Journal of Industrial and Systems Engineering
Vol. 15, No. 4, pp. 65-90
Autumn (November) 2023

JISE

Optimization of Project Scheduling Problem with Limited Resources and uncertainty using the Intelligent Water Drop

Asghar Hemmati[1], Amirhosein Emami[2], Adel Pourghader Chobar[3*], Raheleh Alamiparvin[4], Amin Saeidi Khasraghi[5]

[1]*Department of Industrial Engineering, Abhar Branch, Islamic Azad University, Abhar, Iran*
[2]*Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran,*
[3]*Department of Industrial Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[4]*Department of Industrial Engineering, Bonab Branch, Islamic Azad University, Bonab, Iran*
[5]*Department of Industrial Engineering, Eyvanekey University, Semnan, Iran*

Abstract

Nowadays, the project scheduling problem with limited resources has become one of the most important optimization issues. Given the enormous costs spent on projects as well as materials, resources, and forces, projects are expected to be successful and finish at the planned time. This schedule in functional mode will have many uncertainties due to the needs of the moment. In this research, a new method is presented using the Intelligent Water Drops algorithm for the resource-constrained project scheduling problem. For this reason and because of the importance of these projects, in this research, an optimization model has been developed for project scheduling in the state of uncertainty, which can solve many implementation obstacles. For this purpose, first, the problem is formulated as a mixed-inter linear programming model. Next, the model is optimized using the IDW algorithm. To evaluate the performance of the proposed method, the standard data set was used in previous research and articles, and four datasets with different scales were selected from the PSLIB library. The results show that the proposed method is capable of obtaining the best precision in terms of the least critical deviation from the optimal solution. Moreover, the results of the proposed method were compared with metaheuristic algorithms, such as the particle congestion algorithm, which was able to get the best solution among these algorithms.

Keywords: Resource-constrained project scheduling; Intelligent water drop algorithm; Particle Swarm Optimization; Metaheuristic algorithms; Optimization.

1. Introduction

Today, most of the activities that are implemented to produce goods or provide services in organizations are defined as projects. However, many of these projects are delayed, lost, or

* Corresponding Author
ISSN: 1735-8272, Copyright c 2023 JISE. All rights reserved

mismatched with quality standards. Therefore, it is important to pay attention to project management techniques and methods. The history of project management concepts and methods dates back to more than half a century ago, during which time a variety of definitions of project management have been presented. American National Standards Institute (ANSI)[†] defines project management (Kolisch, 2013; Bathaee et al. 2023): as "Applying knowledge, skills, techniques, and tools for project activities to meet project needs."

Project management and project schedules are the main axes of theoretical and practical research in the field of operations research. Project scheduling seeks to balance and balance the project's competitive and contrasting goals, quality, time, and cost by maintaining the project domain. The three scheduling aspects are shown in Figure (1); therefore, the balance between them has been the subject of many studies and articles (Fallah et al., 2021).

Figure 1. Triple Goals of Projects (Sonmez & Bettemir, 2012).

Resource-Constrained Project Scheduling Problem (RCPSP) with Multi-Mode is the development mode of project scheduling issues. In fact, a class of challenging optimization issues that have been increasingly attracted to the scientific community in recent years, assuming resources can have more than one species (Correia et al., 2012; Tavana et al., 2014; Fallah et al. 2021). RCPSP is modeled on scheduling projects, taking into account some constraints and variables. Project time is often the purpose of project scheduling, while other goals, such as minimizing the total cost of the project, high quality of the project quality, and minimizing delays, can be considered (MA et al., 2016; Mehrani et al. 2020).

[†] The American National Standards Institute is a private nonprofit organization that oversees the development of voluntary consensus standards for products, services, processes, systems, and personnel in the United States

Traditional methods, such as the Critical Path Method (CPM) and Program Evaluation & Review Technique (PERT)[‡], consider infinitive resources. In other words, these methods do not pay attention to resource constraints and costs (Oztemel & Selm, 2012; Ashoka, and Keihani, 2020). Therefore, the project schedule may not result from an application program. Therefore, it is necessary to revise the project schedule.

On the other hand, the optimization methods have led to extensive use of scheduling problems. In this regard, optimization methods can be used as one of the best tools for finding the best project schedules. Considering numerous variables and constraints, RCPSP is considered to be a non-deterministic polynomial timed (NP-Hard) problem in management and research in operations. Therefore, as the number of variables and constraints increases, the time for problem-solving and the memory space it occupies increases (Balouka et al., 2016; Aliahmadi et al. 2016).

Early views on the problem of resource allocation in project control based on optimal solutions from mathematical models such as Integer Programming (IP) require cost and time. Therefore, in the case of major issues, exact optimization algorithms are not able to solve the problem at a reasonable time. Therefore, the use and development of approximate algorithms (meta-heuristics) to solve the PRCPS (Van Eynde & Vanhoucke, 2022, Ghahremani-Nahr et al., 2023, Movahed et al., 2023).

The Intelligent Water Drops (IWD) algorithm is one of the novel meta-heuristic algorithms that Shah-Hosseini (2007) presented. IWD is a population-based algorithm inspired by the mechanism of water dripping in rivers that face many barriers to lakes and oceans but eventually find their way to their destination (Shah- Hosseini, 2007). The idea of this algorithm is based on water drops that flow in nature so that each drop makes a solution with the problem's search space and changes the environment around it (Shah-Hosseini, 2007).

Therefore, as can be seen, resource limitations and uncertainties are among the most important challenges of project scheduling, especially in infrastructure and basic projects. Therefore, designing an optimal model and tool for checking and evaluating the timing of these projects is of

[‡] The program evaluation and review technique is a statistical tool used in project management, which was designed to analyze and represent the tasks involved in completing a given project.

fundamental importance. Because it reduces costs, optimizes time and develops development projects. In this study, it is attempted to model and solve the project scheduling problem with limited resources by using the intelligent water drops algorithm. In this regard, a two-objective mathematical model is proposed and optimized with the IWD algorithm. This algorithm tries simultaneously to minimize the total completion time and cost of the project.

2. Research Background

Myszkowski et al. (2015) solved the project scheduling problem with multi-mode resources using a hybrid Ant Colony Optimization (ACO) algorithm. They presented a hybrid approach that is modeled by classic Heuristic Priority Rules (HPR) for project scheduling. In addition, in the proposed ACO algorithm, a new way to update the pheromone value based on the best and worst solutions was applied. The results showed that the hybrid approach based on the ACO algorithm is efficient in optimizing RCPSP.

In another study, Maghsoudlou et al. (2016) presented a multi-objective weed optimization algorithm (MOIWO) to solve RCPSP. The main goals of that study were minimizing the competition time, the total cost, and the total quality of the project. In order to evaluate the performance of IWO, the results of this algorithm were compared with the Non-Dominated Sorting Genetic Algorithm (NSGA-II). The results of this research showed that the MOIWO performs better in the diversification of non-dominated solutions.

Zheng et al. (2017) used the Teaching-Larning-Based Optimization (TLBO) algorithm to solve the project scheduling problem with limited resources in multi-task mode. This article presented a resource-based program programming with a work list and resource list. To achieve satisfactory functions, the balance between general and local search is emphasized in the design of the Teaching Learning Based Optimization (TLBO) algorithm. The performance of the proposed balanced resource law has been shown in comparison with several heuristic algorithms. The comparative comparison between the TLBO and the existing algorithm showed the effectiveness and efficiency of the proposed method.

Kadri and Boctor (2018) used a Genetic Algorithm (GA) to solve RCPSP concerning transfer time. They assumed that the duration and time of the transfer of activities are recognized and determined.

In this study, the goal was to choose the starting time to start any project activity to minimize priority relationships, availability of resources, and restrictions on resource transfer time. For this purpose, a new genetic algorithm was proposed using a two-point crossover. The tests of this study, which were carried out in a large number of cases, showed that the proposed algorithm has a better performance than several other solution methods.

Ashadi et al. (2021) presented the project scheduling model with limited resources using innovative processing algorithms. In this study, the Monte-Carlo algorithm was introduced, implemented, and shown a numerical example that is more efficient in estimating activities than the PERT method.

Afshar et al. (2022) investigated the multi-resource-constrained project scheduling problem (MRCPSP) given uncertainty during activities and delays. The development of the above model for construction projects is important because of the non-existent space for holding non-renewable resources. In this research, fuzzy logic was used to display the time uncertainty. This article examined the project schedule by providing an intelligent algorithm combined with fuzzy sets and genetic algorithms (GA).

In another study, Javanmard et al. (2022) expanded the problem of scheduling the multi-capacity project with limited resources over time and presenting the Harmoney Search (HS) algorithm. The access to employees during the project planning horizon is not fixed due to official holidays, weekends, or illness. Therefore, in this article, a mathematical model was proposed for the multi-shift project scheduling problem, where the amount of access to resources is variable and time-dependent. The relationship between the activities in the proposed model is a generalized type. The purpose of the proposed model is to minimize the project time. To solve the proposed model, including NP-Hard issues, a meta-heuristic algorithm based on the HS algorithm was developed. In this regard, two intersection operators and new mutations are designed to increase the diversity of solutions and reduce the similar solutions. The proposed algorithm's performance in solving a few sample problems was compared to PSO and GA. The results show the proposed algorithm's superiority in terms of the solution quality and the problem of problem-solving.

3. Intelligent Water Drops (IWD) Algorithm

IWD algorithm is designed based on the intelligent movement of water drops in rivers, lakes, and seas. The intelligence of the drops in the rivers is straightforward; Because, despite the various barriers to the river flow (the collection of drops), they find their way to lakes, seas, and oceans. The intelligent water drops algorithm is a population-based method that water drops together can build appropriate solutions (Shah-Hosseini, 2007). The structure of this algorithm is described in the following sub-sections.

3.1. IWD Rules

The IDW algorithm is based on several rules and general assumptions (shah- Hosseini, 2007), which are presented as follows.

The first rule: Each intelligent water drop (IWD) carries some soil and transfers it from one point of the river to another part.

The second rule: Each IWD has a speed that plays an important role in removing the soil from the river surface. Considering the two drops of water that initially have the same soil and move from one point to another, it is assumed that a faster drop collects more soil than the river surface when it reaches the next point of the route. The drop rate determines the amount of soil harvesting from the river surface.

The third rule: The speed of an IWD is affected by the extent of the route soil. The route increases water speed with a small amount of soil than the route with a significant amount. Therefore, according to rules (2) and (3), the route with low soil to IWD allows more soil collection and more quickly, while the route is more resistant to water drops, the IWD will accumulate less and increase speeds.

The fourth rule: A drop of water makes the path easier to move. This hardness in this algorithm is equivalent to the amount of route soil. Therefore, each IWD, in the face of a few branches in its path, will prefer less soil to other branches.

In nature, numerous water drops combine to build an optimal route to reach the destination. In other words, a population-based intelligent mechanism is applied in the IWD algorithm. Moreover, this algorithm uses this mechanism using a population of water drip to build different paths, and the optimal or closest route gradually emerges over time.

3.2. Implementation of IWD Rules

In this algorithm, it is assumed that each IWD carries some soil, and its current speed is velocity. In addition, it is assumed that IWDs flow in a discrete environment. This environment can be considered composed of Nc nodes. Each IWD needs to move from one node to another in this environment to make its solution. Both nodes are connected by a ridge that contains some soil. Based on the passage of IWDs, the soil of each ridge may increase or decrease. By designing relations, Shah-Hosseini (2007) has realized the necessary rules for implementing the IWD algorithm.

• Implementation of the third rule:

Assuming that the speed of an IWD at node i is equal to vel^{IWD} and the amount of soil is equal to $sol\ (i,j)$, the speed of IWD movement from node i to node j is calculated by Eq. (1).

$$vel^{IWD}\,(t+1) = vel^{IWD}\,(t) + \frac{a_v}{b_v + c_v * soil^2(i,j)} \tag{1}$$

In this regard, $vel^{IWD}\,(t+1)$ is the IWD speed at node j, a_v, b_v, and c_v are fixed parameters that are adjusted according to the problem. It should be noted that in order to prevent the IWD speed from becoming negative, the power of 2 is considered for the term $sol(i,j)$.

• Implementation of the second rule:

The time required for a drop to move with speed $vel^{IWD}\,(t+1)$ from node i to node j is calculated by Eq. (2).

$$time\big(i,j; vel^{IWD}\,(t+1)\big) = \frac{HUD(i,j)}{max\big(\varepsilon, vel^{IWD}\,(t+1)\big)} \tag{2}$$

where ε is a small value that is used to prevent the denominator of Eq. (2) from becoming zero, moreover, HUD is a local revelation function that shows the level of dissatisfaction in the movement of the droplet from node i to node j.

• Implementation of the first rule:

Each IWD, by moving from one node to another node, washes some of the soil of the route and adds to the soil it carries. The amount of soil that moves from route (i,j) is calculated by Eq. (3).

$$\Delta soil(i,j) = \frac{a_s}{b_s + c_s * \left(time^2\left(i,j; vel^{IWD}(t+1)\right)\right)} \tag{3}$$

where $\Delta soil(i,j)$is the amount of transferred soil from route (i,j) and $c_s, b_s, a_s$are fixed parameters that are adjusted according to the problem. The value of $\Delta soil(i,j)$ is used to calculate the new amount of soil which is presented in Eq. (4).

$$soil^{IWD} = soil^{IWD} + \Delta soil(i,j) \tag{4}$$

Moreover, the amount of remaining soil in the route (i,j) is obtained by the Eq. (5) known as the local updating relation.

$$soil(i,j) = \rho_0 . soil(i,j) - \rho_n . \Delta soil(i,j) \tag{5}$$

In this regard, $\rho_0$and $\rho_n$are positive numbers that are chosen in the interval [0,1].

• Implementation of the fourth rule:

The probability of selecting the next node (j) for an IWD located in node i is obtained by Eq (6).

$$p_i^{IWD}(j) = \frac{f\left(soil(i,j)\right)}{\sum_{k \notin vc(IWD)} f\left(soil(i,k)\right)} \tag{6}$$

In this regard, $p_i^{IWD}(j)$ calculates the probability of IWD moving from node i to node j, which is not in the $vc(IWD)$ list. The mentioned nodes have not been met by IWD so far and do not violate the constraints of the problem. $f\left(soil(i,j)\right)$ is a function obtained by using Eq. (7) and ε_s is a small positive constant value that prevents division by zero.

$$f\left(soil(i,j)\right) = \frac{1}{\varepsilon_s + g\left(soil(i,j)\right)} \tag{7}$$

In the IWD algorithm, since each drop of water washes some of its soil by passing through the route (i,j), the amount of soil in the route may become negative (although there is no such situation in nature). Therefore, in Eq. (8), $g\big(soil(i,j)\big)$ shifts the values of $soil(i,j)$ to positive values if the minimum amount of soil in the allowed ridges is negative.

$$g\big(soil(i,j)\big) = \begin{cases} soil(i,j) & if \min\limits_{l \notin vc(IWD)} \big(soil(i,l)\big) \geq 0 \\ soil(i,j) - \min\limits_{l \notin vc(IWD)} \big(soil(i,l)\big) & Otherwise \end{cases} \tag{8}$$

3.3. The Evolution Path in IWD Algorithm

Each water drop created in the IWD algorithm moves from its initial node to the following nodes until it completes its corresponding solution. For each optimization problem, an objective or quality function is defined, so the quality of the solution of an IWD is represented by the function $q(T^{IWD})$. An iteration of the algorithm ends when all IWDs have created their respective solutions. After the end of each iteration, the best solution of that iteration (T^{IB}) is obtained by Eq. (9).

$$T^{IB} = arg \max_{\forall T^{IWD}} \{fitness(T^{IWD})\} \tag{9}$$

After determining T^{IB}, the route in these solutions is updated according to the amount of soil collected by the IWD manufacturer with Eq. (10), known as the global update equation.

$$soil(i,j) = \rho_s soil(i,j) + \rho_{IWD}\, K(N_c) soil_{IB}^{IWD} \quad \forall (i,j) \in T^{IB} \tag{10}$$

In this regard, $\rho_s$is a positive constant number from the interval [0,1] and $\rho_{IWD}$is a negative constant number from the interval [-1,0]. $soil_{IB}^{IWD}$is the amount of soil collected by IWD creating the path (solution), N_C is the total number of nodes in the search space of the desired problem and $K(N_C)$ is a positive coefficient related to $N_C$and is usually considered equal to $\frac{1}{N_C - 1}$. The purpose of using global updating is that the best solution of each iteration is gradually reinforced, and IWDs are driven to search for appropriate solutions in the hope of finding the global-optimal solution.

Moreover, at the end of each iteration of the algorithm, the best solution found so far (T^{IB}) is updated with Eq. (11) to guarantee the maintenance of the best solution.

$$T^{IB} = \begin{cases} T^{TB} & if q(T^{TB}) \geq q(T^{IB}) \\ T^{IB} & Otherwise \end{cases} \tag{11}$$

4. Proposed Mathematical Model

In this section, a proposed model for solving the project scheduling problem with limited resources is presented. First, the assumptions of the problem in this research are presented:

• The problem of determining and scheduling the activities is multi-mode.

• After starting any activity, it is not allowed to stop it.

• The time required to perform each activity is definite.

• There is no need for preparation time to perform the activities.

• The capacity of resources is limited and specific.

• Whenever the prerequisite of an activity has been completed, that activity is executable

• Prerequisite relations of activities are Generalized Precedence Relations (GPR).

4.1. The Decision Variable of the Proposed Model

The decision variable of the proposed mathematical model is a binary one which is shown as x_{imt}. If the activity i starts in the execution mode m at time t, it takes a value of 1, otherwise, it takes a value of zero. This variable is defined as Eq. (12):

$$x_{imt} \in \{0,1\} \quad i = 1,2,\ldots,n.,m = 1,2,\ldots,M_i.,t = es_i,\ldots,Is_i \tag{12}$$

4.2. The parameters of the Proposed Model

The parameters of this model are as follows:

i	Index of activity
M_i	Set of execution modes for activity i
A	set of activities
q_{im}	The quality of performing activity i in execution mode m

L_{si}	The latest time to start my activity
E_{si}	The earliest time to start my activity
C_{im}	The cost of performing activity i in execution mode m
E_{SS}	Start-start set of prerequisite relationships
E_{SF}	Start-finish set of prerequisite relationships
E_{FS}	Finish-start set of prerequisite relationships
E_{FF}	Finish-to-finish set of prerequisite relationships
SS_{ij}	minimum or maximum delay time of the start state - the start of activity i,j
SF_{ij}	minimum or maximum delay time of the start-finish state of activity i,j
FS_{ij}	minimum or maximum delay time of the finish state - the start of activity i,j
FF_{ij}	minimum or maximum delay time of the finish-finish state of activity i,j
d_{im}	Time to perform activity i in execution mode m
r_{imk}	the amount of consumption of activity i in execution mode m from the renewable source of type k
a_k	Access level of renewable resources of type K in each period
r_{imk}^{∂}	the amount of consumption of activity i in execution mode m from the non-renewable resource of type k
a_k^{∂}	The level of access to the non-renewable resource of type K in the whole project

4.3. Objective Functions

According to the research conducted, including Tirkolaee et al. (2019) and Goli et al. (2023), scheduling should address the most important goal, i.e., minimizing the completion time. In this research, the presented model is defined as a two-objective completion time minimization and project cost minimization. The first objectives of this research are defined as Eq. (13):

$$Min\, f = (f_1, f_2) \tag{13}$$

In this model, 2 objective functions are considered: the first objective in Eq. (14) is to minimize the project's completion time.

$$Min\, f_1 = \sum_{t=es_n}^{ls_n} t.x_{n1t} \tag{14}$$

The second objective in Eq. (15) is to minimize the total cost of the project.

$$Min\, f_2 = \sum_{i \in A} \sum_{m \in M} \sum_{t=es_i}^{ls_i} c_{im} \times X_{imt} \tag{15}$$

4.4. Constraints of the Proposed Mathematical Model

There are various constraints in the proposed mathematical model, which are categorized as follows:

• Start time constraint

Constraint (16) specifies that each activity can be implemented in at most one state. In fact, this constraint works in such a way that if an activity has different execution modes and an execution mode is selected, this activity can only be performed in the same mode and must be performed in this way until the end of the activity. It cannot be changed to other modes while performing.

$$\sum_{m=1}^{M_i} \sum_{t=es_i}^{ls_i} X_{imt} = 1 \tag{16}$$

• Prerequisite relationships constraints

In this proposed model, all general prerequisite relationships are considered. Constraints (17)-(20) are the constraints of prerequisite relationships. Constraint (17) is the constraint of the start-to-start relationship. Constraint (18) is a prerequisite constraint from start-to-finish activities. Constraint (19) is the most famous constraint, and in all RCPSP problems with partial prerequisite relations, there is only this prerequisite constraint. This equation is used for finish-to-start activities. Constraint (20) is a prerequisite finish-to-finish relationship that will work for activities that have such a relationship between them.

$$\sum_{m=1}^{M_i}\sum_{t=es_i}^{ls_i}(t+ss_{ij})X_{imt} \le \sum_{m=1}^{M_j}\sum_{t=es_j}^{ls_j} t.x_{jmt} \qquad \forall (i,j)\in E_{ss} \tag{17}$$

$$\sum_{m=1}^{M_i}\sum_{t=es_i}^{ls_i}(t+SF_{ij})X_{imt} \le \sum_{m=1}^{M_j}\sum_{t=es_j}^{ls_j}(t+d_{jm})x_{jmt} \qquad \forall (i,j)\in E_{sf} \tag{18}$$

$$\sum_{m=1}^{M_i}\sum_{t=es_i}^{ls_i}(t+d_{im_i}+FS_{ij})X_{im_it} \le \sum_{m=1}^{M_j}\sum_{t=es_j}^{ls_j} tx_{jm_it} \qquad \forall (i,j)\in E_{fs} \tag{19}$$

$$\sum_{m=1}^{M_i}\sum_{t=es_i}^{ls_i}(t+d_{im_i}+FF_{ij})X_{imt} \le \sum_{m=1}^{M_j}\sum_{t=es_j}^{ls_j}(t+d_{jm_i})x_{jmt} \qquad \forall (i,j) \in E_{ff} \tag{20}$$

• Renewable resources constraint

Constraint (21) is the limitation of renewable resources, which is also seen in the fundamental RCPSP. This constraint includes all the resources that can be used at the maximum rate in each period. Renewable resources, such as workforce, equipment, etc., are all included in this constraint.

$$\sum_{i=1}^{n}\sum_{m=1}^{M_i} r_{imk} \sum_{s=max\{t-d_{im},es_i\}}^{min\{t-1,ls_i\}} \leq \alpha_k \quad k=1,\dots,k, t=1,\dots,T \tag{21}$$

• Non-renewable resources constraint

Constraint (22) is the limitation of non-renewable resources, whose total amounts are known at the beginning of the project, and this amount gradually decreases with their consumption. Resources such as the project budget, all kinds of necessities and consumables, etc., are such materials. Since the cost of performing the activities is minimized through the second objective function, and the project budget can also be considered a type of non-renewable resource; Therefore, bringing a new limit to avoid spending extra money has been omitted, and the project budget limit is included in constraint (23).

$$\sum_{i=1}^{n}\sum_{m=1}^{M_i} r_{imk}^{\vartheta} \sum_{s=es_i}^{ls_i} x_{ims} \leq \alpha_k^{\vartheta} \quad k=1,\dots,K \tag{22}$$

$$\sum_{m=1}^{M} q_{im} \sum_{t=es_i}^{ls_i} x_{imt} \geq \sigma_i \quad i=1,\dots,n \tag{23}$$

5. Numerical Results

To create sample test problems for the proposed model, the PSPLIB standard problems have been used. In this case, multi-mode RCPSP problems are used, the number of modes of performing each activity and the time of performing each activity in that mode are available, and for the only change made in the amount of resources, the amount of resources for this new problem is equal to 2. A laptop performed all the tests with an Intel Core i7 and 6 GB of RAM under the Windows 10 operating system. The IWD algorithm is coded and implemented in MATLAB R2020a.

5.1. Research Data Set

In order to show the correctness and efficiency of the proposed method, it is necessary to solve the existing test problems with the help of the proposed algorithm and compare its performance with other problem-solving methods. The standard data set used in this research includes PSLIB online project scheduling library, in which Single Resource Constraint Project Scheduling Problem (SRCPSP) section is selected to evaluate the proposed method. This section contains four subsets of problems named j30, j60, j90, and j120, where the letter after j indicates the number of project activities. Problems j30, j60, j90, and j120 have 480, 480, 480, and 600 examples (test problems), respectively. These samples were produced by the ProGen standard project generation engine by Kolisch (1997).

Shah Hosseini (2009) has suggested values for the IWD algorithm, and these values are shown in Table (1). Accordingly, these values are used in all PSBLIB test problems optimization.

Table 1. Parameters of the intelligent water drop algorithm.

Parameter	**Symbol**	**Selected value**
The number of drops of intelligent water	N_{IWD}	Number of activities
Schedule number	$iteration$	100
The first parameter is speed update	a_v	1
The second parameter of speed update	b_v	0.01
The third parameter of speed update	c_v	1
The first parameter of soil renewal	a_s	1
The second parameter of soil renewal	b_s	0.01
The third parameter of soil renewal	c_s	1
The initial amount of soil	$initSoil$	1000
The initial value of the speed	$initVel$	100
Local parameter of soil renewal	ρ_n	0.9

Global parameter of soil renewal	ρ_{IWD}	0.9

In order to display the results of the proposed method for each test file from the data set in the program's output, first, it is specified by the file's name, and then the best solution obtained for it (Makespan) is shown. Then the standard deviation obtained from the optimal solution (Error) and the optimal solution (Optimal) as well as the execution time of the proposed method for this file (CPU_Time) will be shown. Figure (2) shows the output sample of the program for the first 30 examples of the j30 set that were used in the initial tests.

```
#1 j301-1: MakeSpan = 45 Error = 0. 046512 Optimal = 43 CPU_Time = 67. 157

#2 j301-2: MakeSpan = 51 Error = 0. 085106 Optimal = 47 CPU_Time = 61. 8956

#3 j301-3: MakeSpan = 47 Error = 0 Optimal = 47 CPU_Time = 5. 7499

#4 j301-4: MakeSpan = 62 Error = 0 Optimal = 62 CPU_Time = 5. 7961

#5 j301-5: MakeSpan = 40 Error = 0. 025641 Optimal = 39 CPU_Time = 59. 2867

#6 j301-6: MakeSpan = 48 Error = 0 Optimal = 48 CPU_Time = 11. 4207

#7 j301-7: MakeSpan = 60 Error = 0 Optimal = 60 CPU_Time = 6. 1639

#8 j301-8: MakeSpan = 53 Error = 0 Optimal = 53 CPU_Time = 36. 6823

#9 j301-9: MakeSpan = 50 Error = 0. 020408 Optimal = 49 CPU_Time = 60. 3655

#10 j301-10: MakeSpan = 46 Error = 0. 022222 Optimal = 45 CPU_Time = 63. 2045

#11 j302-1: MakeSpan = 38 Error = 0 Optimal = 38 CPU_Time = 7. 1744

#12 j302-2: MakeSpan = 51 Error = 0 Optimal = 51 CPU_Time = 6. 4652

#13 j302-3: MakeSpan = 43 Error = 0 Optimal = 43 CPU_Time = 6. 511

#14 j302-4: MakeSpan = 43 Error = 0 Optimal = 43 CPU_Time = 6. 5009

#15 j302-5: MakeSpan = 51 Error = 0 Optimal = 51 CPU_Time = 6. 4601

#16 j302-6: MakeSpan = 47 Error = 0 Optimal = 47 CPU_Time = 6. 3108

#17 j302-7: MakeSpan = 47 Error = 0 Optimal = 47 CPU_Time = 6. 3638
```

#18 j302-8: MakeSpan = 54 Error = 0 Optimal = 54 CPU_Time = 6. 6009

#19 j302-9: MakeSpan = 54 Error = 0 Optimal = 54 CPU_Time = 6. 2486

#20 j302-10: MakeSpan = 45 Error = 0. 046512 Optimal = 43 CPU_Time = 68. 2465

#21 j303-1: MakeSpan = 72 Error = 0 Optimal = 72 CPU_Time = 8. 8111

#22 j303-2: MakeSpan = 40 Error = 0 Optimal = 40 CPU_Time = 6. 3795

#23 j303-3: MakeSpan = 57 Error = 0 Optimal = 57 CPU_Time = 6. 4865

#24 j303-4: MakeSpan = 98 Error = 0 Optimal = 98 CPU_Time = 6. 2709

#25 j303-5: MakeSpan = 53 Error = 0 Optimal = 53 CPU_Time = 6. 7161

#26 j303-6: MakeSpan = 54 Error = 0 Optimal = 54 CPU_Time = 6. 7619

#27 j303-7: MakeSpan = 48 Error = 0 Optimal = 48 CPU_Time = 6. 2636

#28 j303-8: MakeSpan = 54 Error = 0 Optimal = 54 CPU_Time = 6. 4475

#29 j303-9: MakeSpan = 65 Error = 0 Optimal = 65 CPU_Time = 6. 1324

#30 j303-10: MakeSpan = 59 Error = 0 Optimal = 59 CPU_Time = 6. 1941

Number of Optimal Solutions: 24

Average Deviation from Optimal Solution: 0. 0082134

Success Rate: 80

Average CPU Time: 19. 0356

Figure 2. Program output sample for the first 30 examples of the j30 collection.

For example, the convergence of the proposed method for problem j301-1 is shown in Figure (3). As shown in this Figure, the algorithm has reached convergence in step 39 of 100 iterations.

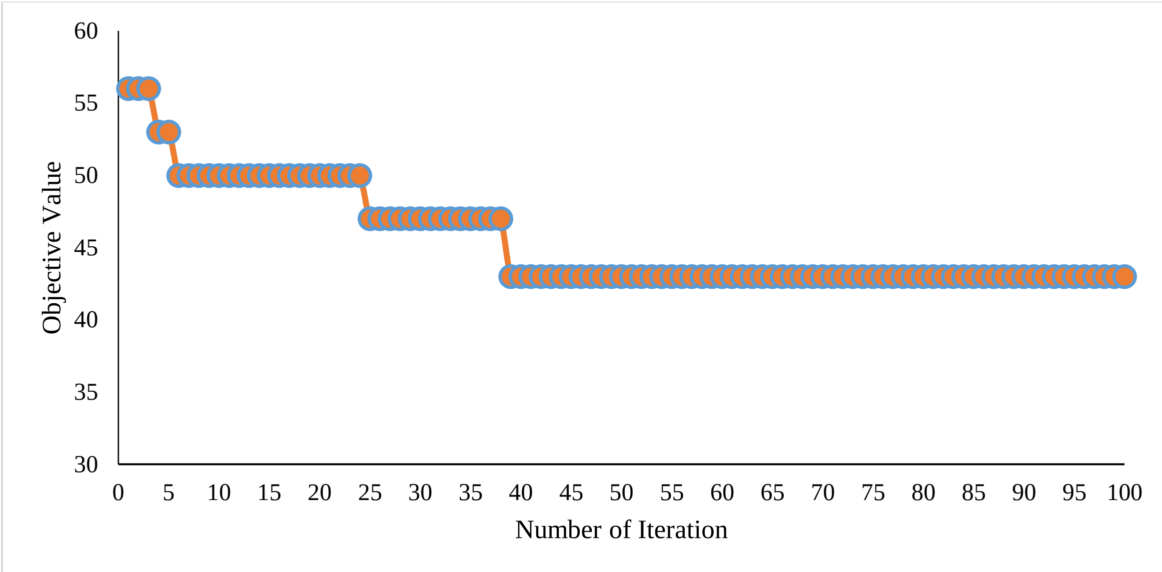

Figure 3. Convergence of the proposed meta-heuristic algorithm for problem j301-1.

The results of the proposed method can also be shown by two graphs of scheduling and interest in resources. As an example, the following two graphs are shown for the first example of all 4 sets of j30, j60, j90, and j120. Figure (4) shows how to schedule problem j301_1. In this example, 30 activities should be scheduled on 4 resources in such a way that the best possible situation is established. The result of the proposed method for this problem is shown in this Figure. As shown in this Figure, at first, the standard deviation is about 46, which decreases over time until it converges to the value of 43. Moreover, Figure (5) shows the efficiency of resources in time for this problem by the proposed method.

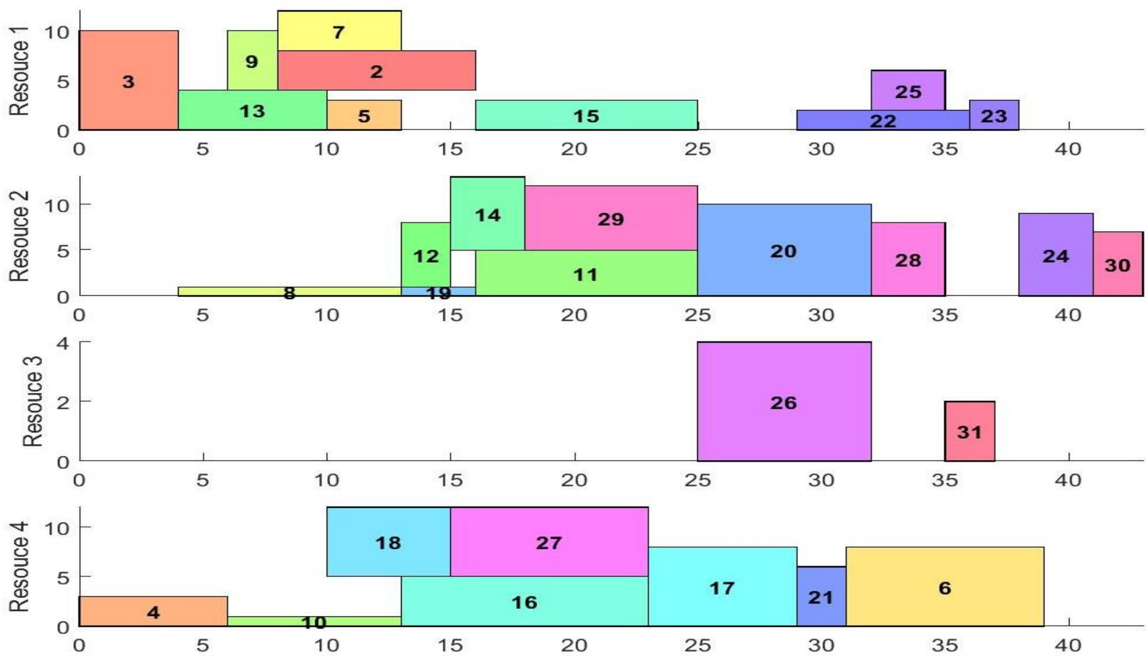

Figure 4. Scheduling of problem j301_1 by the proposed method.

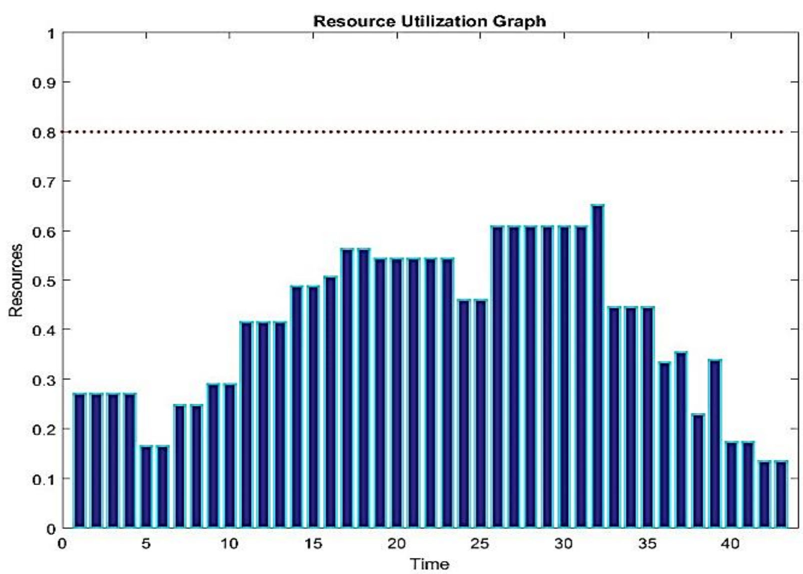

Figure 5. Resource productivity graph in time for problem j301_1 by the proposed method.

Similarly, Figure (6) shows how to schedule problem j601_1. In this example, 60 activities should be scheduled on 4 resources in such a way that the best possible situation is established. The result of the proposed method for this problem is shown in this Figure. Moreover, Figure (7) shows the efficiency of resources in time for this problem by the proposed method.

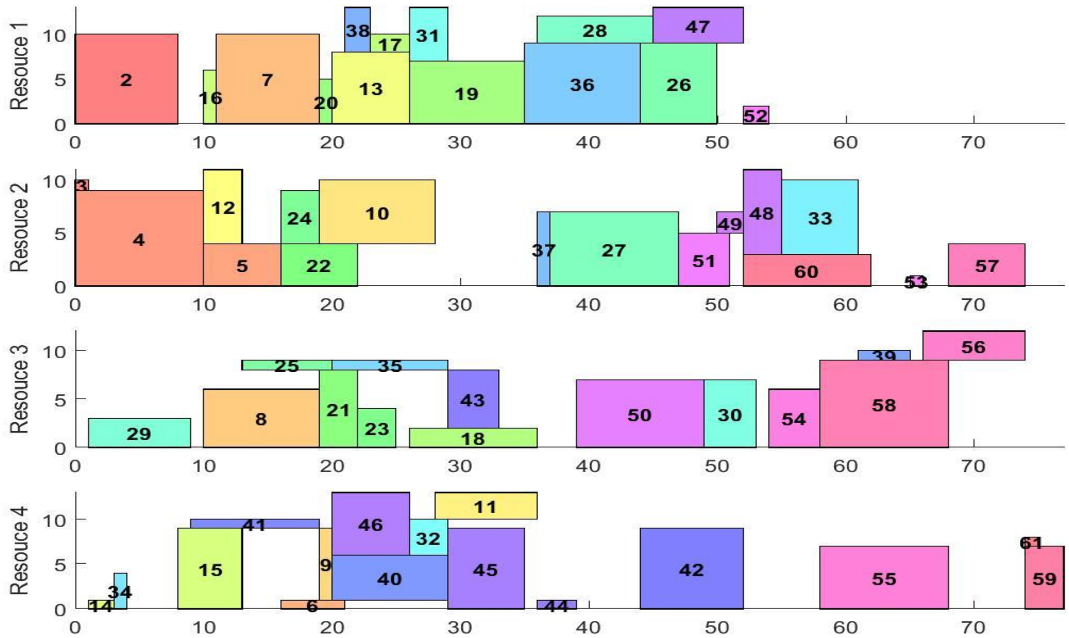

Figure 6. Scheduling of problem j601_1 by the proposed method.

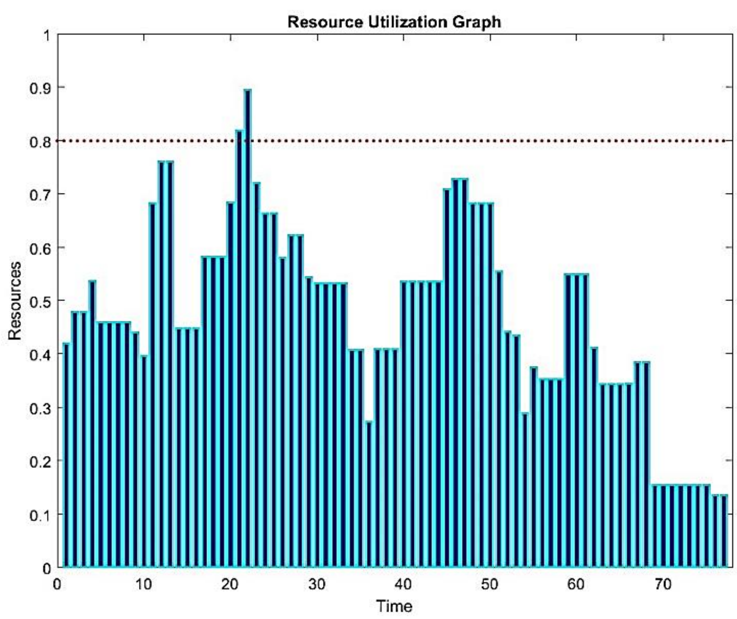

Figure 7. Resource efficiency diagram in time for problem j601_1 by the proposed method.

Figure (8) shows how to schedule the j901_1 problem. In this example, 90 activities should be scheduled on 4 resources in such a way that the best possible situation is established. The result of the proposed method for this problem is shown in this Figure. Moreover, Figure (9) shows the efficiency of resources in time for this problem by the proposed method.

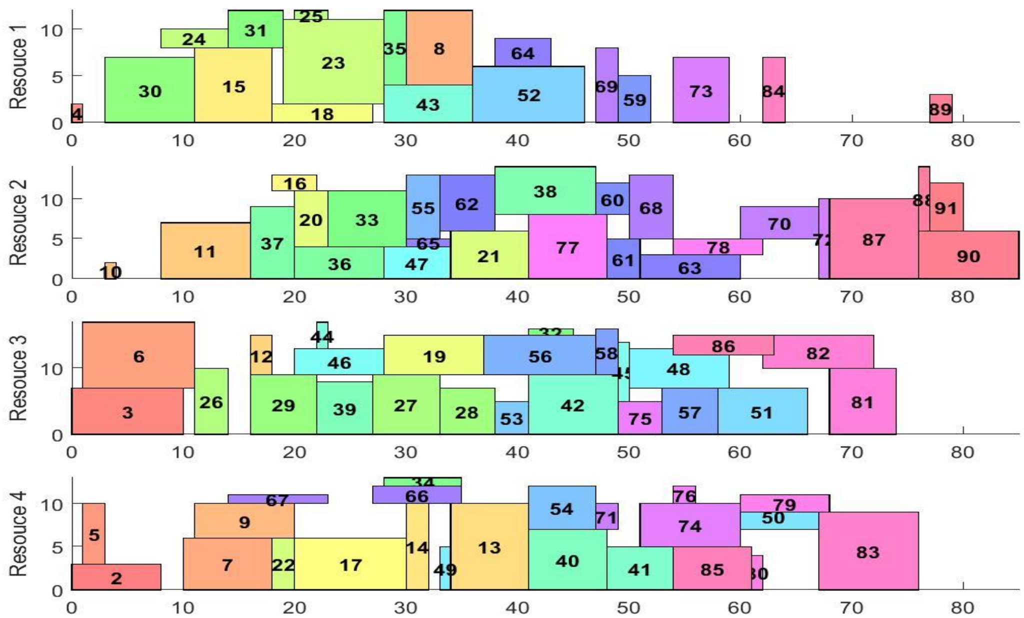

Figure 8. Scheduling of problem j901_1 by the proposed method.

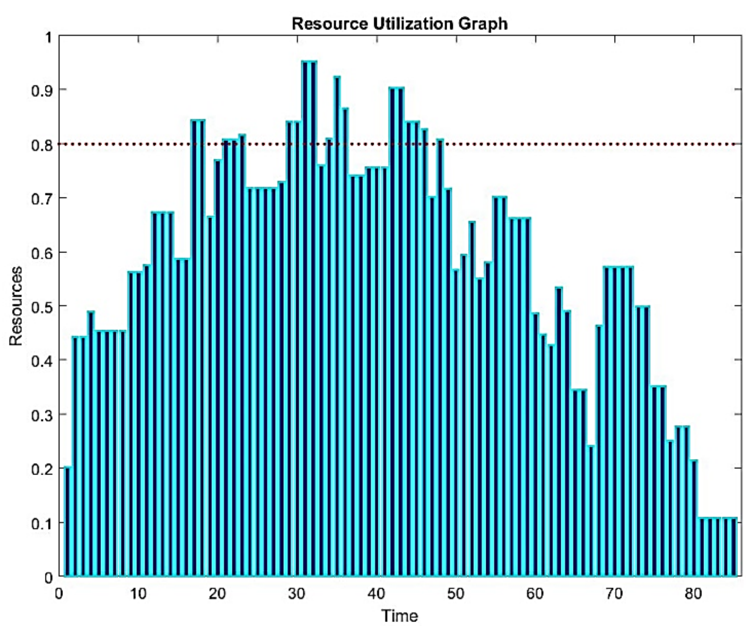

Figure 9. Resource efficiency graph in time for problem j901_1 by the proposed method.

Figure (10) shows how to schedule problem j1201_1. In this example, 120 activities should be scheduled on 4 resources in such a way that the best possible situation is established. The result of the proposed method for this problem is shown in this Figure. In addition, Figure (11) shows the productivity of resources in time for this problem by the proposed method.

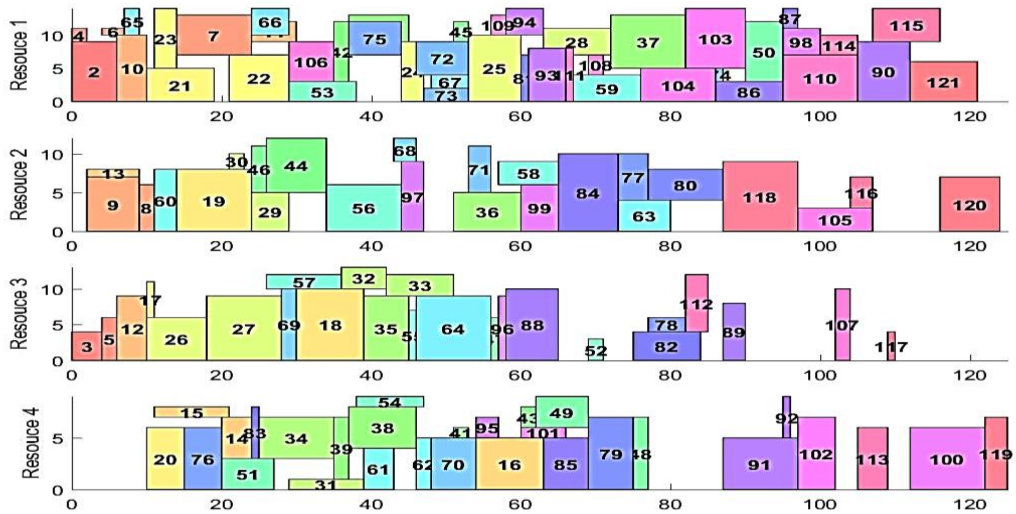

Figure 10. Scheduling of problem j1201_1 by the proposed method.

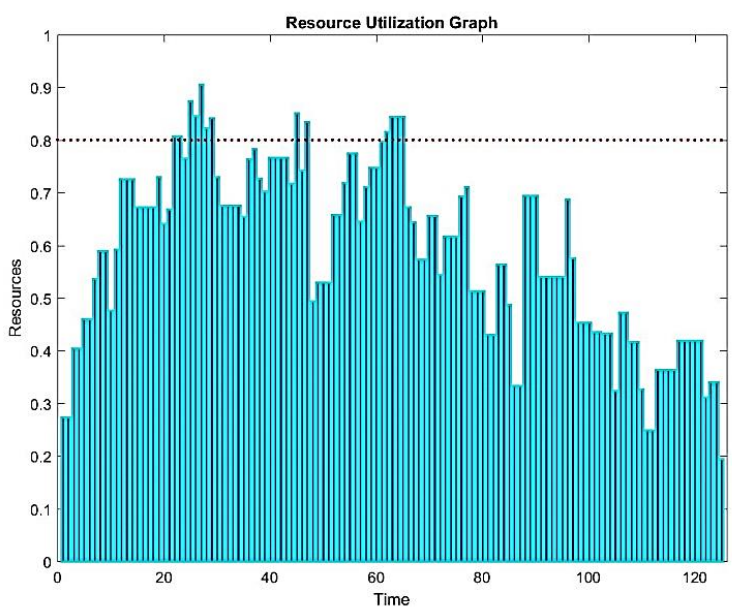

Figure 11. Resource efficiency graph in time for problem j1201_1 by the proposed method.

Finally, in Table (2), the final results of the proposed method are shown along with the comparison with other meta-heuristic methods for evaluation parameters. In each of these algorithms, the initial population is equal to the number of activities (30, 60, 120), and the number of scheduling steps (iterations) is considered equal to 1000.

In Imperialist Competitive Algorithm (ICA), the colonial coefficient (CC) was considered equal to 2. For the Bat Algorithm (BA), the parameters of maximum frequency equal to 5 Hz, minimum frequency equal to 0.1 Hz, maximum loudness equal to 0.9 dB, and minimum loudness equal to 0.1 dB are considered. Moreover, in the Particle Swarm Optimization (PSO) algorithm, the constant training coefficient parameters C1 and C2 are set equal to 2.

Table 2. Final results of the proposed method and comparison with other meta-heuristic methods.

Algorithm	**Standard deviation**			
	J30	J60	J90	J120
Imperialist Competitive Algorithm (ICA)	0.09	11.74	Not reported	36.4
Bat algorithm (BA)	0.42	12.55	Not reported	37.72
Particle swarm optimization (PSO)	0.29	12.03	Not reported	35.71
Current research	**0.016**	**9.28**	15.61	30.85

As shown in Table 2. The calculation results in this research have been investigated and compared using three optimization algorithms. The results show that the results of the calculations are close to each other and in addition, the run time is faster in the presented method. This shows the adequacy of the presented method and therefore it can be used for implementation in the project scheduling process.

7. Discussion and conclusion

Considering the importance of the project scheduling problem and its significant and determining contribution to the success of the projects, RCPSP was studied considering the limited resources. With the addition of resource limitations, the degree of complexity of solving this problem increases, so exact solution methods quickly lose their effectiveness in proportion to the increase in the dimensions of the problem. This research used the meta-heuristic algorithm of intelligent water drops, which had not been used to solve this problem. Moreover, a comprehensive model that has multiple objectives, which is of great interest to project managers and has all the limitations of the real world, was presented. One of the other innovations that took place in this research was the consideration of the multi-objective mode for this extensive problem. Because, as previously stated, the nature of this problem is multi-objective and very few researches have been done in this field, in this research, this matter was taken into consideration, and a multi-objective problem was proposed considering various limitations. Moreover, considering the general prerequisite relationships is another important characteristic of this research because even though considering this limitation makes this problem very complicated, it makes it more practical. In the following, in order to validate the algorithm and examine its ability to solve sample problems in the subject literature and also to measure the algorithm's ability to reach optimal solutions, various experiments were arranged, and the results indicate the appropriate performance of the intelligent water drop algorithm in solving problems. The variety of this problem is wide, and it confirms the good performance of this population-oriented algorithm.

References

Afshar, M. R., Shahhosseini, V., & Sebt, M. H. (2022). A genetic algorithm with a new local search method for solving the multi-mode resource-constrained project scheduling problem. *International Journal of Construction Management*, *22*(3), 357-365.

Aliahmadi, A., Jafari-Eskandari, M., Mozafari, A., & Nozari, H. (2016). Comparing linear regression and artificial neural networks to forecast total productivity growth in Iran. International Journal of Information, Business and Management, 8(1), 93.

Ashadi, R. F., Husin, A. E., & Guntorojati, I. (2021). Scheduling of Infrastructure Construction Projects Using the Manual-Program Evaluation and Review Technique (M-Pert) Method Case Study: Indonesian Sunda Strait Bridge. In *ITB Graduate School Conference* (Vol. 1, No. 1, pp. 76-82).

Ashoka, M. L., & Keihani, H. R. (2020). Factors Influencing the Investors to Invest in Stock Market. International Journal of Management (IJM), 11(1), 166-175.

Balouka, N., Cohen, I., & Shtub, A. (2016). Extending the Multimode Resource-Constrained Project Scheduling Problem by Including Value Considerations. IEEE Transactions on Engineering Management, 63(1), 4-15.

Bathaee, M., Nozari, H., & Szmelter-Jarosz, A. (2023). Designing a new location-allocation and routing model with simultaneous pick-up and delivery in a closed-loop supply chain network under uncertainty. Logistics, 7(1), 3.

Boctor, P.B., Laszczyk, M., Nikulin, I. and Skowroński, M., (2019). iMOPSE: a library for bicriteria optimization in Multi-Skill Resource-Constrained Project Scheduling Problem. Soft Computing, 23(10), pp.3397-3410.

Correia, I., Lourenço, L. L., & Saldanha-da-Gama, F. (2012). Project scheduling with flexible resources: formulation and inequalities. OR spectrum, 34(3), 635-663.

Fallah, M., & Nozari, H. (2021). Neutrosophic mathematical programming for optimization of multi-objective sustainable biomass supply chain network design. Computer Modeling in Engineering & Sciences, 129(2), 927-951.

Fallah, M., Sadeghi, M. E., & Nozari, H. (2021). Quantitative analysis of the applied parts of Internet of Things technology in Iran: an opportunity for economic leapfrogging through technological development. Science and technology policy Letters, 11(4), 45-61.

Ghahremani-Nahr, J., Nozari, H., Rahmaty, M., Zeraati Foukolaei, P., & Sherejsharifi, A. (2023). Development of a novel fuzzy hierarchical location-routing optimization model considering reliability. Logistics, 7(3), 64.

Goli, A., Ala, A., & Hajiaghaei-Keshteli, M. (2023). Efficient multi-objective meta-heuristic algorithms for energy-aware non-permutation flow-shop scheduling problem. *Expert Systems with Applications*, *213*, 119077.

Javanmard, S., Afshar-Nadjafi, B., & Taghi Akhavan Niaki, S. (2022). A bi-objective model for scheduling of multiple projects under multi-skilled workforce for distributed load energy usage. *Operational Research*, *22*(3), 2245-2280.

Kadri, R.L. and Boctor, F.F., (2018). An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. European Journal of Operational Research, 265(2), pp.454-462.

Kolisch, R. (2013). Project scheduling under resource constraints: efficient heuristics for several problem classes: Springer Science & Business Media.

Kolisch, R., & Sprecher, A. (1997). PSPLIB-a project scheduling problem library: OR software-ORSEP operations research software exchange program. *European journal of operational research*, *96*(1), 205-216.

Ma, W., Che, Y., Huang, H., & Ke, H. (2016). Resource-constrained project scheduling problem with uncertain durations and renewable resources. International Journal of Machine Learning and Cybernetics, 7(4), 613-621.

Maghsoudlou, H., Afshar-Nadjafi, B., & Niaki, S. T. A. (2016). A multi-objective invasive weeds optimization algorithm for solving multi-skill multi-mode resource constrained project scheduling problem. Computers & Chemical Engineering, 88, 157-169.

Mehrani, K., Mirshahvalad, A., & Abbasi, E. (2020). Portfolio optimization using black hole meta heuristic algorithm. specialty journal of accounting and economics, 5(2), 1-13.

Movahed, A. B., Aliahmadi, A., Parsanejad, M., & Nozari, H. (2023). A systematic review of collaboration in supply chain 4.0 with meta-synthesis method. Supply Chain Analytics, 100052.

Myszkowski, P. B., Skowroński, M. E., Olech, Ł. P., & Oślizło, K. (2015). Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem. Soft Computing, 19(12), 3599-3619.

Oztemel, E., & Selam, A. A. (2017). Bees Algorithm for multi-mode, resource-constrained project scheduling in molding industry. Computers & Industrial Engineering, 112, 187-196.

Shah-Hosseini, H. (2007). The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. International Journal of Bio-Inspired Computation, 1(1-2), 71-79.

Tavana, M., Abtahi, A.-R., & Khalili-Damghani, K. (2014). A new multi-objective multi-mode model for solving preemptive time–cost–quality trade-off project scheduling problems. Expert Systems with Applications, 41(4), 1830-1846.

Tirkolaee, E. B., Goli, A., Hematian, M., Sangaiah, A. K., & Han, T. (2019). Multi-objective multi-mode resource constrained project scheduling problem using Pareto-based algorithms. *Computing*, *101*(6), 547-570.

Van Eynde, R., & Vanhoucke, M. (2022). A theoretical framework for instance complexity of the resource-constrained project scheduling problem. *Mathematics of Operations Research*.

Zheng, H. Y., Wang, L., & Zheng, X. L. (2017). Teaching–learning-based optimization algorithm for multi-skill resource constrained project scheduling problem. Soft Computing, 21(6), 1537-1548.