Application of Forest Optimization Algorithm for the Resource Constraint Project Scheduling

by Yuesti Et Al

Industrial Engineering
& Management Systems
Vol 21, No 2, June 2022, pp.151-00
ISSN 1598-7248 | EISSN 2234-6473 |

https://doi.org/10.7232/iems.2022.21.2.151

Application of Forest Optimization Algorithm for the Resource Constraint Project Scheduling

Anik Yuesti
Universitas Mahasaraswati Denpasar Bali Indonesia

Marischa Elveny*
Data Science & Computational Intelligence Research Group, Universitas Sumatera Utara, Medan, Indonesia

Hussein Tuama Hazim
University of Misan, Add country

Oleg N. Polukhin
Belgorod State University, Russia

Israr Ahmad
Universiti Utara Malaysia, Kedah, Malaysia

Tzu-Chia Chen
CAIC, DPU, Thailand

Vo Hoang Ca
Faculty of Business Administration, Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam

Huynh Tan Hoi
Language department, FPT University, Ho Chi Minh City, Vietnam

Surendar Aravindhan
Department of Pharmacology, Saveetha Dental College and Hospital, Saveetha Institute of Medical and Technical Sciences, Chennai, India

(Received: December 00, 2020 / Revised: December 00, 2020 / Accepted: December 00, 2020)

ABSTRACT

Project scheduling is one of the most complex yet important issues in project-based planning. Scheduling is a topic that has entered the project from the production space and has been considered by many researchers in recent years. This problem has always faced many challenges due to its complex limitations in the real world. One of these issues is resource scheduling, known in optimization as the RCPSP problem. The study's main aim is to investigate the application of a forest optimization algorithm for resource constraint project scheduling. In this research, the RCPSP problem is optimized. A mathematical model is then proposed and then optimized using meteorological algorithms for forest optimization and refrigeration simulation. A hybrid meta-algorithm has also been developed for this document. Examination and comparisons of this algorithm show that the proposed method has the necessary efficiency both in terms of speed and quality of solutions.

Keywords: Project Optimization, Project Scheduling, Forest Algorithm, Refrigeration Simulation Algorithm

* Corresponding Author, E-mail: anikyuesti@unmas.ac.id

1. INTRODUCTION

Project management techniques have evolved and been practiced for half a century. A large part of efforts in the area of development of project management concepts has been dedicated to generating scheduling models (Sabela, 2020; Hartani *et al*., 2021; Barzamini *et al*., 2022; Zabihi *et al*., 2012). As such, project scheduling is recognized as one of the fundamental pillars of project management (Tirkolaee *et al*., 2019). Among the existing scheduling problems, the resource-constrained project scheduling problem (RCPSP) has been introduced by various researchers as one of the most significant and challenging operational problems (Koulinas *et al*., 2014). In fact, it is the most general scheduling problem, and job shop scheduling, flow shop scheduling, and other scheduling problems are all a subset of this issue (Kurilova, 2021). The main objective of RCPSP is to allocate a set of constrained resources to activities by adhering to prerequisite relationships to achieve various and specific goals. Nevertheless, the most common objective of these methods is to minimize the project completion time (Pahlevan *et al*., 2021; Golmohammadi *et al*., 2016; Alwreikat and Rjoub, 2020; Khan *et al*., 2020). Various objective functions considered for RCPSPs divide the problems into different branches, one of which is the classic RCPSP.The primary goal of this technique is to reduce the implementation time of a project by finding a proper sequence of project activities in a way that precedence and latency constraints of the project and the different types of resource constraints available are satisfied simultaneously (Susilawati, 2021; Goli and Malmir, 2020; Shannaq *et al*., 2020; Seyedhosseini *et al*., 2016). This could be recognized as an NP-hard optimization problem (Bukata *et al*., 2015). Proposing more efficient solutions for such problems has been taken into account by several researchers in the past few years (Valls *et al*., 2005; Rjoub *et al*., 2017).

Different methods have been proposed in the literature for solving RCPSP, including exact solution generation methods and algorithms. One of these algorithms is the branch and bound, which has been presented as a proper technique for solving RCPSP and achieving an optimal solution (Aravindhan *et al*., 2021). Given the classification of RCPSP as an NP-hard problem in terms of computational complexity, these methods can only solve examples with less than 60 activities (Koulinas *et al*., 2014). Therefore, heuristic and metaheuristic algorithms are the only solutions for solving RCPSP with a high number of activities (Valls *et al*., 2005). Several algorithms have been proposed for these problems; for instance, Kharat and Shelar (2021), presented a solution based on priority rules. In addition, Nonobe and Ibaraki (2002) and Klein (2000) developed techniques based on tabu search algorithms. In another study, Bouleimen and Lecocq (2003) applied a simulated annealing algorithm, and Susminingsih *et al*. (2021) and Fleszar and Hindi (2004) created neighborhood search techniques. Among the population-based algorithms, the genetics algorithm was applied in (Qazani *et al*., 2021; Valls *et al*., 2008), whereas scatter search algorithm and ant colony algorithm were used by Debels *et al*. (2006) and Harmoko (2021). Other algorithms used to solve RCPSP include artificial neural network algorithm (Sembekov *et al*., 2021), bee colony algorithm (Ziarati *et al*., 2011), and frog jumping steps algorithm (Fang and Wang, 2012). In addition, many researchers have used a hybrid approach to solving RCPSP. Wang and Fang (2012), and Susminingsih *et al*. (2011), developed techniques with a hybrid approach.

The present study presents a novel technique tosolve classic RCPSP using a forest optimization algo- rithm (FOA) and gradual simulated annealing. The re- mainder of the article is structured as follows: Section 2 describes the mathematical model of the problem and section 3 introduces the FOA. In addition, section 4 brief- ly explains the gradual simulated annealing, and section 5 presents the proposed method. Furthermore, section 6 evaluates the performance of the proposed algorithm. Finally, section 7 concludes and presents suggestions for future studies.

2. STATEMENT OF THE PROBLEM

If we consider the activities of a project in a set such as N and the prerequisite relationships between the activities in a set such as A, RCPSP can be represented by a directed acyclic graph (G = (N, A). Activities are numbered from 0 to 1 + n, in which activities 0 and 1 + n are dummy activities and indicate the beginning and end of the project, respectively. The implementation time of each activity, such as I, is shown by di, and d0=dn+1=0. In addition, there are K types of renewable resources, and there are RK units of the k-th resource. Each activity, such as i, requires a certain amount of k resource to be implemented, which is shown by rik and r0k = rn+1 = 0. The start time and the end time of each activity, such as i, is shown by Si and Fi, respectively. In addition, P(t) presents activities running at t time, and the T variable is equal to the longest project implementation time.

Before presenting the model, the variables used in it are described:

A_i : A set of all risk responses;
R : A set of all risks;
W : A set of project activities;
M : A pair of responses eliminating each other;
$\bar{M}$: A set of paired responses with positive effects on each other;
B : Total budget to provide project risk responses;
W_k : k-*th* activity;
R_j : Response to j-*th* risk;

A_i : Response to i-*th* risk;

C_{ij} : Cost of implementing i-*th* response to j-*th* risk

S_j^k : Estimating days of delay in case of j-*th* risk during k-*th* activity;

q_j^k : Estimating quality loss in case of j-*th* risk during k-*th* activity;

e_{ij} : Estimating risk reduction rate in terms of i-*th* response to j-*th* risk;

S_{ij}^k : Estimating decrease in working days of k-*th* activity in terms of i-*th* response to j-*th* risk;

q_{ij}^k : Estimating improved quality of k-*th* activity in terms of i-*th* response to j-*th* risk;

ε^k : Time between end of k-*th* activity and its subsequent activity;

δ^k : Acceptable quality of k-*th* activity without affecting that of its subsequent ones;

T_{Max} : Maximum allowable project delay;

Q_{Max} : Maximum allowable project quality loss;

X_{ij} : Zero-one decision variable: whenever i-*th* response is chosen for j-*th* risk, it is equal to one, and otherwise zero.

The first objective function of the zero-one goal-programming model maximizes the effect of risk response (Objective Function 1); in other words, this objective function takes an optimal response to reduce the negative impact of risks as much as possible. However, the main point not mentioned in previous research is that the project manager does not aim to mitigate negative effects irrespective of response costs. This means that the minimum cost to deal with risks must be considered (Objective Function 2) along with lowering negative effects, so this generates a dual-objective optimization problem. The model is illustrated below.

$$Max\, z_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} (e_{ij} X_{ij}) \tag{1}$$

$$Min\, z_2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (C_{ij} X_{ij}) \tag{2}$$

$$S.t. \sum_{i=1}^{m} \left(C_{ij} Max_j X_{ij}\right) \le B \qquad j = 1,2,\dots,n \tag{3}$$

$$\sum_{j=1}^{n} S_j^k - \sum_{j=1}^{n} \sum_{i=1}^{m} \left(S_{ij}^k X_{ij}\right) \le \varepsilon^k, \qquad k = 1,2,\dots,l-1 \tag{4}$$

$$\sum_{j} \quad \sum \sum$$

$$\sum_{j=1}^{n} q_j^l - \sum_{j=1}^{n} \sum_{i=1}^{m} \left(q_{ij}^l X_{ij}\right) \le Q_{Max} \tag{7}$$

$$X_{ij} + X_{i'j} \le 1 \left(A_i, A_{i'}\right) \in M, \quad i,i' = 1,2,\dots,m,\ j,j' = 1,2,\dots,n \tag{8}$$

$$X_{ij} + X_{i'j} = 1 \left(A_i, A_{i'}\right) \in M, \quad i,i' = 1,2,\dots,m,\ j,j' = 1,2,\dots,n \tag{9}$$

$$X_{ij} - X_{i'j} \le 1 \left(A_i, A_{i'}\right) \in \bar{M}, \quad i,i' = 1,2,\dots,m,\ j,j' = 1,2,\dots,n \tag{10}$$

$$X_{ij}, X_{i'j'} \in \{0,1\}, i,i' = 1,2,\dots,m,\ j,j' = 1,2,\dots,n \tag{11}$$

In this model, constraint 3 ensures that the total cost of providing risk responses does not exceed the total budget. Inserting Max_j in this constraint is also to avoid recounting the responses. As well, constraint 4 shows that each activity must be fulfilled within its allowable time and not cause delays in the subsequent activities. Constraint 5 is the same as before, except that it confirms that reducing the quality of one activity does not lower that of subsequent ones and result in an unacceptable decline in project quality. Moreover, constraint 6 indicates that the final activity ends with an allowable delay and the project is not delayed in an unreasonable manner. Constraint 7 also determines the standard of the final project quality and does not reduce it. As well, Constraint 8 is utilized for mutually exclusive responses. That is, if one of them is employed, the other response should not be chosen. Constraint 9 is further applied for responses, one of which must be selected. Furthermore, Constraint 10 maintains that if one of these two responses is chosen, the other one must be subsequently selected. The last constraint also shows the problem decision variable.

3. FOREST OPTIMIZATION ALGORITHM (FOA)

FOA is deemed one of the newest evolutionary algorithms introduced by Ghaemi and Derakhshani (2014). The algorithm has been inspired by various seed dispersal methods of trees in forests, such as local seed dispersal and long-distance seed dispersal (Cain *et al*., 2000).

Vol 21, No 2, June 2022, pp.151-00, © 2022 KIIE

$$\sum_{j=1} q_j^k - \sum_{j=1}\sum_{i=1} \left(q_{ij}^k X_{ij}\right) \le \delta^k, \quad k = 1,2,\ldots,l-1 \tag{5}$$

$$\sum_{j=1} S_j^l - \sum_{j=1}\sum_{i=1} \left(S_{ij}^l X_{ij}\right) \le T^{Max} \tag{6}$$

3.1 Construction of Primary Forest Trees

Similar to other evolutionary algorithms, FOA starts with an initial population of trees, where each tree can be a solution for the problem. In addition to the values of the variables, a tree also has an age, which is initially considered zero. In addition, each tree is shown in the form of an array of $1 \times (1 + N_{var})$, where N_{var} shows the number of variables in the problem.

3.2 Local Seeding

The operator acts by choosing a cell from the tree for each zero-aged tree and selecting a random number, such as r, from the range of ($-\Delta x$ and Δx). Afterwards, the value of the house is added to the r number. Δx is a value smaller than the upper bound of the variables. The process is repeated at a frequency of local seeding changes (LSC). In fact, LSC shows the number of trees that must be generated from each zero-aged tree at the local seeding stage. After performing the above operation, the age variable of newly formed trees is set to zero, and the variable increases by one unit for other trees in the forest.

3.3 Forest Size Restriction

Two variables are used to restrict the trees in the forest. The first variable is life time, which determines the maximum age of each tree while the second variable is area limit, which is equal to the maximum number of trees that can exist in the forest. After local dispersal, the trees that have reached their life time are eliminated from the forest and added to the set of candidates, which encompasses the trees eliminated from the forest. Afterwards, if the number of trees left in the forest is more than the area limit variable, the trees in the forest are sorted based on the value of their fit function and some trees are selected from the best trees to the number of area limit for survival of the forest and the rest of the trees are added to the candidate set.

3.4 Global Seeding

In this step, a number of trees in the set of candidates are first selected, and the number is determined by the transfer rate variable, which represents the percentage of trees in the set of candidates. Afterwards, a certain number of cells are selected from each array (tree), and the value of the cell in the tree is replaced by a new value of the permissible interval for problem variables. The number of cells selected from each tree is determined by another variable of the algorithm known as global seeding changes (GSC). After the mentioned operations, the variable of tree age is set at zero and added to the forest.

3.5 Maintenance and Updating the Best Tree

After each iteration of the algorithm and performing the above steps, the best tree is selected in terms of the value of the fit function and its age value is set to zero so that it can perform local dispersal operations in the next step.

3.6 Termination Conditions

Similar to other metaheuristic algorithms, three termination conditions can be considered for forest optimization algorithm: first, a certain number of algorithm iteration; second, no change is observed in the value of the fit function of the best tree during several iterations, and third, the condition for achieving a certain level of accuracy.

4. GRADUAL SIMULATED ANNEALING

Simulated annealing was developed by Kirkpatrick *et al*. as an alternative to local search and a possible general method for solving optimization problems. This technique is practiced to reach a properly sorted solid matter and minimize its energy. This technique involves exposing the substance to high temperature and then gradually reducing the temperature. The main idea in this method is based on accepting the worse solutions in some conditions in order to escape the local optimum. Simulated annealing is so named because of its analogy to the process of physical. The gradual simulated annealing process, which reduces energy in solid, was defined by Zegordi *et al*. as follows: at each stage, the atom is slightly displaced, which leads to a change in the energy system shown by ΔE. If $\Delta E \leq 0$, the displacement of two atoms will be accepted and the displaced solid structure or atom will be used as the starting point for the next step. On the other hand, if $\Delta E > 0$, the process will be dealt with in a probabilistic manner, meaning that the possibili-ty of acceptance of the solid structure will be determinedusing the equation below, where T is the temperature de-gree and k_b is the Boltzmann constant. To solve an optimization problem, the simulated annealing algorithm starts with an initial solution and moves to the neighboring solutions in a loop. The algorithm will select the neighboring solution (i.e., moves the solution) if it is more efficient than the current solution. Otherwise, the algorithm accepts the solution as the current solution with the possibility of $\exp(-\Delta E / T)$. In this equation, ΔE is the difference between the objective function of the cur- rent solution and the neighboring solution, and T is a parameter known as temperature. The temperature is gradually reduced after performing several iterations at each temperature. The temperature is set very high at the initial stages so that there is a higher probability of accepting worse solutions. There will be a lower chance of accepting worse solutions in the final stages by a gradual decrease of temperature, and the algorithm will converge toward a proper solution.

6. PROPOSED METHOD TO SOLVE RCPSP USING FOREST OPTIMIZATION ALGORITHM

Since forest optimization algorithm has been introduced to solve continuous problems and RCPSP is a discrete problem, some changes should be made, such as redefining LSC and GLC steps in the initial version of the algorithm to adapt it to the problem conditions in order to solve the RCPSP using the forest optimization algorithm. In addition, the gradual simulated annealing mechanism is used in the present article to improve the performance of the proposed algorithm and increase the possibility of transferring fit trees to the next stage at each step of the algorithm.

6.1 Structure of Trees

Each tree is shown with an array of $1 \times (1 + N_{var})$, where N_{var} is the number of project activities plus two dummy activities at the beginning and end of the process, and the other cell of the array is related to the variable of tree age. Each tree encompasses a series of project activities in a way that precedence and latency constraints of the project network are observed. In other words, each tree involves a doable sequence for performing project activities. To generate each tree, one activity is randomly selected from activities, the prerequisites for which are placed in the current home, for each cell of the array, and the process continues until reaching the final activity.

6.2 Implementation Duration Calculation Method

In order to estimate the duration of the project for each tree, we start from the array and identify activities that could be performed simultaneously (i.e., their prerequisites have been performed completely). These activities are implemented simultaneously if allowed by resource constraints. Otherwise, the activity with the shortest implementation time remained is selected among the running activities and ended, followed by re-assessing the mentioned operation at the current time. The process continues until reaching the n+1 activity; evidently, the fit value of each tree equates to the start time of the n+1 process.

6.3 Simulation of Local Seed Dispersal Operator

In order to simulate the local seed dispersal operator, two elements of the array are first selected and the current activity in the cell with a larger number is transferred to a cell with a smaller number if the precedence and latency constraints of the project network are not violated. Afterwards, the activities between the two selected cells are transferred one unit forward. Otherwise, the process is repeated until the process is made possible.

6.4 Simulation of Global Seed Dispersal Operator

In order to simulate the global seed dispersal operator, a cell of the tree array is selected first, and the process of generating a doable solution is continued from that cell. Since one cell of the array is randomly selected from the candidate activities, there is a high possibility of generating new sequences by this technique. In the proposed method, the GSC variable equates to the maximum number of selected cells. If the value is chosen to be high, there is a lower possibility of changes in the structure of the selected tree. On the other hand, if the value is chosen to be very low, the selected tree will be extremely changed after the global seed dispersal process. Therefore, it is crucial to select an appropriate value for the parame-ter in the algorithm implementation process. Following the generation of a new tree with the method explained above, the gradual simulated annealing mechanism is used to select between the tree chosen from the candidatelist and the changed tree to be added to the forest in the next algorithm stages. To this end, we first calculate the time (fit) of the tree changed in the previous stage. After-wards, the selected tree will be added to the forest and itsage will be set at zero if its time is shorter than the time of the tree selected from the list of candidates. Otherwise, an appropriate solution will be selected with the possibility of $e^{\frac{-df}{T_0}}$ and if the probability is larger than a uniform random number between zero and one.

7. COMPUTATIONAL RESULTS

The examples existing in PSPLIB, which are known as reference examples fir RSPSP, are used to evaluate the performance of the proposed method. This set includes 90, 60, 30 and 120-activity examples, and the 30, 60, and 120-activity examples have been used in previous studies to assess the performance of various algorithms. While an optimal solution exists for 30-activity examples, the lower bound scheduling, which is obtained by the classic critical path method (CPM), is used as the criterion for assessing the deviation of algorithms. In this article, 20 examples are selected randomly from the examples existing for each mentioned group. All of these examples have been mentioned in (Cain *et al*., 2000). The proposed method is coded by C# programming language and run in Visual studio 2013 programming space on a computer with 2.5 GHz Intel Core i5 processor and 4 GB of RAM with Windows 8.1 operating system. Adjusting the variables of any evolutionary algorithm plays an important role in its implementation. The variables are initialized for the proposed method following the implementation of various experiments, as follows: the number of initial forest trees and remaining trees in the forest at each stage (area limit) equal to six, GSC variable value equal to one-fifth of the number of activities per problem, LSC variable value equal to five, transfer rate variable value equal to four, maximum age of each tree (life time) equal to 5, and cooling rate equal to 0.85. In addition, the termination condition of the algorithm, is equal to producing a certain number of scheduling (trees), similar to previous studies. In this article, this number is equal to 5000.

Regarding 30-activity examples, the proposed method is able to achieve optimal solutions for all 20 examples selected in a rapid manner and solve the examples without deviation. Table 1 shows the lower bound of scheduling for 60-activity examples and the solutions obtained in this regard. As observed, the proposed method is able to solve the examples of this category with very little difference compared to the low bound scheduling.

Table 2 presents the lower bound of scheduling for

Table 1. Results related to examples with 60 activities

Lower bound of scheduling	Results of the proposed method	Deviation (%)
66	69	4.54546
72	81	12.5
75	86	14.6667
85	93	9.41177
80	84	5
65	67	3.07692
82	84	2.43902
78	80	2.5641
78	79	1.28205
54	58	7.40741
64	67	4.6875
53	58	9.43396
66	68	3.0303
65	69	6.15385
69	76	10.1449
60	68	13.3333
69	71	2.89855
105	106	0.95238
81	83	2.46914
83	86	3.61446

Table 2. Results related to examples with 120 activities

Lower bound of scheduling	Results of proposed method	Deviation (%)
107	133	24.2991
116	136	17.2414
112	144	28.5714
99	115	17.1717
112	147	31.25
75	100	33.3333
107	130	23.3645
106	134	26.4151
125	146	16.8
104	129	24.0385
101	130	29.703
108	140	29.6296
108	147	33.3333
100	122	24
117	148	26.4957
106	131	23.5849
87	110	26.4368
99	124	25.2525
109	140	30.2752
92	125	33.6957

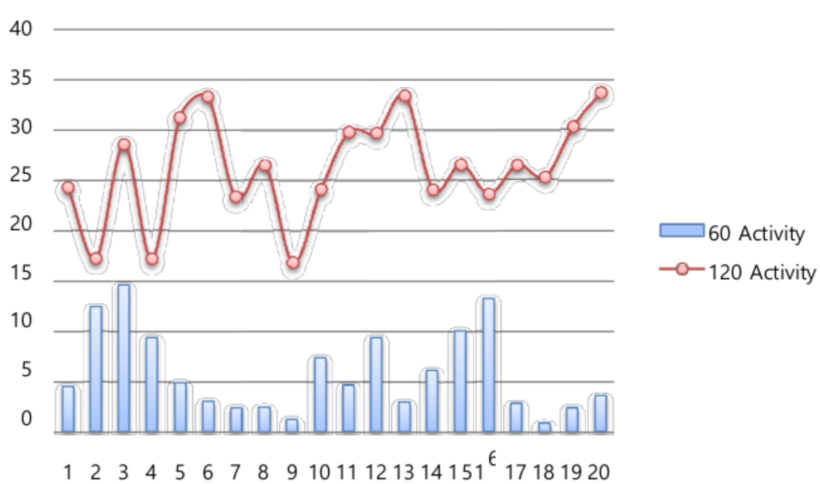

Figure 1. Comparison of deviation percentage in different projects.

examples with 120 activities and solutions obtained by the proposed method. The solutions of the presented method show the suitable applicability of the method based on the high number of activities and the mean deviation of the existing methods, which is equal to 34.384. Figure 1 compares the percentage of deviation in two 60- and 120-activity examples.

According to Figure 1, the deviation percentage of different activities is higher when a project with 120 activities is assessed, compared to a project with 60 activities. In other words, increasing the number of activities has a direct impact on the degree of project deviation, which confirms the fact that the larger the size of the project, the higher the necessity of resource consumption optimization in this regard.

8. CONCLUSION

The present study suggested a novel technique for RCPSP based on a forest optimization algorithm. Since the primary forest optimization algorithm is an evolutionary algorithm for solving continuous optimization problems, a new method was proposed in this paper by applying changes and redefining its leading operators to solve the project scheduling problem with classic resource constraints. In addition, tree selection was optimized for the following stages of the algorithm by using the gradual simulated annealing mechanism in the global seed dispersal stage. According to computational results, the proposed method had an extremely suitable performance regarding solving the problem. Given the suitability of the proposed method in solving RCPSP, it should be used to solve other problems related to project scheduling, including multi-mode resource-constrained project scheduling problems and resource-constrained project scheduling problems with minimal and maximal time lags in future studies.

REFERENCES

Agarwal, A., Colak, S., and Erenguc, S. (2011), A neurogenetic approach for the resource-constrained project scheduling problem, *Computers & Operations Research*, **38**(1), 44-50. – not cited

Alwreikat, A. A. M. and Rjoub, H. (2020), Impact of mobile advertising wearout on consumer irritation, perceived intrusiveness, engagement and loyalty: A partial least squares structural equation modelling analysis, *South African Journal of Business Management*, **51**(1), 1-11.

Aravindhan, S., Younus, L. A., Hadi Lafta, M., Markov, A., Ivanovna Enina, Y., Yushchenko, N. A., Thangavelu, L., Mostafavi, S. M., Pokrovskii, M. V., and Ahmadi, M. (2021), P53 long noncoding RNA regulatory network in cancer development, *Cell Biology International*, **45**(8), 1583-1598.

Barzamini, H., Shahzad, M., Alhoori, H., and Rahimi, M. (2022), A multi-level semantic web for hard-to-specify domain concept, Pedestrian, in ML-based software, *Requirements Eng*, Available from: https://

Bouleimen, K. L. E. I. N. and Lecocq, H. O. U. S. N. I. (2003), A new efficient simulated annealing algorithm for the resource-constrained project schedulingproblem and its multiple mode version, *European Journal of Operational Research*, **149**(2), 268-281.

Bukata, L., Šůcha, P., and Hanzálek, Z. (2015), Solving the resource constrained project scheduling problem using the parallel Tabu search designed for the CUDA platform, *Journal of Parallel and Distributed Computing*, **77**, 58-68.

Cain, M. L., Milligan, B. G., and Strand, A. E. (2000), Long-distance seed dispersal in plant populations, *American Journal of Botany*, **87**(9), 1217-1227.

Christofides, N., Alvarez-Valdes, R., and Tamarit, J. M. (1987), Project scheduling with resource constraints: a branch and bound approach, *European Journal of Operational Research*, **29**(3), 262-273. – not cited

Debels, D., De Reyck, B., Leus, R., and Vanhoucke, M. (2006), A hybrid scatter search/electromagnetism meta-heuristic for project scheduling, *European Journal of Operational Research*, **169**(2), 638-653.

Fang, C. and Wang, L. (2012), An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem, *Computers & Operations Research*, **39**(5), 890-901.

Fleszar, K. and Hindi, K. S. (2004), Solving the resource-constrained project scheduling problem by a variable neighbourhood search, *European Journal of Operational Research*, **155**(2), 402-413.

Ghaemi, M. and Feizi-Derakhshi, M. R. (2014), Forest optimization algorithm, *Expert Systems with Applications*, **41**(15), 6676-6687.

Goli, A. and Malmir, B. (2020), A covering tour approach for disaster relief locating and routing with fuzzy demand, *International Journal of Intelligent Transportation Systems Research*, **18**(1), 140-152.

Golmohammadi, A., Bani-Asadi, H., Zanjani, H., and Tikani, H. (2016), A genetic algorithm for preemptive scheduling of a single machine, *International Journal of Industrial Engineering Computations*, **7**(4), 607-614.

Harmoko, H. (2021), The five dimensions of digital technology assessment with the focus on robotic process automation (RPA), *Tehnički Glasnik*, **15**(2), 267-274.

Hartani, N. H., Haron, N., and Tajuddin, N. I. I. (2021), The impact of strategic alignment on the sustainable competitive advantages: Mediating role of it implementation success and it managerial resource, *International Journal of eBusiness and eGovernment Studies*, **13**(1), 78-96.

Khan, W., Ahmed, A. A. A., Hossain, M. S., Neogy, T. K. (2020), The interactive approach to working capital knowledge: Survey evidence, *International Journal of Nonlinear Analysis and Applications*, **11**(Special Issue), 379-393.

Kharat, V. and Shelar, M. (2021), An efficient query optimizer with materialized intermediate views in distributed and cloud environment, *Tehnički Glasnik*, **15**(1), 105-111.

Klein, R. (2000), Project scheduling with time-varying resource constraints, *International Journal of Production Research*, **38**(16), 3937-3952.

Koulinas, G., Kotsikas, L., and Anagnostopoulos, K. (2014), A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem, *Information Sciences*, **277**, 680-693.

Kurilova, A. A. (2021), A mathematical model of the vehicles routing problem of perishable materials using genetic algorithm, *Industrial Engineering & Management Systems*, **20**(2), 315-321.

Nonobe, K. and Ibaraki, T. (2002), Formulation and tabu search algorithm for the resource constrained project scheduling problem, *Essays and Surveys in Metaheuristics*, Springer, Boston, MA, 557-588.

Pahlevan, S. M., Hosseini, S. M. S., and Goli, A. (2021), Sustainable supply chain network design using products' life cycle in the aluminum industry, *Environmental Science and Pollution Research*, 1-25.

Qazani, M. R. C., Asadi, H., Lim, C. P., Mohamed, S., and Nahavandi, S. (2021), Prediction of motion simulator signals using time-series neural networks, *IEEE Transactions on Aerospace and Electronic Systems*, **57**(5), 3383-3392.

Rjoub, H., Civcir, I., and Resatoglu, N. G. (2017), Micro and macroeconomic determinants of stock prices: The case of Turkish banking sector, *Romanian Journal of Economic Forecasting*, **20**(1), 150-166.

Sabela, P. T. (2020), Trends and realities: Provision of low-cost housing at king cetshwayo district municipality, kwazulu-natal province, South Africa, *International Journal of Social Sciences and Humanity Studies*, **12**(1), 130-145.

Sembekov, A., Tazhbayev, N., Ulakov, N., Tatiyeva, G., and Budeshov, Y. (2021), Digital modernization of Kazakhstan's economy in the context of global trends, *Economic Annals-XXI*, **187**(1-2), 51-62.

Seyedhosseini, S. M., Esfahani, M. J., and Ghaffari, M. (2016), A novel hybrid algorithm based on a harmony search and artificial bee colony for solving a portfolio optimization problem using a mean-semi variance approach, *Journal of Central South University*, **23**(1), 181-188.

Shannaq, B., Al Shamsi, I., and AlAzzawi, F. J. I. (2020), Innovative algorithm for managing the number of clusters, *International Journal of Recent Technology and Engineering*, **8**(5), 310-315.

Susilawati, D. M. (2021), Transforming the digital leadership to improve public service performance in the COVID-19 outbreak, *Economic Annals-XXI*, **188**(3-4), 31-38.

Susminingsih, Alias, N., Supriyanto, A. S., Kanafi, I., and Subiyadi, A. (2021), Labeling local brand to increase coffee shop visits during the COVID-19 pandemic, *Economic Annals-XXI*, **188**(3-4), 149-157.

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

Valls, V., Ballestín, F., and Quintanilla, S. (2005), Justification and RCPSP: A technique that pays, *European Journal of Operational Research*, **165**(2), 375-386.

Valls, V., Ballestin, F., and Quintanilla, S. (2008), A hybrid genetic algorithm for the resource-constrained project scheduling problem, *European Journal of Operational Research*, **185**(2), 495-508.

Wang, L. and Fang, C. (2012), A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem, *Expert Systems with Applications*, **39**(3), 2451-2460.

Zabihi, O., Khodabandeh, A., and Mostafavi, S. M. (2012), Preparation, optimization and thermal characterization of a novel conductive thermoset nanocomposite containing polythiophene nanoparticles using dynamic thermal analysis, *Polymer Degradation and Stability*, **97**(1), 3-13.

Ziarati, K., Akbari, R., and Zeighami, V. (2011), On the performance of bee algorithms for resource-constrained project scheduling problem, *Applied Soft Computing*, **11**(4), 3720-3733.

AUTHOR BIO

Application of Forest Optimization Algorithm for the Resource Constraint Project Scheduling

ORIGINALITY REPORT

3%	2%	3%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Manizheh Ghaemi, Mohammad-Reza Feizi-Derakhshi. "Forest Optimization Algorithm", Expert Systems with Applications, 2014 Publication	2%
2	kzyjc.alljournals.cn Internet Source	2%

Exclude quotes	Off	Exclude matches	< 2%
Exclude bibliography	On		