

Solving Multi mode Resource Constrained Project Scheduling with IC Algorithm and Compare It with PSO Algorithm

Sina Namazi^{1,*}, Mohammad Reza Zare Fard², Seyed Mohammad Reza Mousavi¹,
Mohammad Bahrami Nasab²

¹Department of Industrial Engineering, Islamic Azad University, Karaj Branch, Iran

²Department of Industrial Engineering, Islamic Azad University, Qazvin Branch, Iran

Abstract Well-timed execution of project activities requires an integrated and effective planning and management in an accurate adjustment of the project activities execution time. Although a lot of works has been done on MRCPSP (multi-resource constrained project scheduling problem) with the goal of minimizing the project makespan, but considering other goals in current projects like the project total cost is also important to the managers. So in this article, a mathematical model has been considered for resource-constrained project scheduling with two objectives: minimizing both project total cost and makespan. And this problem has been solved with a meta-heuristic algorithm called IC (Imperialist Competitive) algorithm which has never been used to solve this type of problems. The results have been compared with PSO algorithm in which they are satisfactory.

Keywords MRCPSP, Imperialist Competitive algorithm, MORCPSP, PSP

1. Introduction

RCPSP (Resource-Constrained Project Scheduling Problem) is scheduling the project activities according to precedence relations and constraint of resources. RCPSP consists of a project with j activities $j=1, \dots, j$. the duration of an activity j is donated by D_j . All activities can start just once, and they can be preemptive or not. Because of technical needs, there is some precedence relations among activities donated as a set of relations P_j . It shows that activity j cannot be started unless all of its precedence relations and precedents ($i \in P_j$) are fulfilled. Precedence relations can be represented by an AON (Activity on Node network); with the presupposition of acyclic feature of the network. Each activity needs a special amount of resources for completion and execution [1].

In standard RCPSP, an activity can only be executed in one way. So the execution time and amount of needed resources will be definite [1].

First research on multi-state activities has been done by Elmaghraby [2]. An activity, which only could have been executed in one way in standard state, now can be executed in several states or methods. Each state needs its own

execution time and needed resources.

Multi-state RCPSP is shown MRCPSP in this problem. RCPSP problem has a presupposition of being multi-state. This presupposition is defined as follows:

"An activity j must be started at one point and must be executed in one of its execution states which is shown by $1, \dots, M_j$. The activity must be terminated in the same execution state which it had been started. Preemption is not suitable in the basic model of MRCPSP [2]. The execution time for activity j in state m is shown by D_{jm} . The amount of resource needed for execution of activity j in state m is shown by r_{jmk} . Besides the renewable resources, non-renewable resources are also being used in multi-state models. A scheduling plan for multi-state RCPSP model is specified with a start time S_j and a state M_j for each j activity. The most well-known target function in these kinds of problems is minimizing the project makespan. [2]"

It's obvious that if there is only one state for each activity, and there will be no non-renewable resource, the problem will turn into a standard RCPSP problem. It should be mentioned that including a non-renewable resource there is a chance that we couldn't have any feasible scheduling, and this has been indicated by Kolish and Drexl [3]. The problem of finding a feasible scheduling in this problem is a kind of NP-Complete problem itself (in case there exist at least two non-renewable resources considered for the problem). According to Broker et al. symbolization, parameter α will be started with MPS [4].

* Corresponding author:

sina.namazi.ie@outlook.com (Sina Namazi)

Published online at <http://journal.sapub.org/als>

Copyright © 2014 Scientific & Academic Publishing. All Rights Reserved

2. Proposed Bi-objective MRCPSP Problem

According to the literature that has been reviewed, the proposed model has been considered in Bi-objective state [5], multi-mode state, with renewable or non-renewable resources and dual resources which is a particular state of resources.

Also for the goals, two well-known and practical goals has used which the first one is minimizing project makespan and the second one is minimizing project total cost.

The mathematical model for the proposed problem is as follows:

$$\text{Min } F_1 = \sum_{t=es}^{ls_n} t \cdot x_{n1t}$$

$$\text{Min } F_2 = \sum_{i \in A} \sum_{m \in M_i}^n c_{im} \cdot X_{imt}$$

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

$$\sum_{m=1}^{M_i} \sum_{t=es_i}^{ls_i} (t + d_{im_i} + FS_{ij}) X_{im_i,t} \leq \sum_{m=1}^{M_j} \sum_{t=es_j}^{ls_j} t x_{jm,t} \quad (i, j) \in E_{fs}$$

$$\sum_{i=1}^n \sum_{m=1}^{M_i} r_{imk} \sum_{s=\max\{t-d_{im}, es_i\}}^{\min\{t-1, ls_i\}} x_{ims} \leq \alpha_k$$

$$\sum_{i=1}^n \sum_{m=1}^{M_i} r_{imk}^g \sum_{s=es_i}^{ls_i} x_{ims} \leq \alpha_k^g$$

$$x_{imt} \in \{0,1\} \quad m_i = 1, \dots, M \quad i = 1, \dots, n ; t = 1, \dots, T$$

Constraint 1 is like a basic MRCPSP problem which causes the activity to be started in any executive mode and specified time that had been chosen and other modes will not be used until the end of the execution.

Constraint 2 is the end-to-start precedence relations constraint. If this constraint is used between activities **i** and **j**, activity **j** won't start unless the activity **i** finished.

Constraint 3 is the constraint of renewable resources which is the basic model of MRCPSP.

Constraint 4 is the constraint of non-renewable resources which is the basic model of MRCPSP.

Constraint 5 is the constraint of decision variable being binary.

Now we should mention two important notes about project cost function and dual-resource constraint. According to the fact that the cost constraint of each activity is a kind of inborn non-renewable resource, so consideration of this constraint has been ignored and it is considered in non-renewable resource constraint, which completely discussed in sample problems posed for this proposed problem [4].

According to the definition of dual-resource constraint which mentioned before [4], some items had been considered such as project budget. If the project finance won't be available at the beginning of the project, and some amount of the budget will be allotted to the managers periodically, in this case, the project budget is a kind of dual-resource. On the one hand it's a non-renewable resource since the project budget has a specific amount and it will be finished; on the other hand it must be considered a renewable resource because some amount of it is available in each period. So for each bi-resource in this proposed problem, two more renewable & non-renewable constraints must be considered, which is also completely described in sample problems posed for this proposed problem.

3. Proposed Algorithm to Solve the Proposed Problem

ICA (Imperialist Competitive Algorithm) is a method in the field of evolutionary computation which is used to find the optimum solution in different optimization problems. This algorithm represents an algorithm to solve the mathematical optimization problems with the use of mathematical modeling of the social-political evolution process. From the usage perspective, this algorithm is grouped with evolutionary optimization algorithms like genetic algorithms, ant colony optimization, simulated annealing and etc. like all algorithms of this group, ICA algorithm forms a basic set of probable answers. These basic answers are known as "chromosomes" in genetic algorithm, "particle" in particle swarm algorithm and "country" in ICA algorithm. ICA algorithm improves these basic answers (countries) gradually with a special method which will be discussed later, and finally will find the appropriate answer (desired country) for the optimization problem.

In the beginning, the steps of the applied algorithm are presented to solve the proposed RCPSP problem. The process of the algorithm for these kinds of problems is as follows:

Step 1: in this step, a series of possible orders considering the constraints (precedence consideration and resource constraint will be created).

The process is as follows: at first, for each project activity,

a random number between 0 and 1 is produced. Then these random numbers are arranged and saved in a list. The sequence number of these activities are saved in another list.

Step 2: the results are arranged depending on their suitability and the first 10 results are chosen as imperialist countries. Then the normalized cost and power of an imperialist is defined by

$$C_n = \max_i \{C_i\} - C_n \quad (1)$$

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (2)$$

Step 3: the normalized power of an imperialist is the portion of colonies that should be possessed by that imperialist. Then the number of colonies of an empire will be

$$NC_n = \text{round} \{P_n \times N_{col}\} \quad (3)$$

Step 4: the best imperialist is saved in **BestSol**. The major circle of the algorithm, steps 5 to 7, is repeated for a specific number of times (here, the number of repetitions is the criteria to make the algorithm to stop).

Step 5: imperialists and their colonies are invoked. Colonies move toward imperialists on the basis of this formula

$$B_2 = \text{Rand} \times (A - B_1) \times 1 \angle \theta \quad (4)$$

After the movement, if a colony is more powerful than its imperialist, the colony and the imperialist change their places.

Step 6: the operation in step 5 is done for all imperialists. The power of imperialists is calculated one more time. A pre-determined number of colonies go under the control of the most powerful imperialist.

Step 7: this is the last step. Imperialists are arranged depending on their suitability and the best imperialist is saved in **bestSol**.

4. The Results of the Proposed Imperialist Competitive Algorithm for MRCPSP

In this section, RCPSP problems with multi-mode (multi-state) activities are being considered. There are j10, j12, j14, j16, j18, j20 and j30 sets in the library of these problems. One more feature of MRCPSP problems is that beside the renewable resources, they have non-renewable resources as well.

Problems with names of c15, c21, m1, m2, m4, m5, n0, n1, n3, r1, r3, r4 and r5 can also be found among the standard MRCPSP problems which have some differences in the number of resources and activities. For instance, group c problems have 18 activities, 2 renewable resources and 2 non-renewable resources. Group m problems have also 18 activities, 2 renewable resources and 2 non-renewable resources. Group n0 problems have 14 activities and only 2 renewable resources and n1 problems have 18 activities, 2 renewable resources and 1 non-renewable resource, and n3 problems also have 18 activities with 2 renewable and 3 non-renewable resources. Group r problems also differs in the number of renewable resources in the way that r1 problems have 18 activities with 1 renewable and 2 non-renewable resources, r3 have 3 renewable and 2 non-renewable resources, r4 have 4 renewable and 2 non-renewable resources and r5 have 5 renewable and 2 non-renewable resources [6].

In this study, only group j problems have been used for comparison. Between group j problems, j10, j12, j14, j 20 and j30 problems has been used. These problems have 2 renewable and 2 non-renewable resources.

In the table (1), the average deviation from the best answers in j10 up to j20 problem series is calculated, which the optimum answer is available. So the average deviation from these optimum answers has been compared in the table (1), using the best and newest ever-applied algorithms, in which the proposed algorithm holds the 3rd place in this study.

Table 1. Results ICO algorithm for MRCPSP problem

Av Dev %	J10	J12	J14	J16	J18	J20
SFLA [7]	0.10	0.21	0.46	0.58	0.94	1.14
EFEA [8]	0.14	0.24	0.77	0.91	1.30	1.62
This paper	0.13	0.25	0.66	0.83	1.36	1.62
RSS [9]	0.18	0.65	0.89	0.95	1.21	1.64
AGA [10]	0.24	0.73	1.00	1.12	1.43	1.91
JSA [11]	1.16	1.73	2.60	4.7	5.52	6.74

The second criteria which is evaluated in these kinds of problems is the percentage of the optimum answers and another one is the process time of the CPU, which has been evaluated in the table (2).

Table 2. Result CPU time for solving MRCPSP problem with ICO algorithm

Optimal rate % / CPU time (sec)	J10	J12	J14	J16	J18	J20
SFLA [7]	99.8 / 0.035	99.4 / 0.039	98.9 / 0.13	95.6 / 0.21	94.7 / 0.46	90.6 / 0.91
EFEA [8]	99.6 / 0.045	99.6 / 0.045	99.4 / 0.11	94.3 / 0.26	91.09 / 0.54	89.89 / 1.11
This paper	99.5 / 0.043	99.5 / 0.045	98.88 / 0.15	95.00 / 0.27	92.03 / 0.55	89.80 / 1.26
RSS [9]	99.1 / 0.066	98.8 / 0.10	97.76 / 0.23	97.00 / 0.36	90.01 / 0.66	85.40 / 1.675
AGA [10]	98.8 / 0.088	93.87 / 0.11	95.60 / 0.23	91.01 / 0.33	86.65 / 0.55	82.35 / 1.90
JSA [11]	96.1 / 0.800	90.68 / 1.10	88.65 / 1.45	86.00 / 1.66	81.12 / 1.78	80.02 / 2.66

5. The Results of the Proposed ICA for Solving the Proposed Problem

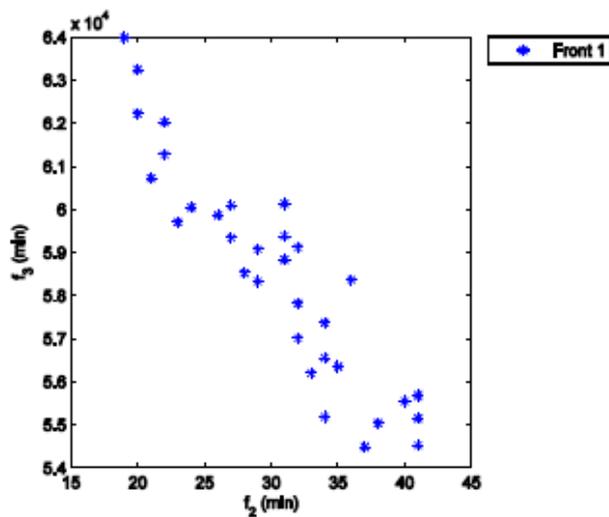


Figure 1. Pareto Answers with ICO algorithm

According to the fact that a bi-objective model has been developed for RCPSP problems in this study, at first a

problem is solved as an instance in order to show the algorithm performance in producing pareto answers in the form of figure (1). It is obvious from the chart that the algorithm could produce the suitable pareto to solve this sample problem which the answers and diversity are desirably suitable.

The algorithm has been used to solve some bi-objective problems from the MRCPSP problems and results has been evaluated through 2 important criteria namely diversity and spacing metric to check the quality of the produced pareto answers in multi-objective problems.

1- Diversity Criterion

Zitzler defined a specific measurement criterion in which the length of a diameter is measured. This diameter is the diameter of the space used by final value of the objects for pareto answer set [12].

2- Spacing Metric criterion

It's another criterion for checking pareto's charts. This criterion studies distance of the answers from each other in order to finds the distance between answers of one point with its nearest neighbor. This criterion could be considered either ordinarily or normalized. Formulas of each group are brought later.

6. Checking the Performance of the Proposed Algorithm from the Point of Evaluation Criteria with PSP Algorithm

According to the discussed criteria in previous section, the algorithm is being used to solve different MRCPSp problems and its results have been compared with PSO algorithm which the results of these two criteria are brought in the table(3). At the end, two algorithms are being evaluated from the aspect of problems' solving time. Results indicated satisfactory performance of the ICA algorithm.

Table 3. Compare ICO algorithm with PSO algorithm

Algorithms \ Evaluating facts	Proposed problem	First criterion (diversity)	Second criterion (spacing metric)
This paper ICO algorithm	101-1	88651.11	32
	J122-4	90111.74	30
	J145-6	76245.653	28
	J181-5	893112.244	31
	J201-1	84356.543	33
PSO algorithm	J101-1	82451.21	30
	J122-4	92111.76	30
	J145-6	71245.564	28
	J181-5	793442.145	32
	J201-1	76436.455	28

7. Numerical Example for Showing ICA Algorithm Performance in Problem Solving

Table 4. Colonial power and the number of samples in problem solving

Imperial	1	2	3	4	5	6	7	8	9	10
Cost	21	26	26	27	28	30	31	31	32	33
power	0.266667	0.155556	0.155556	0.133333	0.111111	0.0666	0.04444	0.0444	0.022	0
NC	24	14	14	12	10	6	4	4	2	0

Table 5. The best example of elegance in problem solving

Imperial	1	2	3	4	5	6	7	8	9	10
Cost	21	24	25	26	26	26	31	31	32	33
power	0.266667	0.16	0.16	0.155556	0.155556	0.155556	0.044444	0.044444	0.022222	0
NC	34	14	14	12	6	6	4	0	0	0

To show the algorithm performance better, we're using a sample example for step-by-step functionality of ICA algorithm.

Problem definition is stated in bee's algorithm section. So in this section, we'll only discuss about the functionality of ICA algorithm.

10 imperialist are being chosen at first. Propriety, power and number of colonies will be specified for each imperialist and brought in the table (4).

As it is obvious from the table, imperialist number 1 has the best propriety and owns the most colonies. This imperialist will be stored in BestSol as the best answer.

After that each colony changed toward its imperialist, the power of each imperialist will be calculated again and the best imperialist will be specified. The results are presented in the table (5).

As it is obvious, imperialist number 1 has the best

propriety and the most empire power. So 10 weakest colonies will be added to the colonies of this imperialist. This imperialist will be stored in BestSol as the best result. In addition, as it is clear, empires 8, 9 and 10 fail, because the number of their colonies become zero.

This process will be repeated for a specific number of times. On more important criterion for this algorithm to stop is when there is only one imperialist and all other empires had fail.

8. Conclusions

In this paper, a bi-objective problem was suggested for MRCPSp problem series, and then MRCPSp problem series and a proposed problem were solved with ICA algorithm. Results of MRCPSp problem solution have compared with the newest algorithm used to solve these series of problems

and the results of proposed problem have compared with PSO algorithm which the results indicated a successful performance of this algorithm to solve these kinds of problems. This algorithm can be used to solve other RCPSP problems in future researches.

REFERENCES

- [1] Using Bees Algorithm to Solve the Resource Constrained Project Scheduling Problem in PSPLIB. Amir Sadeghi, Abolfazl Kalanaki, Azadeh Noktehdan, Azamdokht Safi Samghabadi and Farnaz Barzinpour, *Theoretical and Mathematical Foundations of Computer Science Communications in Computer and Information Science*, 2011, Volume 164, 486-494.
- [2] Activity networks: project planning and control by network models, Elmaghraby, S.E. 1977. Wiley.
- [3] Adaptive search for solving hard project scheduling problems, Rainer Kolisch* and Andreas Drexl, *Naval Research Logistics (NRL)*, February 1996, Volume 43, Issue 1, pages 23-40.
- [4] Resource-constrained project scheduling: Notation, classification, models, and methods, Brucker, Drexl A, Mohring R, Neumann K, Pesch E. *European Journal of Operational Research*, 1999, 112,3-41.
- [5] Bi-objective resource-constrained project scheduling with robustness and makespan criteria, B. Abbasi, S. Shadrokh, *Applied Mathematics and Computation*, 2006, J. Arkat 180 (1), 146-152.
- [6] A random key based genetic algorithm for the resource constrained project scheduling problem, Mendes JJ, Goncalves JF, Resende MG., *Computers and Operations Research* 2009;36(1),92-109.
- [7] An efficient hybrid algorithm for resource-constrained project scheduling, Chen W, Shi YJ, Teng HF, Lan XP, Hu LC. *Information Sciences* 2010;180(5),1031-9.
- [8] A decomposition-based genetic algorithm for the resource-constrained project scheduling problem, Debels D, Vanhoucke M. *Operations Research* 2007;55(3),457-6.
- [9] A hybrid genetic algorithm for the resource-constrained project scheduling problem, Valls V, Ballestin F, Quintanilla S. *European Journal of Operational Research* 2008; 185(2), 495-508.
- [10] A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem, 2012, *Expert systems and application*, 39,2451-2460.
- [11] A neurogenetic approach for the resource-constrained project scheduling problem, Agarwal, A., Colak, S., & Erenguc, S. *Computers & Operations Research*, 2011, 38, 44-50.
- [12] Quality assessment of Pareto set approximations. In *Multiobjective Optimization*. Zitzler, E., Knowles, J., Thiele, L., Springer Berlin Heidelberg, 2008. 373-404.