Samin MAHMOODI, MSc
saminmahmoudi@gmail.com
K. N. Toosi University of Technology, Tehran, Iran

Amir Abbas NAJAFI, PhD (corresponding author)
aanajafi@kntu.ac.ir
K. N. Toosi University of Technology, Tehran, Iran

Kian GHODS, PhD Candidate
kianghods@gmail.com
Polytechnique Montreal, Montreal, Canada

A Multi-Objective Water Cycle Algorithm for the BI-Objective Multi-Mode Project Resource Renting Problem

Abstract. *A resource renting problem is a project scheduling problem in which the required resources should be rented, and the goal is to find a schedule and resource renting plan such that the total cost of the resources minimises. Traditionally, the model of a resource renting problem contains single-mode activities and a single objective function. This research aims to present a new mathematical model for a bi-objective multi-mode resource renting problem. The objectives are to minimise the project makespan and also the total cost of resources, including the time-independent resource procurement costs and time-dependent resource renting costs, simultaneously. A novel evolutionary algorithm, namely the Multi-Objective Water Cycle Algorithm (MOWCA), is employed to solve this NP-hard problem. In order to evaluate the proposed algorithm, the Non-Dominated Sorting Genetic Algorithm (NSGA-II) is applied, too. A set of instances is selected from the digital library of project scheduling problems to analyse the performances of evolutionary algorithms. The results of the experimentation are quite satisfactory.*

Keywords: *resource renting, resource constraint, project scheduling, rental policy, metaheuristic algorithms.*

JEL Classification: C44, E40.

1. Introduction

One of the challenges of project management is the reasonable allocation of the project resources as it can optimise the project performance and the project makespan. However, this allocation might be strewn with difficulties due to the limitation of the resources, known as resource constraints. The impact of resource limitations on prolongation of the project makespan has led to develop the resource-constrained project scheduling models aiming at minimising the project makespan (Khalili et al., 2013). As an extension of resource-constrained project scheduling problem, the Resource Investment Problem (RIP) is developed to optimise the resource project costs (Najafi and Azimi, 2009). In RIP, the resources can be

DOI: 10.24818/18423264/58.1.24.07

purchased at the beginning of a project and are available until the end without considering when they are required and whether they are used only once or throughout the project makespan. Resource costs are also considered as time-independent costs in RIPs. It means that the required resources are purchased at the beginning of the project and, as a result, the costs of resource providing are independent of the duration of the project. However, it is not always economical to purchase resources in the real world, and the necessary resources might only be rented. For instance, human resources and heavy machinery cannot be purchased at the beginning of a project by spending a fixed time-independent costs. In such cases, it is better to rent the necessary resources for a limited period.

Hence, it appears essential to consider the Resource Renting Problem (RRP) to make scheduling problems more realistic. Nubel (2001) introduced the RRP, in which the time-dependent renting cost is also considered when new resources are added to the available project resources in addition to the procurement cost. Solving RRPs specifies the optimal rental policy of a project which in turn determines the rental period of resources. Nubel (2001) employed the branch-and-bound method to solve the RRP by assuming the possibility of delay as well as the minimisation and maximisation of intervals for the prerequisite relationships of activities. After defining a new variable for the RRP, Ballestin (2007) employed a Genetic Algorithm (GA) to solve the model. The GA outperformed the branch-and-bound method in projects with 20 activities. Vandenheede et al. (2016) introduced the extended version of the project RRP by adding the activation cost to the objective function and developed a scatter search to solve the problem. Kerkhove et al. (2017) proposed a new modelling framework for the RRP by considering overtime; therefore, the total cost was considered equal to the costs of procuring and renting resources for use during the normal and overtime. Afshar-Nadjafi et al. (2017) analysed the RRP with tardiness and used the GA and Ant Colony Optimization (ACO) algorithm to solve the problem. Schanbel et al. (2018) presented a problem where only variable costs are considered, and the trade-off between the cost for additional resources and the minimum makespan is reviewed. Siamakmanesh et al. (2022) considered a resource availability cost problem and developed a self-adaptive genetic algorithm to tackle the problem.

Studies such as Arjomand et al. (2020), Nabipoor Afruzi et al. (2020), Rezaei et al. (2020), Servranckx et al. (2022), and Hartmanna and Briskorn (2022) are also among the related research work that attempted to address resource-constrained project scheduling problem.

According to the literature on RRP, the proposed methods were single-objective models aiming to minimise the total costs of resources without considering multiple objectives simultaneously. Moreover, there was only one activity mode; however, an activity may have different modes. This study proposes the multi-mode RRP to minimise the total cost of resources and the project makespan. Metaheuristic algorithms must be employed to solve this NP-hard problem to obtain the optimal resource rental policy. Accordingly, the Non-Dominated Sorting Genetic Algorithm (NSGA-II) and Multi-Objective Water Cycle Algorithm (MOWCA) were used in this study. Considering the two objectives of the problem, non-dominated solutions were obtained by solving the problem. The Pareto solutions of the algorithms can be analysed by evaluating the performance evaluation criteria of these metaheuristic algorithms.

The problem statement and its model are presented in the next section. The two metaheuristic algorithms (NSGA-II and MOWCA) are then introduced along with their steps and common grounds. After that, optimal parameters of these two algorithms are determined. Then problems of different dimensions and cost ratios are solved by using these algorithms. In addition, evaluation criteria are employed to compare the two algorithms in terms of performance and efficiency. Finally, concluding remarks are presented.

2. Problem Statement

This study introduces the multi-mode activity project RRP. The problem assumptions are discussed in the following subsection. The mathematical model is then proposed to minimise the total cost of resources and the project makespan simultaneously.

2.1 Problem Assumptions

The project includes *n* activities, numbered from *1* to *n*. The two dummy activities *0* and *n+1* are then added and regarded as the starting and ending activities of the project. The predecessor set of project activities is denoted by *E*. Each activity can be performed in different modes. The set for possible modes of each activity is shown by M_i. Each mode includes the activity duration and required resources, respectively shown by p_{im} and r_{ikm}. The duration of activities (*1* to *n*) is positive in all modes, and those of the dummy activities (*0* and *n+1*) are zero. The activities require K types of resources, and none are available at the starting point of the project. Therefore, a resource rental plan is needed to determine when and how many resources will be rented at any time during the project. The resource costs include the time-independent procurement cost and time-dependent renting cost indicated by C_k^p and C_k^r, respectively.

To formulate the problem, let us define the decision variables as follows:

x_{imt} : A binary variable where it is one if activity *i* starts in mode *m* at time *t* and zero otherwise

α_{kt} : The number of units of resource *k* which are added at time *t*

ω_{kt}: The number of units of resource *k* which are withdrawn at time *t*

2.2 Mathematical Model

This model includes a multi-mode RRP aiming to minimise the total resource cost and the project makespan.

$$\text{Min} \sum_{k\in \mathrm{R}} C_k^p \sum_{t=0}^{UB} \alpha_{kt} + \sum_{k\in \mathrm{R}} C_k^r \sum_{t=0}^{UB} \sum_{j=0}^{t} (\alpha_{kj} - \omega_{kj}) \tag{1}$$

$$\text{Min} \sum_{t=0}^{UB} t x_{n+1,1,t} \tag{2}$$

$$\sum_{m=1}^{M_i} \sum_{t=0}^{UB} (t + p_{im}) x_{imt} \leq \sum_{m=1}^{M_j} \sum_{t=0}^{UB} t x_{jmt} \qquad ; \forall (i,j) \in E \tag{3}$$

$$\sum_{i=1}^{n} \sum_{m=1}^{M_j} \sum_{j=t-p_{im}+1}^{t} r_{ikm} * x_{imj} \leq \sum_{j=0}^{t} (\alpha_{kj} - \omega_{kj}) \quad ; \ \forall k, \forall t \tag{4}$$

$$\sum_{t=0}^{UB} \alpha_{kt} - \sum_{t=0}^{UB} \omega_{kt} = 0 \qquad ; \forall k \tag{5}$$

$$\sum_{t=0}^{UB} \sum_{m=1}^{M_i} x_{imt} = 1 \qquad ; \forall i \tag{6}$$

$$\alpha_{kt}\ ,\ \ \omega_{kt} \in Z^{+} \qquad ; \forall k\ ,\ \forall t \tag{7}$$

$$x_{imt} \in \{0,1\} \qquad ; \forall t, \forall m, \forall i \tag{8}$$

According to Eqs. 1 and 2, the objective functions aim to minimise the total resource cost and the project makespan, respectively. The resource cost is obtained by adding the resource procurement costs, when new resources are added to the set of available resources, to the renting costs for available resources throughout the project. The project makespan equals the commencing time of the dummy activity *n+1*. In these equations, UB shows the upper bound allocated for project completion. Eq. 3 shows the constraint of precedence relationships and guarantees the feasibility of scheduling. In other words, each activity can start only after the completion of its previous activities. Eq. 4 ensures the availability of sufficient capacity for each resource throughout the project. According to Eq. 5, the resources added to the set of available resources are equal to those withdrawn from the set of available resources during the project. That is to show that every resource added to the set of available resources, must be deleted until the end of the project so that the set of available resources will be empty when the project is completed. Eq. 6 states that every activity must be started only once in one mode. The sets of constraints 7 and 8 denote the domain of the variables.

3. The Solution Algorithms

Two metaheuristic algorithms were employed to solve the problem under study. Therefore, the common grounds for these two algorithms are addressed in this section. Each algorithm is then explained in detail.

3.1 Common Grounds of Metaheuristic Algorithms

- **Solution Structure**

A *2×n+2* matrix is employed here to show the solution structure. Each column of this matrix indicates the activity of the same number shown above the matrix. The first row indicates the modes assigned to each activity. The second row includes integers showing the starting times of the activities. Figure 1 shows how to code a project with four activities, each of which can be performed in three modes. In this case, Activities 0 and 5 are dummies. In this solution, the second activity is performed in mode 3 at time 2.

0	1	2	3	4	5
1	1	3	3	2	1
0	0	2	3	4	8

Figure 1. The solution structure

- **Initial Population Generation**

When the possible number of activity modes is known, one mode is first allocated randomly to each activity. Once a specific mode is allocated to activities, the problem changes from a multi-mode case to a single-mode case. A starting time is then allocated to each activity in an interval ranging from the earliest starting time to the latest starting time in order of activity numbers by complying with precedence constraints.

- **Fitness Function**

In this problem, the objective functions include the project makespan and total resource cost. The project makespan equals the time spent for completing the $(n+1)^{th}$ activity, i.e. the final activity of the project, obtained directly from the solution structure. To determine the total resource cost, the profiles of resources are first obtained to show the required extent of each resource in each project interval. The profiles of resources and costs of procuring and renting resources are taken into account to determine the optimal rental policy for scheduling. According to the optimal rental policy, a resource is not removed from the set of available resources in each interval when the cost of renting idle resources is lower than that of repurchasing those resources. The resource rental policy is employed to determine the times of procuring resources and adding them to the set of available resources and the times of withdrawing resources from the set of available resources. The resulting information allows the determination of the total resource cost of the project.

- **Solution Structure Correction**

The newly obtained solutions are analysed for correction if they are not feasible. First, the activity modes are checked in terms of feasibility. If the mode allocated to an activity is not available, that mode is corrected, and one of the possible modes of that activity is randomly assigned to it. In the next step, the starting times are checked in order of activities that concern primacy and latency. If an activity is scheduled earlier than its own earliest starting time, its starting time changes to the earliest starting time. On the contrary, if an activity is scheduled later than its starting time, the time is corrected and changed to the latest starting time. The precedent activities are also checked. If an activity is scheduled before its precedent activities, its starting time is changed to a time following the completion of those precedent activities.

- **Termination Condition**

The algorithm is terminated after the predetermined maximum iterations are performed. The Pareto solutions obtained in the last iteration are then selected as the final solution.

3.2 Multi-Objective Water Cycle Algorithm

Sadollah et al. (2015) developed Multi-Objective Water Cycle Algorithm (MOWCA) for solving multi-objective optimisation problems. This algorithm is an extension of the single-objective water cycle algorithm which is inspired by the natural water cycle and flows of rivers and streams into the sea to solve multi-objective optimisation problems (Eskandar et al., 2012). The MOWCA is the generalised version of the single-objective water cycle algorithm. The initial population of MOWCA is generated after rain to form the population consisting of the sea, rivers, and streams. Then the evaporation process prevents the early convergence of the local optimal solution by vaporising the water in rivers and streams that flow into the sea. The ranking and crowding distance criteria are employed to determine the sea, rivers, and streams. The crowding distance is then used to allocate a stream to the sea or a river. In each cycle, a new stream or river is generated by considering each stream and the sea with the river allocated to it or each river with the sea. If the new stream or river dominates the previous solution, the new stream or river replaces that solution.

- **Determining the Sea, Rivers, and Streams**

In the single-objective water cycle algorithm, after obtaining the cost of each solution, the population members are sorted by the costs, and the best solution is named the sea. Then, the rivers are chosen from the following best solutions. The number of seas and rivers are collectively shown with N_{sr}. The remaining members of the population are also called streams, and their number is calculated by Eq. 9.

$$N_{Stream} = N_{pop} - N_{sr} \tag{9}$$

In the MOWCA, the cost of each solution cannot be used to order the set. Therefore, the amount of swarm distance is computed, and, hence, the non-dominated solutions are ranked by this distance. Then, based on the ranked population, the seas, rivers, and streams are determined and other non-dominated solutions can be classified as rivers or streams around the sea. As a result, the sea is in the centre and the rivers, flowing into the sea, are around it. The streams can either be flowing into the sea or they can be moving toward the rivers.

In the single-objective algorithm, for locating the streams, the cost of specified solutions is used as the seas and rivers. First, N_{pop} streams are created, then N_{sr} of them with minimum cost values are selected as a sea and rivers, with the best individual as the sea. The rest of the population are considered streams that flow to the sea or specific rivers. Eq. 10 helps determine the number of streams, flown into the sea and the rivers. By the number obtained for the sea, the best cost-wise streams are chosen to be flown into the sea. Then the better streams are assigned to the first river. Subsequently, the streams, assigned to the second river, are specified as well, and so the process continues to allocate all of the streams.

$$NS_n = round\left\{\left|\frac{Cost_n}{\sum_{i=1}^{N_{sr}} Cost_i}\right| \times N_{Stream}\right\} \quad , \quad n = 1, 2, \dots, N_{sr} \tag{10}$$

In the MOWCA, to determine the number of streams assigned to the sea and rivers in Eq.10, the swarm distance replaces the costs. Furthermore, as with the single-objective algorithm, the location of the stream flows is determined such that the largest number of streams are flown into the sea and rivers and the number of the streams flowing into the first river is higher than the number of the streams flowing into the second river. Similarly, fewer streams should be flowing into the subsequent rivers.

- **Generating new Rivers and Streams**

In each iteration of the algorithm, each stream is firstly compared to the sea or river flown into it and is generated by Eqs. 11 & 12 in the form of a new stream. In this algorithm, like the NSGA-II algorithm, after the generation of a stream, a correction is performed to justify the obtained result. In these equations, *C* can have a value between one and two, where two is the best possible value that leads to the creation of newer solutions.

$$\vec{X}_{Stream}^{i+1} = \vec{X}_{Stream}^{i} + \text{rand} \times \text{C} \times (\vec{X}_{River}^{i} - \vec{X}_{Stream}^{i}) \tag{11}$$

$$\vec{X}_{Stream}^{i+1} = \vec{X}_{Stream}^{i} + \text{rand} \times \text{C} \times (\vec{X}_{Sea}^{i} - \vec{X}_{Stream}^{i}) \tag{12}$$

Then each river is compared with the sea, and, according to Eq. 13, a solution is produced in the form of a new river. After the generation of the river, a correction is also done.

$$\vec{X}_{River}^{i+1} = \vec{X}_{River}^{i} + \text{rand} \times \text{C} \times (\vec{X}_{Sea}^{i} - \vec{X}_{River}^{i}) \tag{13}$$

If the new stream succeeds in defeating its connecting river and is a better solution, it will replace the river, and their positions are exchanged. If the new river is better than the sea, it can similarly replace the sea. Then, the river becomes a sea and the sea becomes a river.

- **The Evaporation**

In each population, the rivers and streams must have an acceptable distance to the sea, so the Euclidean distance of each river or stream, flowing into the sea, with the sea, is calculated and compared with the evaporation control parameter (Eq. 14). If the distance is less than the evaporation control parameter, new streams flow into the sea, and the reviewed river or stream evaporates.

$$\left\|\vec{X}_{Sea}^{i} - \vec{X}_{River/Stream}^{i}\right\| < d_{max} \tag{14}$$

Figure 2 summarises the coding sequence of this algorithm.

1) Specification of the parameters N_{pop}‘ N_{sr}‘ d_{max} and the maximum number of the iterations.
2) Specification of the number of streams.
3) Production of the initial population
4) Determination of number of the streams, flowing into each of the rivers and the sea.
5) If the stop condition is not met
  5-1) New streams are created based on the rivers and the sea they flow into, and the target functions of the new streams are recalculated.
    5-1-1) If the new stream dominates the river, it will replace it.
    5-2-2) If the new stream dominates the sea, it will replace it.
  5-2) New rivers are created based on the sea and the target functions of the new rivers are recalculated.
    5-2-1) If the river dominates the sea, it will replace it.
  5-3) For the rivers set, new rivers are produced if the distance between the sea and the rivers is less than the evaporation control parameter
6) Obtaining Pareto answers

Figure 2. MOWCA Pseudo-Code

3.3 Non-Dominated Sorting Genetic Algorithm (NSGA-II)

The Genetic Algorithm (GA) is the method of finding approximate solutions and optimising problems (Deb et al., 2002). The solution structure in this algorithm is shown as a chromosome. The NSGA-II is an extended version of the GA, obtained by adding ranking and crowding distance criteria to the GA. In NSGA-II, the concept of dominance is employed to divide the resulting solutions into different fronts. Then, on each front, the crowding distance of each solution is obtained. The rankings and crowding distances of solutions are employed in each iteration to select better solutions. These solutions are then moved on to the next iteration in which crossover and mutation operations are applied. This cycle continues until the termination condition is met. The non-domination solutions obtained from the last iteration are, in fact, the Pareto solutions to the problem.

- **Crossover**

For each crossover operation, two members of the population are randomly selected as parents. The one-point crossover operator and the two-point crossover operator are then applied to them. As a result, two new children are obtained. Once the resulting population is obtained from the crossover operation, the chromosome correction process is performed to ensure the scheduling feasibility of each solution.

- **Mutation**

In the mutation operation, the chromosomes can be changed by three methods with the same probability. In the first method, change is only applied to the first row of the chromosomes. In other words, the selected activity modes change to one of the possible activity modes. In the second method, only the second row of the chromosomes is changed. This means that the starting time of the selected activities ranges from the earliest starting time to the latest starting time. Both rows are changed in the third method. Said otherwise, the starting times of the selected activities change after changing their activity modes. After the resulting population is obtained from mutation, the chromosome correction process is performed to verify the feasibility of scheduling and ensure compliance with precedence constraints.

4. Computational Results

In this section, a few criteria are applied to evaluate the metaheuristic algorithms. These criteria are employed in the following sections. Moreover, the best control parameters of each algorithm must be determined to analyse the results of metaheuristic algorithms. These parameters are obtained from the Taguchi test. Then 60 different problems are solved by both algorithms. In the final section, the resulting criteria are employed to compare the algorithms in terms of efficiency.

4.1 Evaluation Criteria

Quantitative and qualitative criteria are utilised to evaluate the performance of multi-objective metaheuristic algorithms. Some of these criteria are introduced below:

A) Diversity Measure (DM): This index measures the spread of the Pareto solution set generated by a procedure. The DM can be calculated by Eq. 15 and a larger dispersion is more favourable.

$$DM = \sqrt{(\max f_{1.i} - \min f_{1.i})^2 + (\max f_{2.i} - \min f_{2.i})^2} \tag{15}$$

B) Spacing: This criterion determines the expansion and distance of the resulting solutions through Eq. 16. A smaller spacing shows a shorter distance and a normal dispersion of solutions, and thus is more favourable.

$$S = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} \left(d_i - \bar{d}\right)^2} \tag{16}$$

$$d_i = \min_{k \in Q\,.\; k \neq i} \sum_{m=1}^{k} \left|f_m^i - f_m^k\right| \qquad \bar{d} = \frac{\sum_{i=1}^{|Q|} d_i}{|Q|} \tag{17}$$

In Eq. 17, Q shows the Pareto solutions obtained from the algorithm, and, in Eqs 16 and 17, $|Q|$ indicates the number of Pareto solutions. The resulting distance of this equation differs from the Euclidean distance. The standard deviations of different values are also measured.

C) Mean Ideal Distance (MID): This criterion determines the mean distance of the Pareto solutions from the ideal solution or the origin through Eq. 18. In this case, the distance was measured from the origin. Smaller values of MID show that the solutions are closer to the origin and that the algorithm is more efficient. In this equation, $f_{k.i}$ shows the value of the k^{th} objective function in the i^{th} Pareto solution where n and C_i respectively indicate the number of Pareto solutions and Euclidean distance between each Pareto solution and the origin.

$$MID = \frac{\sum_{i=1}^{n} C_i}{n} \qquad C_i = \sqrt{f_{1.i}^2 + f_{2.i}^2} \tag{18}$$

D) Rate of Achievement to Objectives Simultaneously (RAS): This criterion determines the rate of achievement to the ideal values of objective functions simultaneously through Eq. 19. The smaller values of this criterion indicate the higher efficiency of the algorithm:

$$RAS = \frac{\sum_{i=1}^{n} \left(\frac{|f_{1.i} - f_{1.best}|}{f_{1.total}^{max} - f_{1.total}^{min}} + \frac{|f_{2.i} - f_{2.best}|}{f_{2.total}^{max} - f_{2.total}^{min}} \right)}{n} \tag{19}$$

E) The Number of Pareto Solutions: This criterion displayed by NFS shows the number of Pareto solutions, i.e. the non-dominated solutions obtained from the studied algorithms. The higher the values of this criterion, the higher the efficiency of the algorithm.

F) Algorithm Runtime: This criterion indicates the algorithm runtime required to achieve the final Pareto solution. A shorter runtime shows the higher efficiency of the algorithm.

4.2 Setting of the Algorithm Parameters

Determination of appropriate parameters can affect the quality of algorithms (Shahsavar et al., 2011), and, therefore, the Taguchi method is used in this study to set the parameters. In each algorithm, two series of parameters were obtained for the problems with 10 activities, regarded as small problems, and the problems with 20 activities, regarded as big problems. Each test included five problems, each of which was tested five times. The best solution was then used in the following steps. Diversity, spacing, the number of Pareto solutions, and algorithm runtime are utilised to set the parameters.

- **Determining MOWCA Parameters**

Four control parameters must be configured in the MOWCA, including the number of iterations, the population size, the total number of seas and rivers, and the evaporation control parameter. Table 1 shows the four levels defined for each control factor.

Table 1. MOWCA control factors

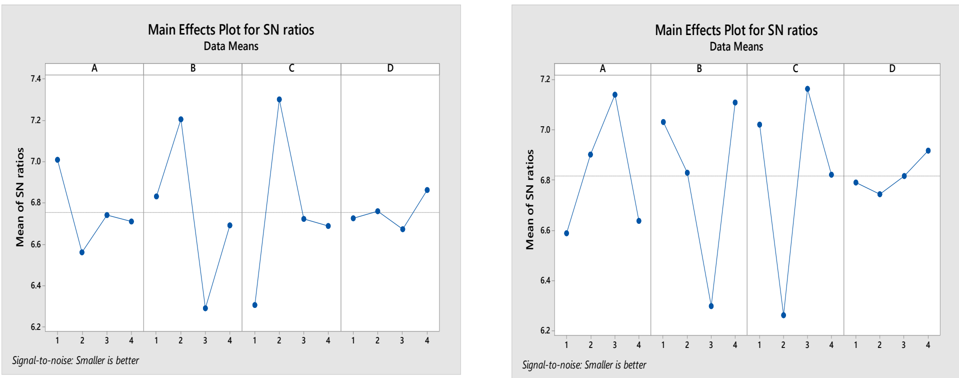
Factor	Symbol	Level 1	Level 2	Level 3	Level 4
MaxIt	A	300	350	400	450
nPop	B	50	70	100	125
Nsr	C	2	4	6	8
dmax	D	0.5	1	1.5	2

Based on the Taguchi method L16b orthogonal array, the experiment was performed for small and large projects separately. First, for each experiment, five problems were randomly chosen. For each one of them, the experiment was performed five times to get better solutions, then the best solution was used for further steps. In the next step, the relative deviation index of the diversity measure, the spacing, the number of Pareto solutions, and the algorithm runtime were all measured by Eq. 20.

$$RDI_i = \left|\frac{Solution_i - Best\ Solution}{Max\ Solution - Min\ Solution}\right| \quad (20)$$

Then, the average relative deviation index, with a weight of 2 for the number of Pareto solutions, and 1 for the rest of the parameters, was obtained.

The outputs are converted into S/N ratios in Minitab, which are shown in Fig. 3, to obtain the best level of each factor in Table 2.



(a) Small size problems (b) large size problems

Figure 3. S/N ratios in MOWCA

Table 2. Optimal levels of MOWCA factors

(a) Small size problems

Factor	***MaxIt***	***nPop***	***Nsr***	***dmax***
Symbol	A	B	C	D
Best Level	1	2	3	4
Best Value	300	75	4	2

(b) Large size problems

Factor	***MaxIt***	***nPop***	***Nsr***	***dmax***
Symbol	A	B	C	D
Best Level	3	4	3	4
Best Value	400	125	6	2

- **Determining NSGA-II Parameters**

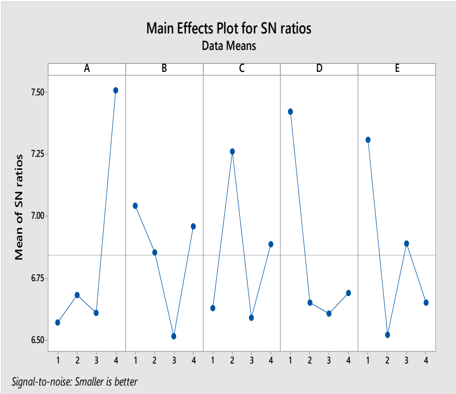
Five control parameters must be set in the NSGA-II, including the number of iterations, the population size, population crossover rate, mutation rate, and mutation effect rate. Four levels were then defined for each control factor. Table 3 shows the parameters set in this algorithm.

Similar to determining MOWCA parameters, the Taguchi method L16b orthogonal array was performed and, then, average relative deviation indexes were obtained for each parameter.

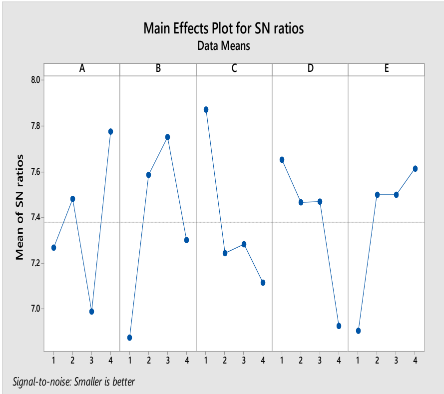
Table 3. Control factors of NSGA-II

Factor	Symbol	Level 1	Level 2	Level 3	Level 4
MaxIt	A	300	350	400	450
nPop	B	50	75	100	125
pc	C	0.4	0.5	0.6	0.7
pm	D	0.1	0.2	0.3	0.4
pmu	E	0.1	0.15	0.2	0.3

The outputs are converted into S/N ratios in Minitab, which are shown in Fig. 4, to obtain the best level of each factor in Table 4.



(a) Small size problems



(b) large size problems

Figure 4. S/N ratios in NSGA-II

Table 4. Optimal levels of NSGA-II factors

(a) Small size problems

Factor	*MaxIt*	*nPop*	*pc*	*pm*	*pmu*
Symbol	A	B	C	D	E
Best Level	4	1	2	1	1
Best Value	450	50	0.5	0.1	0.1

(b) Large size problems

Factor	*MaxIt*	*nPop*	*pc*	*pm*	*pmu*
Symbol	A	B	C	D	E
Best Level	4	3	1	1	4
Best Value	450	100	0.4	0.1	0.3

4.3 Analysis of Metaheuristic Algorithms Results

To compare MOWCA with NSGA-II, 30 problems of 10 activities and 30 problems of 20 activities are tested. The information on these problems is gathered from the PSPLIB website (htttp://www.om-db.wi.tum.de/psplib). In each problem, activities have three different modes and four required resources. A fixed cost of 200 is considered in all problems for procuring resources, whereas the resource rental costs are determined to be 10, 20, and 50% of fixed costs. Diversity, distance from the origin, spacing, RAS, and algorithm runtime are obtained in each test. Table 5 shows the mean and standard deviation of each criterion.

Table 5. Overview of computational outputs

		Mean	SD
DM	NSGA-II	7287.007	5521.973
	MOWCA	7867.67	7917.909
S	NSGA-II	1212.026	1317.434
	MOWCA	948.936	1441.074
RAS	NSGA-II	0.8773	0.1054
	MOWCA	0.93229	0.12886
Time	NSGA-II	70.074	31.8473
	MOWCA	132.7056	78.8

According to the mean values in Table 5, MOWCA outperformed NSGA-II in terms of diversity and showed a higher mean. MOWCA also performs better in terms of spacing. NSGA-II showed a lower RAS and outperformed MOWCA in terms of efficiency. In addition, NSGA-II had a shorter runtime than MOWCA and obtained the Pareto solutions in a shorter time.

In addition, Fig. 5 displays the boxplots of the criteria examined in the tests. These diagrams give a better understanding of the mean, dispersion, and differences of the values obtained from both algorithms in terms of each criterion.

According to the resulting solutions and distances from the ideal point in each test, NSGA-II produced better and closer results to the origin. However, there were relatively fewer solutions, and one or two different scheduling plans were proposed for each cost point. Nonetheless, MOWCA produced more solutions, although Pareto solutions are costlier. In other words, MOWCA solutions showed different starting times and various modes of activities at similar costs.

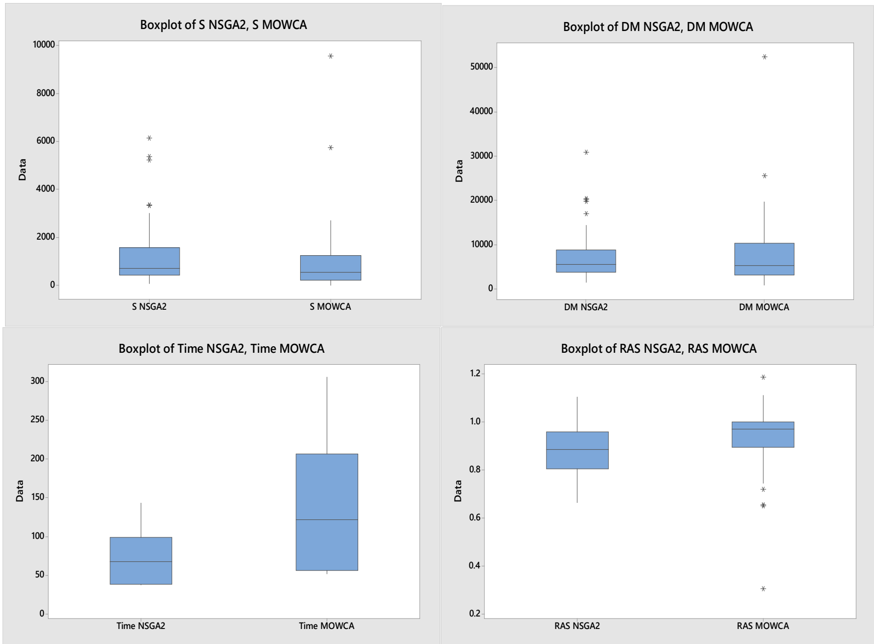


Figure 5. Boxplots of computational outputs

Considering non-normal results, the Mann-Whitney nonparametric test was employed to evaluate the differences in algorithms. The test is conducted on independent samples, and the outputs are combined and sorted in ascending order. The total ranks of each sample are determined and compared separately. If samples have no effects on data, the total ranks of both samples must be the same. Table 6 shows the results obtained from the tests in addition to the superior algorithm for each criterion, indicated in the last columns. The p-value of each test was compared to 0.05. If the P-value<0.5, the null hypothesis concerning the equality of the two samples is rejected; otherwise, the null hypothesis cannot be rejected. Consequently, the samples were not superior to each other, and both algorithms showed a similar efficiency.

Table 6. The results of the Mann-Whitney test

Superior Algorithm	Null Hypothesis: Equality of Two Samples	p-value	Evaluation Criteria
NSGA-II, MOWCA	Not refused	0.8174	DM
NSGA-II, MOWCA	Not refused	0.0756	S
NSGA-II	refused	0.0004	RAS
NSGA-II	refused	0.0000	Time

5. Conclusions

In this study, we introduced a new bi-objective multi-mode resource renting. This two-objective problem aim at minimising the total cost of available resources and the project makespan, simultaneously. The required resources are characterised by time-independent and time-dependent costs. The time-independent cost refers to the cost of procurement of new resources and adding them to the set of available resources in the project. Different modes are also assumed for activities. The optimal rental policy is obtained after solving the problem. This policy determines when resources are procured and added to the set of available resources. After the mathematical model is developed for this problem, two metaheuristic algorithms, namely MOWCA and NSGA-II, are proposed to solve the model. The resulting output indicates that NSGA-II outperformed MOWCA in terms of efficiency; however, MOWCA produced more scheduling plans for a pair of costs.

References

[1] Afshar-Nadjafi, B., Basati, M., Maghsoudlou, H. (2017), *Project scheduling for minimizing temporary availability cost of rental resources and tardiness penalty of activities. Applied Soft Computing,* 61, 536-548.

[2] Arjmand, M., Najafi, A.A., Ebrahimzadeh M. (2020), *Evolutionary algorithms for multi-objective stochastic resource availability cost problem. OPSEARCH,* 57(3), 935-985.

[3] Ballestín, F. (2007), *A Genetic Algorithm for the Resource Renting Problem with Minimum and Maximum Time Lags. Evolutionary Computation in Combinatorial Optimization,* 25-35.

[4] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002), *A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.

[5] Eskandar, H., Sadollah, A., Bahreininejad A., Hamdi, M. (2012), *Water cycle algorithm - a novel metaheuristic optimization method for solving constrained engineering optimization problems. Computers & Structures,* 110-111, 151-160.

[6] Hartmann, S., Briskorn, D. (2022), *An updated survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research,* 297(1), 1-14.

[7] Kerkhove, L., Vanhoucke, M., Maenhout, B. (2017), *On the resource renting problem with overtime. Computers & Industrial Engineering,* 111, 303-319.

[8] Khalili S, Najafi AA, Niaki S.T.A. (2013), *Bi-objective Resource Constrained Project Scheduling Problem with Makespan and Net Present Value Criteria: Two Meta-Heuristic Algorithms, International Journal of Advanced Manufacturing Technology,* 69, 617-626.

[9] Nabipoor Afruzi, E., Aghaie, A., Najafi, A. (2020), *Robust optimization for the resource-constrained multi-project scheduling problem with uncertain activity durations. Scientia Iranica,* 27(1), 361-376.

[10] Najafi, AA., Azimi, F. (2009), *A Priority Rule-Based Heuristic for Resource Investment Project Scheduling Problem with Discounted Cash Flows and Tardiness Penalties. Mathematical Problems in Engineering,* 2009, 1-10.

[11] Nübel, H. (2001), *The resource renting problem subject to temporal constraints. OR-Spektrum*, 23(3), 359-381.

[12] Rezaei, F., Najafi, A.A., Ramezanian, R., (2020), *Mean-conditional value at risk model for the stochastic project scheduling problem, Computers and Industrial Engineering,* 142, 106356.

[13] Sadollah, A., Eskandar, H., Kim, J. (2015), *Water cycle algorithm for solving constrained multi-objective optimization problems. Applied Soft Computing,* 27, 279-298.

[14] Schnabel, A., Kellenbrink, C., Helber, S. (2018), *Profit-oriented scheduling of resource-constrained projects with flexible capacity constraints. Business Research,* 11, 329-356.

[15] Servranckx, T., Coelho, J., Vanhoucke, M. (2022), *Various extensions in resource-constrained project scheduling with alternative subgraphs. International Journal of Production Research*, 60(11), 3501-3520.

[16] Shahsavar M., Najafi A.A., Niaki, S.T.A. (2011), *Statistical Design of Genetic Algorithms for Combinatorial Optimization Problems", Mathematical Problems in Engineering,* 2011, 2-17.

[17] Siamakmanesh, N., Najafi, A.A., Azimi, F. (2022), *A Work-content Based Resource Availability Cost Problem: Mathematical Modeling and Solving Procedure. Economic Computation and Economic Cybernetics Studies and Research,* 56(1), 75-90.

[18] Vandenheede, L., Vanhoucke, M., Maenhout, B. (2016), *A scatter search for the extended resource renting problem. International Journal of Production Research,* 54(16), 4723-4743.