075.94	9482.98	202.38	218.57	215.13	197.90
31	3278.98	4462.40	4331.02	3319.98	87.93	95.80	91.46	90.96	8654.44	7100.07	5850.41	6864.40	251.39	256.33	252.01	249.14
32	3281.83	4447.36	4254.12	3314.73	91.24	98.51	94.38	96.18	9014.10	7039.77	5999.51	6682.42	241.32	262.29	258.29	238.92
33	4449.26	5359.68	5271.57	4496.32	94.08	101.11	97.56	98.54	7649.13	8314.17	6120.17	7278.61	191.35	196.63	192.81	188.83
34	4458.51	5559.76	5323.53	4500.95	107.99	116.03	112.20	111.63	8535.40	7429.60	6772.81	7670.72	168.17	172.56	170.85	165.00
35	5456.38	7036.70	6603.23	5493.40	128.75	135.69	131.71	130.29	10 552.35	7453.00	8094.14	9638.64	336.19	351.62	348.78	331.70
36	5398.79	6901.61	6554.08	5439.05	124.95	133.40	128.57	129.36	10 651.93	8178.33	8010.09	10 295.32	318.24	326.90	322.12	313.83
Average	1256.81	1850.79	1781.74	1296.69	34.29	36.87	35.69	35.81	6476.75	5312.53	5241.03	5928.67	129.46	135.96	133.09	125.98

TABLE 11. Standard deviation of values obtained by algorithms.

Problem No.	MID				SM				DM				CPU time			
	MOMAOA	NSGA-II	PESA-II	MOPSO	MOMAOA	NSGA-II	PESA-II	MOPSO	MOMAOA	NSGA-II	PESA-II	MOPSO	MOMAOA	NSGA-II	PESA-II	MOPSO
1	5.15	10.66	8.45	7.92	0.05	0.52	0.40	0.35	126.46	146.63	141.01	135.31	1.39	2.19	1.97	1.82
2	5.01	9.46	7.59	6.89	0.12	0.48	0.35	0.27	52.07	70.00	63.27	58.02	1.46	2.26	1.94	1.75
3	2.33	6.83	5.56	4.80	0.18	0.68	0.55	0.46	96.56	109.90	106.24	101.59	0.80	1.58	1.31	1.22
4	2.05	6.97	5.61	4.90	0.15	0.68	0.55	0.47	120.34	140.37	133.69	128.97	1.38	2.12	1.90	1.75
5	2.86	8.89	6.56	5.73	0.21	0.62	0.51	0.45	104.98	124.47	118.30	112.61	0.93	1.68	1.35	1.26
6	4.52	8.55	7.51	6.70	0.07	0.64	0.45	0.38	43.28	61.78	56.47	52.09	1.65	2.32	2.02	1.87
7	2.81	7.47	5.73	5.08	0.19	0.65	0.51	0.44	114.01	131.98	127.22	120.83	1.19	1.71	1.50	1.35
8	2.98	8.70	7.45	6.73	0.10	0.51	0.33	0.23	72.75	85.52	81.49	75.61	1.88	2.65	2.32	2.26
9	2.80	7.36	4.89	4.38	0.10	0.54	0.34	0.28	124.19	142.97	136.96	130.18	0.81	1.50	1.30	1.21
10	4.86	11.14	9.07	8.08	0.18	0.64	0.46	0.37	81.89	97.05	93.13	86.33	0.56	1.14	0.94	0.86
11	3.90	9.94	8.19	7.61	0.05	0.58	0.40	0.32	71.68	84.42	81.17	74.62	0.83	1.64	1.29	1.14
12	1.83	7.52	5.82	5.27	0.11	0.65	0.48	0.41	72.78	87.54	81.47	77.34	1.35	2.02	1.72	1.54
13	3.45	6.78	5.69	5.00	0.05	0.53	0.41	0.35	105.00	124.45	118.76	114.48	0.97	1.62	1.36	1.26
14	2.75	7.43	5.40	4.80	0.17	0.61	0.43	0.36	75.72	90.81	84.95	79.67	1.53	2.30	2.02	1.85
15	2.09	5.99	4.93	4.19	0.05	0.58	0.35	0.28	77.17	91.88	86.31	80.19	1.80	2.69	2.37	2.22
16	2.80	7.60	6.50	5.83	0.18	0.63	0.40	0.34	101.83	119.16	114.49	107.87	1.67	2.43	2.23	2.18
17	4.72	8.85	7.07	6.09	0.14	0.50	0.36	0.28	69.32	88.14	83.58	77.95	1.19	1.86	1.56	1.42
18	3.38	8.33	7.19	6.23	0.17	0.53	0.40	0.34	37.07	54.95	48.68	41.40	1.70	2.42	2.12	2.01
19	2.73	7.23	5.00	4.47	0.08	0.60	0.42	0.35	127.82	145.01	140.74	133.87	0.75	1.59	1.32	1.13
20	2.53	8.39	6.16	5.29	0.16	0.66	0.46	0.38	102.43	117.76	111.50	103.63	0.97	1.64	1.35	1.30
21	2.23	7.55	5.46	4.83	0.16	0.56	0.40	0.34	54.99	75.82	69.66	63.53	2.14	2.98	2.73	2.61
22	3.01	8.18	6.95	6.24	0.12	0.55	0.42	0.36	60.93	79.13	72.72	67.42	1.31	2.02	1.80	1.69
23	3.59	9.36	7.37	6.60	0.14	0.61	0.37	0.29	96.58	114.36	109.34	104.92	1.32	1.93	1.67	1.55
24	3.86	9.90	8.12	7.15	0.10	0.43	0.31	0.25	117.50	131.30	125.75	119.31	1.12	2.01	1.67	1.50
25	2.99	9.68	7.22	6.51	0.14	0.62	0.51	0.42	61.94	82.41	75.60	68.48	1.80	2.53	2.29	2.19
26	1.72	6.52	4.55	3.56	0.16	0.60	0.48	0.38	62.66	80.68	75.90	70.21	1.46	2.18	1.93	1.76
27	2.63	8.57	6.37	5.72	0.05	0.56	0.43	0.34	66.12	78.20	74.96	70.60	1.79	2.41	2.16	2.04
28	1.92	6.09	4.41	3.56	0.23	0.76	0.56	0.49	103.19	121.33	114.87	109.80	1.95	2.47	2.26	2.20
29	2.08	5.81	4.16	3.33	0.10	0.60	0.41	0.33	16.42	32.23	26.71	22.10	1.22	1.87	1.59	1.51
30	1.86	7.96	5.73	4.96	0.15	0.62	0.51	0.45	99.68	115.92	111.50	106.38	0.96	1.69	1.37	1.21
31	2.35	6.73	5.60	4.75	0.10	0.72	0.48	0.38	139.81	157.59	150.60	144.84	1.37	2.20	1.86	1.74
32	2.24	6.64	5.44	4.61	0.11	0.72	0.51	0.42	90.42	105.22	101.32	95.21	1.71	2.62	2.31	2.24
33	2.84	8.37	7.11	6.52	0.08	0.49	0.28	0.19	118.10	132.98	127.37	121.54	1.28	1.90	1.64	1.54
34	2.33	7.44	5.85	5.29	0.07	0.56	0.45	0.39	77.23	92.44	87.02	79.52	1.70	2.47	2.22	2.08
35	4.44	10.23	7.98	6.98	0.13	0.68	0.45	0.37	47.85	60.55	56.00	49.93	0.38	1.25	0.95	0.87
36	3.18	9.54	7.33	6.74	0.13	0.77	0.53	0.48	102.35	120.38	116.81	109.04	0.30	1.07	0.74	0.58
Average	3.02	8.13	6.39	5.65	0.12	0.60	0.43	0.36	85.92	102.65	97.38	91.54	1.29	2.03	1.75	1.63

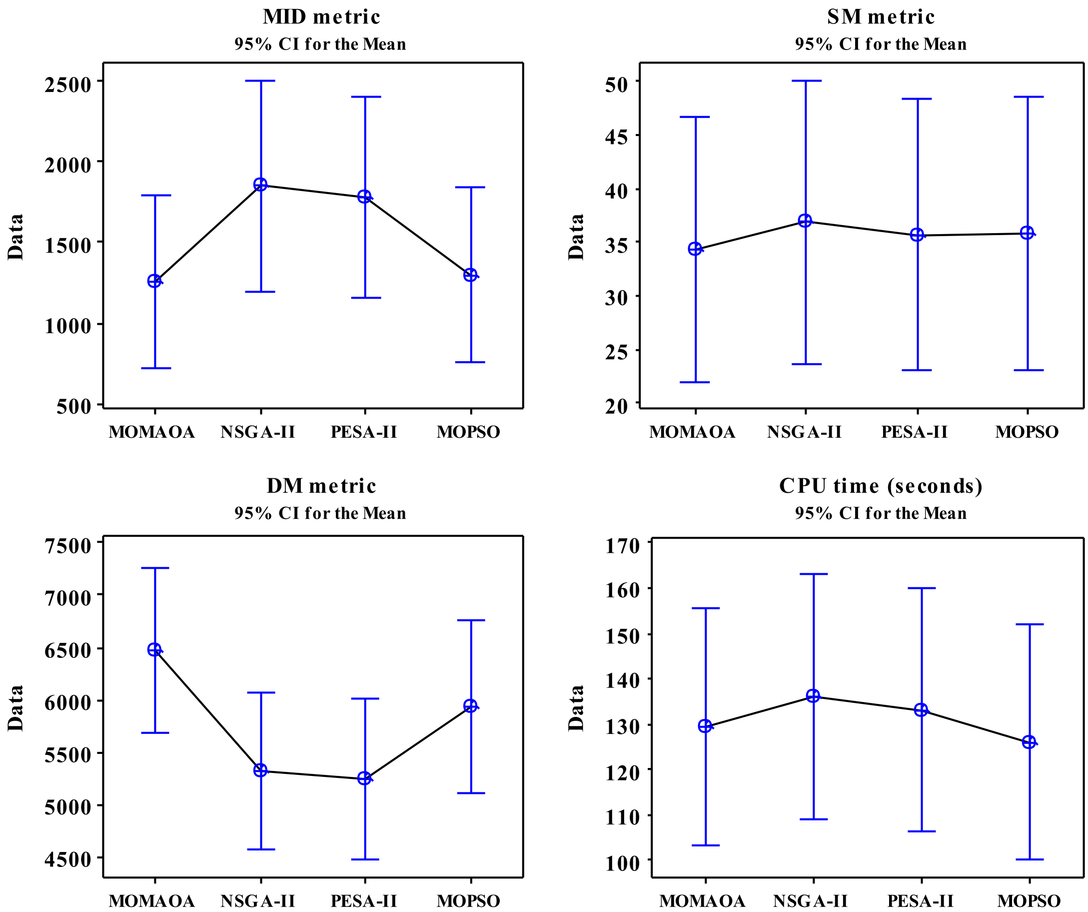

FIGURE 17. Comparison of algorithms on the MID, SM, DM, and CPU time.

Table 12 reports the comparisons between algorithms in terms of set coverage metric (*C-metric*). Figure 18 shows the boxplots of *C-metric* values for all algorithms. The lower and the upper ends of each box imply the first and the third quartiles, respectively. The line in each box indicates the median.

It can be inferred from Table 12 and Figure 18 that the MOMAOA obtained larger C (MOMAOA, NSGA-II), C (MOMAOA, PESA-II), and C (MOMAOA, MOPSO) values for most of test problems. It means that the Pareto solutions obtained by the MOMAOA are more dominant than the solutions obtained by the NSGA-II, PESA-II, and MOPSO. The statistical analysis has been conducted on the *C-metric* values as well; hence, the non-parametric Kruskal–Wallis test is again hired on a 95% confidence interval. The null hypothesis assumes that there is no significant difference between the performances of algorithms. If P-Value < 0.05, the null hypothesis is rejected. Tables 13–15 report the results of the Kruskal–Wallis tests. The outputs of the Kruskal–Wallis tests show the significant differences between algorithms in terms of *C-metric*.

Table 16 reports the average of objective function values obtained by solving the test problems of the iMOPSE dataset. The algorithms have been run for 10 times and the outputs have been obtained by 10 runs of each algorithm. Since the algorithms have some probabilistic features, the average values have been reported to evaluate the overall performance of optimizers. As shown in Table 16, the MOMAOA has strongly prevailed other methods in terms of both make-span and total cost of project.

TABLE 12. Comparisons between the MOMAOA, NSGA-II, PESA-II, and MOPSO in terms of the *C-metric*.

Problem No.	C(MOMAOA, NSGA-II)	C(NSGA-II, MOMAOA)	C(MOMAOA, PESA-II)	C(PESA-II, MOMAOA)	C(MOMAOA, MOPSO)	C(MOPSO, MOMAOA)
1	0.74	0.12	0.71	0.16	0.65	0.29
2	0.68	0.26	0.65	0.30	0.57	0.30
3	0.59	0.38	0.53	0.44	0.70	0.24
4	0.92	0.06	0.86	0.15	0.51	0.17
5	0.85	0.09	0.79	0.19	0.59	0.20
6	0.73	0.14	0.66	0.24	0.58	0.33
7	0.65	0.23	0.61	0.30	0.66	0.18
8	0.61	0.19	0.57	0.26	0.67	0.32
9	0.94	0.00	0.86	0.03	0.54	0.26
10	0.86	0.02	0.82	0.06	0.60	0.35
11	0.48	0.21	0.41	0.30	0.59	0.17
12	0.26	0.05	0.21	0.13	0.64	0.24
13	0.91	0.07	0.84	0.09	0.65	0.17
14	0.52	0.33	0.45	0.38	0.66	0.34
15	0.98	0.00	0.89	0.02	0.56	0.15
16	0.83	0.10	0.77	0.15	0.64	0.31
17	0.60	0.28	0.54	0.36	0.61	0.28
18	0.57	0.16	0.52	0.24	0.53	0.32
19	0.81	0.11	0.75	0.13	0.58	0.17
20	0.72	0.14	0.69	0.22	0.60	0.23
21	0.99	0.00	0.93	0.03	0.70	0.20
22	0.91	0.00	0.86	0.07	0.57	0.31
23	0.65	0.22	0.61	0.30	0.62	0.24
24	0.44	0.37	0.40	0.46	0.55	0.33
25	0.87	0.10	0.83	0.18	0.66	0.19
26	0.82	0.18	0.78	0.12	0.55	0.20
27	0.76	0.15	0.73	0.22	0.61	0.18
28	0.58	0.31	0.51	0.37	0.65	0.14
29	0.98	0.01	0.93	0.00	0.69	0.26
30	0.79	0.13	0.72	0.20	0.70	0.14
31	0.62	0.28	0.55	0.38	0.61	0.26
32	0.91	0.00	0.87	0.08	0.53	0.18
33	0.66	0.24	0.62	0.27	0.56	0.32
34	0.96	0.02	0.89	0.08	0.55	0.27
35	0.98	0.00	0.92	0.00	0.68	0.22
36	0.33	0.00	0.25	0.05	0.55	0.25
Average	0.74	0.14	0.68	0.19	0.61	0.24

TABLE 13. Kruskal–Wallis test for difference of the MOMAOA and NSGA-II in terms of *C-Metric*.

Source	SS	*df*	MS	Chi-sq	P-Value	Result
Columns	22 684.5	1	22 684.5	51.85	5.98 e-0.13	H_0 is rejected
Error	8378	70	119.7			
Total	31 062.5	71				

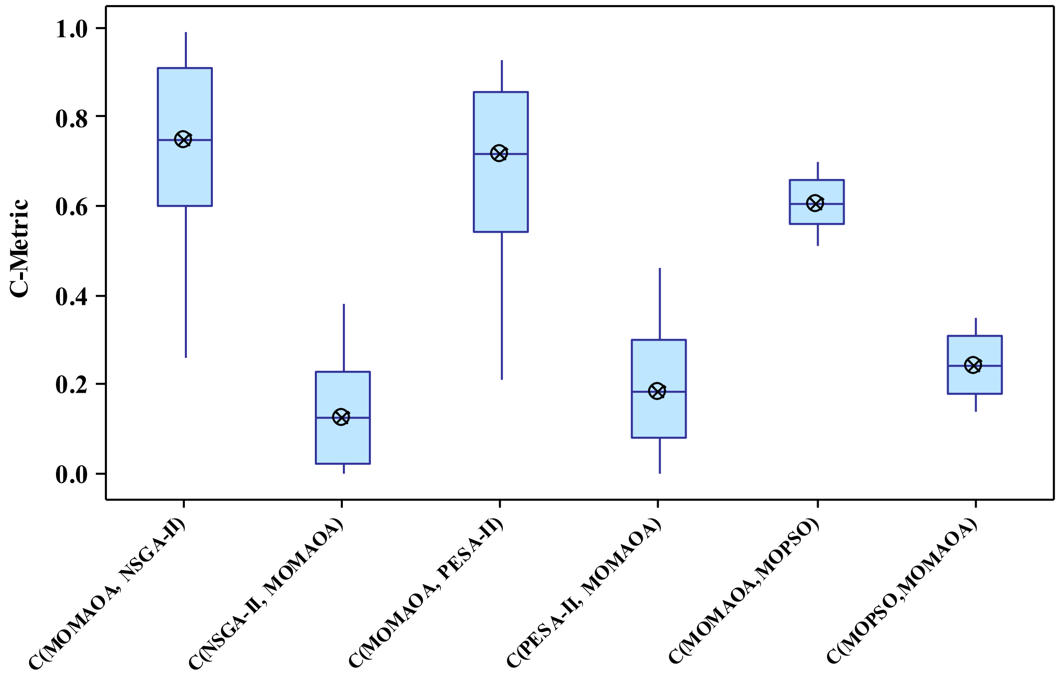

FIGURE 18. Boxplots of *C-metric* values.

TABLE 14. Kruskal–Wallis test for difference of the MOMAOA and PESA-II in terms of *C-Metric*.

Source	SS	*df*	MS	Chi-sq	*P*-Value	Result
Columns	21 012.5	1	21 012.5	47.99	4.27 e-012	H_0 is rejected
Error	10 073	70	143.9			
Total	31 085.5	71				

TABLE 15. Kruskal–Wallis test for difference of the MOMAOA and MOPSO in terms of *C-Metric*.

Source	SS	*df*	MS	Chi-sq	*P*-Value	Result
Columns	23 328	1	23 328	53.32	2.83 e-013	H_0 is rejected
Error	7737	70	110.53			
Total	31065	71				

5.5. Impact of resource transfer times on objective function values

To examine the impact of transfer times on make-span and total cost of project, all algorithms were used to solve the problems with and without consideration of transfer times. The best values of algorithms were averaged and the results of both cases have been shown in Figure 19. According to this figure, transfer times have a remarkable impact on both objectives. The Kruskall–Wallis test has been used to offer a statistical analysis on the effect of transfer times on the objectives. From a statistical perspective, Tables A.5 and A.6 (Appendix A) indicate that transfer times can significantly increase both objectives.

TABLE 16. Comparing the algorithms in terms of objective function values.

Problem No.	MOMAOA		NSGA-II		PESA-II		MOPSO	
	Make-span	Cost	Make-span	Cost	Make-span	Cost	Make-span	Cost
1	175.28	53 439.38	242.34	98 429.20	221.44	71 282.40	218.24	68 309.11
2	134.56	118 884.22	201.96	164 197.34	180.37	136 683.63	175.25	127 805.62
3	131.12	130 976.81	199.81	178 238.94	178.98	150 602.92	160.75	142 069.24
4	166.83	143 969.13	236.16	191 238.85	214.59	162 550.56	202.52	154 008.66
5	131.85	124 837.55	202.25	169 570.04	180.72	144 710.63	174.72	135 257.19
6	243.53	117 990.66	312.73	165 116.37	291.73	137 821.53	285.70	129 536.63
7	223.16	44 600.05	291.57	91 460.77	270.74	64 059.99	252.35	55 739.56
8	238.09	129 659.15	307.37	175 516.74	286.36	148 622.49	273.53	139 665.62
9	259.50	144 862.30	329.56	189 717.60	308.82	164 768.25	293.36	155 599.47
10	248.31	138 144.47	318.20	182 760.08	296.59	156 119.91	291.05	145 802.34
11	249.49	116 029.14	317.10	162 021.40	295.38	134 862.02	287.39	126 382.57
12	247.99	154 295.93	319.18	198 807.32	297.65	172 566.29	282.58	164 504.15
13	395.85	41 614.72	464.18	89 549.36	442.50	60 643.09	420.09	52 421.72
14	487.94	121 590.86	555.98	169 177.08	534.86	140 606.28	528.57	131 627.17
15	493.60	195 567.11	560.95	240 623.23	540.33	213 239.23	530.65	206 510.69
16	532.07	207 148.61	602.50	250 758.78	581.26	224 352.59	561.84	211 518.63
17	477.88	103 848.60	546.67	148 463.13	525.11	123 190.16	512.85	114 663.45
18	486.08	148 257.29	556.62	197 248.33	535.12	167 790.47	527.03	159 389.43
19	185.64	275 928.69	256.25	321 404.12	234.77	294 568.47	228.20	286 515.64
20	167.00	264 742.15	234.03	311 333.52	212.81	283 470.78	269.48	346 796.92
21	184.60	297 448.30	252.98	343 565.00	232.28	316 381.63	225.20	307 716.63
22	147.71	250 889.95	217.07	297 917.02	196.64	269 927.00	165.53	261 487.51
23	516.14	105 776.55	584.53	150 543.01	562.95	125 073.05	550.83	116 485.21
24	154.35	284 963.54	221.80	331 793.39	200.71	302 826.03	181.42	288 605.13
25	260.86	234 188.82	328.37	281 621.60	308.07	253 010.99	294.78	244 747.09
26	264.14	300 981.35	333.80	346 860.81	312.09	318 439.51	298.18	302 517.86
27	251.17	279 620.82	318.97	325 261.66	297.83	299 373.09	350.16	349 854.10
28	339.66	293 392.21	407.31	341 137.50	386.03	312 607.49	361.36	303 537.31
29	910.62	91 572.01	978.14	138 774.51	956.90	110 737.28	942.79	102 096.54
30	242.48	280 596.93	312.21	324 075.68	290.42	297 704.22	271.21	285 947.36
31	490.93	255 592.20	559.42	301 666.21	538.82	273 865.25	589.95	335 749.83
32	490.84	193 977.32	562.66	238 756.82	541.27	210 980.77	527.31	199 875.49
33	513.07	228 195.03	581.49	274 588.65	560.95	247 505.51	553.23	239 286.55
34	480.69	310 055.95	549.46	355 508.47	527.80	329 410.17	503.15	320 886.71
35	696.26	103 893.93	765.41	147 808.24	744.93	122 054.24	729.12	113 980.62
36	466.25	180 614.95	536.49	222 095.72	515.33	197 635.10	508.29	187 533.92
Average	335.71	179 670.74	404.60	225 489.07	383.42	198 334.53	375.79	194 845.30

6. Conclusions and future extensions

In this paper, a bi-objective mathematical formulation has been developed for the multi-skill RCPSP considering resource transfer times (MSRCPSP-TT). The first objective was to minimize the make-span of the project, while the second objective aimed at minimizing total cost of project. Since the problem was classified as an NP-hard problem, a multi-objective multi-agent optimization algorithm (MOMAOA) was developed to approximate the Pareto-optimal front. The proposed algorithm was inspired by multi-agent system and swarm intelligence. The MOMAOA utilizes the TOPSIS method in various stages such as finding the best ranked agents in each iteration, finding the global best agent, and leader agent of each group. To assess the performance of

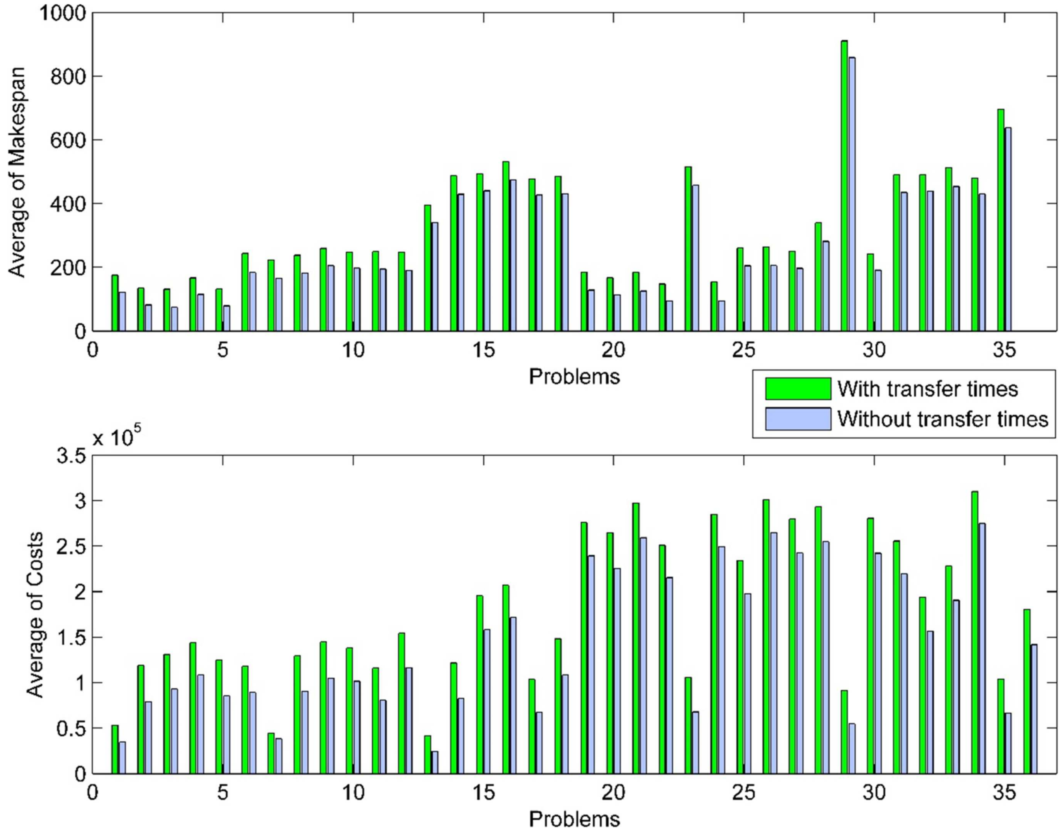

FIGURE 19. Comparing objective function values with and without considering transfer times.

the proposed algorithm and to validate the obtained results, three meta-heuristics called the non-dominated sorting genetic algorithm II (NSGA-II), the Pareto envelope-based selection algorithm II (PESA-II), and the multi-objective particle swarm optimization (MOPSO) method were employed to solve the iMOPSE dataset consisting of 36 test problems. The input parameters of all algorithms were tuned *via* the response surface methodology (RSM). The algorithms were evaluated in terms of several well-known comparison measures. Besides, the algorithms were statistically compared to each other *via* the Kruskal–Wallis test. Based on the computational experiments, the MOMAOA was superior to the other three algorithms in most of evaluations. To show the effect of resource transfer times on the make-span and total cost of project, we solved the iMOPSE test problems with and without considering resource transfer times. The results show that considering resource transfer times has a significant impact on values of both objective functions. To extend the proposed model, the resource transfer times can be considered uncertain. Besides, the multi-skill resource-constrained multi-project scheduling problem with transfer times will also be a potential subject for further studies. Moreover, the newly proposed multi-objective algorithms can also be applied for the MSRCPSP-TT and compared with the proposed algorithm in this research. To solve the iMOPSE test problems, the literature offers some novel algorithms such as the Non-dominated Tournament Genetic Algorithm (NTGA) [44] that offer promising results comparing to classical multi-objective genetic algorithms. Therefore, one of the directions for future studies is to compare the results of recently developed algorithms for the MSRCPSP with the MOMAOA which has been proposed in this study. The MOMAOA can be developed for other complex optimization problems as well.

Appendix A.

Table A.1. Kruskal–Wallis test for difference of algorithms in terms of the MID.

Source	SS	*df*	MS	Chi-sq	P-Value	Result
Columns	17 659.30	3	5886.44	10.15	0.0173	H_0 is rejected
Error	231 160.70	140	1651.15			
Total	248 820.00	143				

Table A.2. Kruskal–Wallis test for difference of algorithms in terms of the SM.

Source	SS	*df*	MS	Chi-sq	P-Value	Result
Columns	447.70	3	149.23	0.26	0.9679	H_0 is not rejected
Error	248 371.80	140	1774.08			
Total	248 819.50	143				

Table A.3. Kruskal–Wallis test for difference of algorithms in terms of the DM.

Source	SS	*df*	MS	Chi-sq	$\boldsymbol{P}$**-Value**	Result
Columns	14 121.10	3	4707.02	8.12	0.0437	H_0 is rejected
Error	234 698.90	140	1676.42			
Total	248 820.00	143				

Table A.4. Kruskal–Wallis test for difference of algorithms in terms of CPU time.

Source	SS	*df*	MS	Chi-sq	P-Value	Result
Columns	1489.30	3	496.43	0.86	0.8361	H_0 is not rejected
Error	247 330.70	140	1766.65			
Total	248 820.00	143				

Table A.5. Kruskal–Wallis test on impact of transfer times on make-span.

Source	SS	*df*	MS	Chi-sq	P-Value	Result
Columns	1680.70	1	1680.70	4.06	0.044	H_0 is rejected
Error	26 896.80	68	395.54			
Total	28 577.50	69				

TABLE A.6. Kruskal–Wallis test on impact of transfer times on costs.

Source	SS	*df*	MS	Chi-sq	*P*-Value	Result
Columns	2005.60	1	2005.56	4.58	0.032	H_0 is rejected
Error	29 092.40	70	415.61			
Total	31 098.00	71				

Acknowledgements. The authors deeply appreciate the anonymous referees for their constructive comments.

References

[1] S. Adhau, M.L. Mittal and A. Mittal, A multi-agent system for distributed multi-project scheduling: an auction-based negotiation approach. *Eng. Appl. Artif. Intell.* **25** (2012) 1738–1751.

[2] P.G. Balaji and D. Srinivasan, An introduction to multi-agent systems, edited by. D. Srinivasan and L.C. Jain. In: Vol. 310 of Innovations in Multi-Agent Systems and Applications – 1. Studies in Computational Intelligence. Springer, Berlin, Heidelberg (2010). DOI: 10.1007/978-3-642-14435-6_1.

[3] O. Bellenguez and E. Néron, Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills, edited by E. Burke and M. Trick. In: Vol. 3616 of Practice and Theory of Automated Timetabling V. PATAT 2004. *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg (2005) 229–243.

[4] J. Blazewicz, J.K. Lenstra and A. Kan, Scheduling subject to resource constraints: classification and complexity. *Disc. Appl. Math.* **5** (1983) 11–24.

[5] A. Brandolese, A. Brun and A.P. Staudacher, A multi-agent approach for the capacity allocation problem. *Int. J. Prod. Econ.* **66** (2000) 269–285.

[6] J. Böcker, J. Lind and B. Zirkler, Using a multi-agent approach to optimize the train coupling and sharing system. *Eur. J. Oper. Res.* **134** (2001) 242–252.

[7] J. Cai, Z. Peng, S. Ding and J. Sun, A robust genetic algorithm to solve multi-skill resource constrained project scheduling problem with transfer time and uncertainty skills. In: 2020 IEEE 16th International Conference on Control & Automation (ICCA), Singapore, 9–11 Oct. (2020). DOI: 10.1109/ICCA51439.2020.9264319.

[8] S. Carpitella, A. Certa, J. Izquierdo and C. Manuela La Fata, k-out-of-n systems: an exact formula for the stationary availability and multi-objective configuration design based on mathematical programming and TOPSIS. *J. Comput. Appl. Math.* **330** (2018) 1007–1015.

[9] Y.M. Chen and S.C. Wang, Framework of agent-based intelligence system with two-stage decision-making process for distributed dynamic scheduling. *Appl. Soft Comput.* **7** (2007) 229–245.

[10] W. Chen, Y.J. Shi, H.F. Teng, X.P. Lan and L.C. Hu, An efficient hybrid algorithm for resource-constrained project scheduling. *Inf. Sci.* **180** (2010) 1031–1039.

[11] R. Chen, C. Liang, D. Gu and J. Leung, A multi-objective model for multi-project scheduling and multi-skilled staff assignment for IT product development considering competency evolution. *Int. J. Prod. Res.* **55** (2017) 6207–6234.

[12] G. Confessore, S. Giordani and S. Rismondo, A market-based multi-agent system model for decentralized multi-project scheduling. *Ann. Oper. Res.* **150** (2007) 115–135.

[13] I. Correia and F. Saldanha-da-Gama, The impact of fixed and variable costs in a multi-skill project scheduling problem: an empirical study. *Comput. Ind. Eng.* **72** (2014) 230–238.

[14] H. Dang Quoc, L.N. The, C.N. Doan and T.P. Thanh, New effective differential evolution algorithm for the project scheduling problem. In: 2020 2nd International Conference on Computer Communication and the Internet (ICCCI), Nagoya, Japan, 26–29 June (2020). DOI: 10.1109/ICCCI49374.2020.9145982.

[15] H. Dang Quoc, L.N. The, C.N. Doan and T.P. Thanh, New Cuckoo Search algorithm for the resource constrained project scheduling problem. In: 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh City, Vietnam, 14–15 Oct. (2020). DOI: 10.1109/RIVF48685.2020.9140728.

[16] H. Dang Quoc, L.N. The, C.N. Doan and N. Xiong, Effective evolutionary algorithm for solving the real-resource-constrained scheduling problem. *J. Adv. Transp.* **2020** (2020). DOI: 10.1155/2020/8897710.

[17] H. Dai, W. Cheng and P. Guo, An improved tabu search for multi-skill resource-constrained project scheduling problems under step-deterioration. *Arab. J. Sci. Eng.* **43** (2018) 3279–3290.

[18] H. Dai, W. Cheng, W. Yang and Y. Wang, A general variable neighbourhood search for multi-skill resource-constrained project scheduling problem with step-deterioration. *Int. J. Ind. Syst. Eng.* **34** (2020) 145–164.

[19] K. Deb, A. Pratap, S. Agrawal and T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6** (2000) 182–197.

[20] Y. Fu, H. Wang, G. Tian, Z. Li and H. Hu, Two-agent stochastic flow shop deteriorating scheduling via a hybrid multi-objective evolutionary algorithm. *J. Intell. Manuf.* **30** (2019) 2257–2272.

[21] J. Gao, R. Chen and W. Deng, An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. *Int. J. Prod. Res.* **51** (2013) 641–651.

[22] W.J. Gutjahr, S. Katzensteiner, P. Reiter, C. Stummer and M. Denk, Competence-driven project portfolio selection, scheduling and staff assignment. *Cent. Eur. J. Oper. Res.* **16** (2008) 281–306.

[23] V. Hajipour, E. Mehdizadeh and R. Tavakkoli-Moghaddam, A novel Pareto-based multi-objective vibration damping optimization algorithm to solve multi-objec- tive optimization problems. *Sci. Iran. Trans. E* **21** (2014) 2368–2378.

[24] D. Han, B. Yang, J. Li, J. Wang, M. Sun and Q. Zhou, A multi-agent-based system for two-stage scheduling problem of offshore project. *Adv. Mech. Eng.* **9** (2017) 1–17.

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

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

[27] S. He, E. Qi and G. Li, A study on the project scheduling based on multi-agent systems. *Math. Pract. Theory* **1** (2005) 43–47.

[28] C. Hermerl and R. Kolisch, Scheduling and staffing multiple projects with a multi-skilled workforce. *OR Spectr.* **32** (2010) 343–368.

[29] J. Homberger, A multi-agent system for the decentralized resource-constrained multi-project scheduling problem. *Int. Trans. Oper. Res.* **14** (2007) 565–599.

[30] A.H. Hosseinian and V. Baradaran, A multi-objective multi-agent optimization algorithm for the community detection problem. *J. Inf. Syst. Telecommun.* **6** (2019) 169–179.

[31] A.H. Hosseinian and V. Baradaran, Detecting communities of workforces for the multi-skill resource-constrained project scheduling problem: a dandelion solution approach. *J. Ind. Syst. Eng. (JISE)* **12** (2019) 72–99.

[32] A.H. Hosseinian and V. Baradaran, An evolutionary algorithm based on a hybrid multi-attribute decision making method for the multi-mode multi-skilled resource-constrained project scheduling problem. *J. Optim. Ind. Eng.* **12** (2019) 155–178.

[33] A.H. Hosseinian and V. Baradaran, P-GWO and MOFA: two new algorithms for the MSRCPSP with the deterioration effect and financial constraints (case study of a gas treating company). *Appl. Intell.* **50** (2020) 2151–2176.

[34] A.H. Hosseinian and V. Baradaran, Modified Pareto archived evolution strategy for the multi-skill project scheduling problem with generalized precedence relations. *J. Ind. Eng. Manage. Stud. (JIEMS)* **7** (2020) 59–86.

[35] A.H. Hosseinian, V. Baradaran and M. Bashiri, Modeling of the time-dependent multi-skilled RCPSP considering learning effect: an evolutionary solution approach. *J. Model. Manage. (JM2)* **14** (2019) 521–558.

[36] S. Javanmard, B. Afshar-Nadjafi and S.T.A. Niaki, Preemptive multi-skilled resource investment project scheduling problem; mathematical modelling and solution approaches. *Comput. Chem. Eng.* **96** (2016) 55–68.

[37] N.R. Jennings, K. Sycara and M. Wooldridge, A roadmap of agent research and development. *Auton. Agents Multi-Agent Syst.* **1** (1998) 7–38.

[38] R.L. Kadri and F.F. Boctor, An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: the single mode case. *Eur. J. Oper. Res.* **265** (2018) 454–462.

[39] H. Kazemipoor, R. Tavakkoli-Moghaddam, P. Shahrezaei and A. Azaron, A differential evolution algorithm to solve multi-skilled project portfolio scheduling problems. *Int. J. Adv. Manuf. Tech.* **64** (2013) 1099–1111.

[40] G. Knotts and M. Dror, Agent-based project scheduling: computational study of large problems. *IIE Trans.* **35** (2003) 143–159.

[41] G. Knotts, M. Dror and B. Hartman, Agent-based project scheduling. *IIE Trans.* **32** (2000) 387–401.

[42] D. Krüger and A. Scholl, A heuristic solution framework for the resource constrained (multi-) project scheduling problem with sequence-dependent transfer times. *Eur. J. Oper. Res.* **197** (2009) 492–508.

[43] D. Krüger and A. Scholl, Managing and modelling general resource transfers in (multi-) project scheduling. *OR Spectr.* **32** (2010) 369–394.

[44] M. Laszczyk and P. Myszkowski, Improved selection in evolutionary multi-objective optimization of multi-skill resource-constrained project scheduling problem. *Inf. Sci.* **481** (2019) 412–431.

[45] Y.H. Lee, K. Chatterjee and S.R.T. Kumara, Multi-agent based dynamic resource scheduling for distributed multiple project using a market mechanism. *J. Intell. Manuf.* **14** (2003) 471–484.

[46] K.Y. Li and R.J. Willis, An iterative scheduling technique for resource constrained project scheduling. *Eur. J. Oper. Res.* **56** (1992) 370–379.

[47] H. Li and K. Womer, Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. *J. Sched.* **12** (2009) 281–298.

[48] J. Li, H. Jing and Y.Y. Tang, Multi-agent oriented constraint satisfaction. *Artif. Intell.* **136** (2002) 101–144.

[49] J. Lin, L. Zhu and K. Gao, A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. *Expert Syst. App.* **140** (2020) 112915.

[50] S. Liu and C. Wang, Optimizing linear project scheduling with multi-skilled crews. *Autom. Construct.* **24** (2012) 16–23.

[51] H.R. Maghsoudlou, B. Afshar-Nadjafi and S.T.A. Niaki, A multi-objective invasive weeds optimization algorithm for solving multi-skill multi-mode resource constrained project scheduling problem. *Comput. Chem. Eng.* **8** (2016) 157–169.

[52] H.R. Maghsoudlou, B. Afshar-Nadjafi and S.T.A. Niaki, Multi-skilled project scheduling with level-dependent rework risk; three multi-objective mechanisms based on cuckoo search. *Appl. Soft Comput.* **54** (2017) 46–61.

[53] A. Majumder, A. Singh and A. Goyal, Application of response surface methodology for Glucan production from leuconostoc dextranicum and its structural characterization. *Carbohydr. Polym.* **75** (2009) 150–156.

[54] S. Martin, D. Ouelhadj, P. Beullens, E. Ozcan, A.A. Juan and E.K. Burke, A multi-agent based cooperative approach to scheduling and routing. *Eur. J. Oper. Res.* **254** (2016) 169–178.

[55] E. Mehdizadeh, S.T.A. Niaki and M. Hemati, A bi-objective aggregate production planning problem with learning effect and machine deterioration: modeling and solution. *Comput. Oper. Res.* **91** (2018) 21–36.

[56] E. Mehmanchi and S. Shadrokh, Solving a new mixed integer non-linear programming model of the multi-skilled project scheduling problem considering learning and forgetting effect. In: Proceedings of the 2013 IEEE IEEM. Bangkok, Thailand (2013). DOI: 10.1109/IEEM.2013.6962442.

[57] C. Montoya, O. Bellenguez, E. Pinson and D. Rivera, Branch-and-price approach for the multi-skill project scheduling problem. *Optim. Lett.* **8** (2014) 1721–1734.

[58] P. Myszkowski, M. Skowronski, L.P. Olech and K. Oslizlo, Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem. *Soft Comput.* **19** (2015) 3599–3619.

[59] P.B. Myszkowski, L.P. Olech, M. Laszczyk and M.E. Skowronski, Hybrid differential evolution and greedy algorithm (DEGR) for solving multi-skill resource-constrained project scheduling problem. *Appl. Soft Comput.* **62** (2018) 1–14.

[60] C. Pessan, O. Morineau and E. Neron, Multi-skill project scheduling problem and total productive maintenance. In: Proceedings of 3rd Multidisciplinary International Conference on Scheduling: Theory and Application (MISTA 2007). Paris, France (2007) 608–610.

[61] J. Poppenborg and S. Knust, A flow-based tabu search algorithm for the RCPSP with transfer times. *OR Spectr.* **38** (2016) 305–335.

[62] M. Rabiee, F. Jolai, H. Asefi, P. Fattahi and S. Lim, A biogeography-based optimisation algorithm for a realistic no-wait hybrid flow shop with unrelated parallel machines to minimise mean tardiness. *Int. J. Comput. Integr. Manuf.* **29** (2016) 1007–1024.

[63] S.H.A. Rahmati, V. Hajipour and S.T.A. Niaki, A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem. *Appl. Soft Comput.* **13** (2013) 1728–1740.

[64] J.R. Schott, *Fault tolerant design using single and multicriteria genetic algorithms optimization*, Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA (1995).

[65] B.H. Tabrizi, R. Tavvakoli-Moghaddam and S.F. Ghaderi, A two-phase method for a multi-skilled project scheduling problem with discounted cash flows. *Sci. Iran.* **21** (2014) 1083–1095.

[66] F. Tao, Y.J. Laili, L. Zhang and A.Y.C. Nee, QMAEA: a quantum multi-agent evolutionary algorithm for multi-objective combinatorial optimization. *Simulation* **90** (2014) 182–204.

[67] Y. Tian, T. Xiong, Z. Liu, P. Deng and L. Wan, Novel feedback-based operators in solving multi-skill resource-constrained project scheduling problem. In: 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 2020, 22–24 Aug. (2020), 296–301. DOI: 10.1109/CCDC49329.2020.9164711.

[68] M. Tritschler, A. Naber and R. Kolisch, A hybrid metaheuristic for resource-constrained project scheduling with flexible resource profiles. *Eur. J. Oper. Res.* **262** (2017) 262–273.

[69] L. Wang and X.L. Zheng, A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem. *Swarm Evol. Comput.* **38** (2018) 54–63.

[70] M. Wooldridge and N.R. Jennings, Intelligent agents: theory and practice. *Knowl. Eng. Rev.* **10** (1995) 115–152.

[71] Y. Yan, T. Kuphal and J. Bode, Application of multi-agent system in project management. *Int. J. Prod. Econ.* **68** (2000) 185–197.

[72] R. Zamani, A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem. *Eur. J. Oper. Res.* **229** (2013) 552–559.

[73] X.L. Zheng and L. Wang, A multi-agent optimization algorithm for resource constrained project scheduling problem. *Expert Syst. App.* **42** (2015) 6039–6049.

[74] H. Zheng, L. Wang and X. Zheng, Teaching–learning-based optimization algorithm for multi-skill resource constrained project scheduling problem. *Soft Comput.* **21** (2015) 1537–1548.

[75] W.C. Zhong, J. Liu, M.Z. Xue and L.C. Jiao, A multi-agent genetic algorithm for global numerical optimization. *IEEE Trans. Syst. Man Cybern. B, Cybern.* **34** (2004) 229–244.

[76] L. Zhu, J. Lin and Z-J. Wang, A discrete oppositional multi-verse optimization algorithm for multi-skill resource constrained project scheduling problem. *Appl. Soft Comput.* **85** (2019) 105805.

[77] E. Zitzler and L. Thiele, Multi-objective optimization using evolutionary algorithms a comparative case study, edited by A.E. Eiben, T. Back, M. Schoenauer and H.P. Schwefel. In: Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V). Berlin, Germany (1998) 292–301.