Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 17, No. 3, September 2017, pp. 221-227
http://dx.doi.org/10.5391/IJFIS.2017.17.3.221

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

A Robust Solution to Resource-Constraint Project Scheduling Problem

Milad Keshavarz Kord Mahalleh[1], Behzad Ashjari[1], Fereshteh Yousefi[2], and Morteza Saberi[3]

[1]Department of Industrial Engineering, Tafresh University, Tafresh, Iran
[2]Department of Industrial Engineering, University of science and technology, Nur, Iran
[3]School of Business, UNSW Canberra, Australia

Abstract

This paper aims to propose a solution to the resource-constraint project scheduling problem (RCPSP). RCPSP is a significant scheduling problem in project management. Currently, there are insufficient studies dealing with the robustness of RCPSP. This paper improves the robustness of RCPSP and develops a Robust RCPSP, namely RRCSP. RRCSP is structured with relaxing a fundamental assumption that is 'the tasks start on time as planned'. Relaxing this assumption makes the model more realistic. The proposed solution minimizes the makespan while maximizing the robustness. Maximizing the robustness requires maximizing floating time of activities (it is NP hard). This creates more stability in the project finishing time. RCPSP stands as the root cause of many other problems such as multi-mode resource-constrained project scheduling problems (MRCPSP), multi-skill resource-constrained project scheduling problem (MSRCPSP), or similar problems and hence proposing a solution to this problem contributes to pave a new line for future research in other mentioned areas. The applicability of the proposed model is examined through a numerical example.

Keywords: RCPSP, Resource-constraint, Project scheduling, Differential evolution

1. Introduction

Resource-constraint project scheduling problem (RCPSP) is a significant scheduling problems in project management [1, 2]. With regard to this problem, the time and resource estimation of project activities are considered as a critical task. This is because activities are subjected to delay with regard to uncontrollable factors that may increase their durations [3]. The current literature is enriched with models that have been proposed by researchers [4–6].

Since decades ago, scheduling problem in real project has been treated as a multi-objective problem [7–9]. This paper is developed based on two papers by Al-Fawzan and Haouari [3] and Abbasi et al. [10]. Al-Fawzan and Haouari [3] proposed a bi-objective model in which two significant objectives were taken into account: 'makespan minimizing' and 'robustness maximizing'. The model was then solved using multi objective tabu search (MOTS). In the following year, Abbasi et al. [10] solved the same problem using simulate annealing (SA) based approach. Aligned with those papers, in this paper, a single renewable resource and bi-objective model is considered and addressed using differential evolution (DE) algorithm [11].

The paper provides a brief section on the importance of robustness in scheduling problem.

Received: Aug. 16, 2017
Revised : Sep. 3, 2017
Accepted: Sep. 3, 2017

Correspondence to: Morteza Saberi
(m.saberi.post@gamil.com)

Next, the DE algorithm is explained. Following that, the model is proposed. Then, the model is tested through an illustrative example. The results are compared in particular with the results of the paper proposed by Abbasi.

2. Robustness Importance

Before starting a project, it often happens that the required time for activities as well as the resources of the project are estimated according to the project manager's experience. But after starting project the activity's required time and resources may dramatically change due to uncontrollable factors and incidents. This may increase the duration or activities or may require unavailable resources during the project. In such incidents, the time of activities and consequently the makespan will be delayed. Using a robust scheduling model, we reduce the consequences of such incidents. Robust scheduling generally means that if activities were delayed, the makespan would not be delayed that is in terms of the floating time concept. Icmeli-Tukel and Rom [12] used rework concept to introduce robust scheduling. Al-Fawzan and Haouari [3] provided a method for robustness concept and used MOTS in order to find global optimum solution using local right shift (LRS). In contrast, this paper uses global right shift (GRS) method for robustness relevant calculated rather than LRS. The reason for this choice is because activities that cannot be delayed in LRS, can be delayed in GLS.

Six figures are provided below. Figure 1 shows a simple network project CPM. Figure 2 shows a solution of this example when the available resources is 6 units. As we can see in the Figure 2, activity B cannot have a delay in LRS method while using GRS it can. Figures 3 and 4 present two solutions to the mentioned example when the available resource is 7 units and makespan of both of them is equal to 9 units but their summation floating time is different. The summation of floating time in Figure 3 is 8 units and in Figure 4 it is 10 units. It can be argued that because the summation of floating time of Figure 4 is more than the summation floating time of Figure 3, solution four is more robust than solution three. Because, for example, if starting time of activities B and E delay for 7 and 3 units, due to mentioned uncontrollable and unexpected factors and incidents, according to the sequence that project manager has provided, the makespan of solution three according to Figure 5 is equal to 13 and makespan of solution four according to Figure 6 remains unchanged. This is a simple example that explains the importance of the robust scheduling.

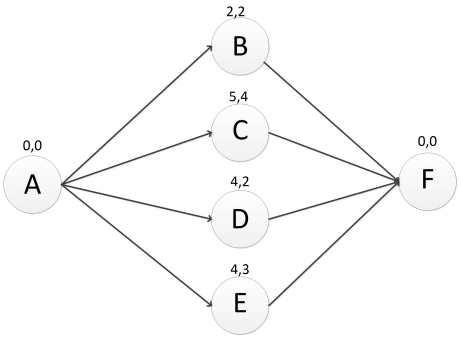


Figure 1. Network example.

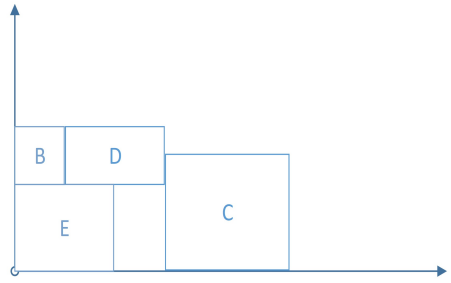


Figure 2. Solution with 6 available resource unit.

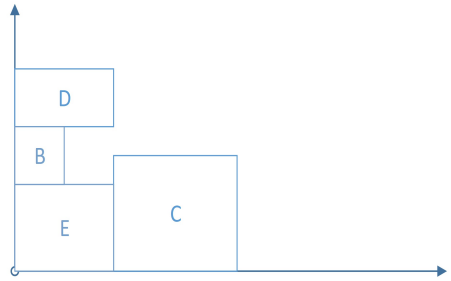


Figure 3. The first solution.

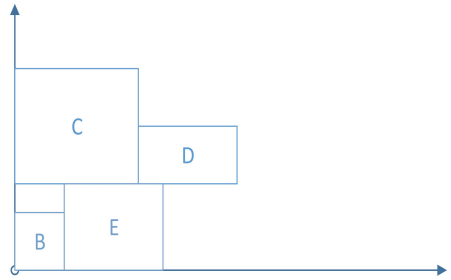


Figure 4. The second solution.

3. Differential Evolution

DE algorithm as an evolutionary algorithm was proposed by Storn and Price [13]. Despite the similarities of DE to evo-

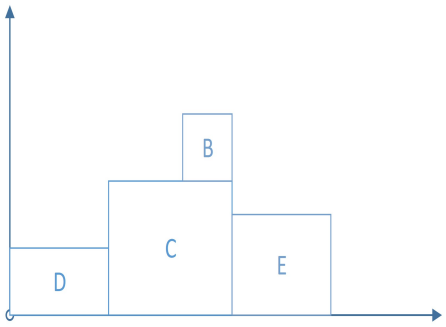


Figure 5. The first solution after the delay.

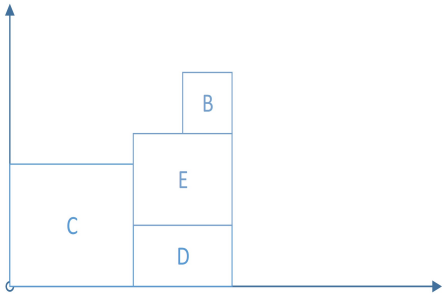


Figure 6. The second solution after the delay.

lutionary algorithms, it has unique method in generation new population. The evolutionary algorithms usually generate new population by crossover and mutation where applicable a mixture of both of them starting by application of crossover and continued by application of mutation.

DE differentiates itself from other evolutionary algorithms from two reasons. First, it begins by applying mutation generated initial solution and then with it applies crossover to generate new solution. Second is that the method of mutation that is applied by a distance and difference from the available solutions in population.

3.1 The Steps of Differential Evolution

- Defining of the parameters, problem, and the algorithm
- Generating the initial population and their evaluation member
- Repeating the following steps while the termination condition is not realized
  - Repeating the following steps for every member of the population
    - Generating temporary solution by mutation operation
    - Evaluating a new solution using crossover operation
    - Replacing the old one with the new one if new solution is better than the old solution. Otherwise the old solution remains untouched.
- The best solution is provided.

4. The Model

By considering makespan and robustness, RRCPSP is modelled as follows:

Parameters:

$n =$ number of activity,
$G =$ acyclic graph that represents the project,
$d_j =$ duration of activity j,
$ES_j =$ earliest start time of activity j,
$LS_j =$ latest start time of activity j,
$EF_j =$ earliest finish time of activity j,
$LF_j =$ latest finish time of activity j,
$P_j =$ set of precedence of activity j,
$S_j =$ set of successor of activity j,
$R =$ value of renewable sources,
$r_j =$ resource requirements for activity j,
$C_{max} =$ finish time of whole project,
$FT =$ floating time of whole project,
$FT_j =$ floating time of activity j,
$X_{jt} = 1$, if activity j in time t is executed and zero, otherwise.

Model:

$$\min f(\text{makespan}(C_{max}), \text{floating time}(FT))$$

subject to

$$\sum_{t=EF_J}^{LF_J} X_{jt} = 1 \; \forall \; j = 1, 2, ..., n, \tag{1}$$

$$\sum_{t=EF_w}^{LF_w} t.X_{wt} \leq \sum_{t=EF_j}^{LF_j} (t - d_j) \, .X_{jt}, \; \forall \; j, w \in P_j, \tag{2}$$

$$C_j = \sum_{t=EF_j}^{LF_j} t.X_{jt}, \; \forall \; j = 1, 2, ..., n, \tag{3}$$

$$C_{max} \geq C_j, \; \forall \; j = 1, 2, ..., n, \tag{4}$$

$$\sum_{j=1}^{n} r_j . \sum_{b=t}^{t+d_j-1} X_{jb} \leq R, \ \forall \, t, \tag{5}$$

$$FT_j = LS_j - ES_j, \ \forall \, j = 1, 2, ..., n, \tag{6}$$

$$FT = \sum_{j=1}^{n} FT_j, \tag{7}$$

$$ES_1 = 0, \tag{8}$$

$$EF_i = ES_i + d_j, \ \forall \, j = 1, 2, ..., n, \tag{9}$$

$$ES_j = \max\{EF_i\}, \ \forall \, i \in P_i, j = 1, 2, ..., n, \tag{10}$$

$$LF_n = C_{max}, \tag{11}$$

$$LS_j = LF_j - d_j, \ \forall \, j = 1, 2, ..., n, \tag{12}$$

$$LF_j = \max\{LS_j\}, \ \forall \, i \in S_j, \ j = 1, 2, ..., n, \tag{13}$$

$$X_{jt} = \{0, 1\}, \ \forall \, j, t. \tag{14}$$

The objective function is approximated using a linear combination of makespan and sum of the floating time. $(\lambda * C_{max}) - (1 - \lambda) * (FT)$.

The value of λ, within $[0 \ \ 1]$ interval, determines the importance of makespan versus floating time and is very dependent on the project type changes. As observed above, the algorithm objective is the minimization type thus for maximizing FT in top objective, $(-FT)$ is used in the approximating process. In fact, we minimize C_{max} and maximize FT in the objective function.

The first constraint ensures that activity j will finish between EF_j and LF_j. The second constraint ensures priority of activities. Constraint (3) calculates finish time of activities and constraint (4) calculates finish time of project. Constraint (5) considers conditions resource constraint that is applicate to all part of the project. Constraint (6) calculates TF for each activity. Constraint (7) calculates TF for the whole project. Constraint (8) show that project starts at zero time and constraint (9) calculates the latest start time of each activity. Constraint (10) calculates the earliest time of each activity. Constraint (11) before Eq. (13) completes calculates the latest start and finish time of each activity. Finally, constraint (14) determine the amount of decision variable so that if equals it is equal to 1, activity j in time t applies and it zero, otherwise.

As RRCPSP is the extension of RCPSP, and so it is also considered as NP-hard problem.

Beginning we start coding a program that generates a feasible solution by considering resource constraint and precedence constraints to calculate makespan is needed. Then, we use floating time base GRS method to obtain value of robustness. Hence we needed difference earliest start time and the latest start time of each activity. That's why for obtaining the latest start time of each activity, we perform coding a program and we set all of the activities of that sequence solution by considering the newest first so that it starts from finishing project time (makespan) and new sequence will set towards the time zero and in this way all of the activities exceed their durations that they could possibly get and we may obtain the latest start and latest finish time of each activity.

Finally, we obtain makespan and robustness value to each solution then we can calculate top objective and use it in DE explained previous section.

5. Numerical Example

To illustrate the method, a numerical example is used. The example is addressed using MS project 2016. At the end, the results are compared with the work presented by Abbasi *et al*. (2006).

Assume that there are 50 activities and there is one continuous resource with 6 units over the project. The durations of activities, requirement resources, and their precedence are ordered by the numbers of activities as in Table 1.

After solving this example by the model, using DE algorithm and DE parameters algorithm, set with MATLAB R 2016a and CPU core i5-2430M and 4 RAM, the results are as follows (presented in Table 2):

Maximum number of iteration=100

Population size=50

Lower bound for scaling factor to mutation= 0.2

Upper bound for scaling factor to mutation=0.5

Crossover probability=0.4

Note that when the value of λ is very low, we may fail to find a proper solution. Because the difference between makespan and floating time value is relatively large, may with increase or decrease of makespan have no effect in objective when λ is smaller than above value. So finding optimum λ needs application of optimization algorithm.

The example is solved using MS project 2016 and the results are presented in Table 3.

Table 4 presents the results from the paper by Abbasi et al. [10]. They used LRS to calculate the time of activities.

As can be seen in Tables 2-4, makespan and floating time of our model is better than other solution that this subject shows efficiency of our model.

Table 1. Duration and requirement resources of each activities

Activity	Duration	Requirement resources	Precedence	Activity	Duration	Requirement resources	Precedence
1	2	2	-	26	3	2	14
2	2	1	1	27	6	3	14
3	5	3	1	28	3	3	16
4	4	1	1	29	2	1	16
5	2	1	3	30	3	2	17
6	3	2	2	31	1	1	20
7	1	1	-	32	2	2	20
8	3	2	7	33	5	3	20
9	5	3	7	34	4	2	21
10	6	1	7	35	3	2	21
11	4	3	7	36	1	2	22
12	2	1	8	37	1	1	22
13	3	1	-	38	2	1	22
14	5	2	9	39	6	2	22
15	6	2	9	40	4	1	25
16	7	1	10	41	3	2	25
17	4	1	-	42	1	3	29
18	2	1	10	43	5	1	29
19	5	1	10	44	3	2	30
20	4	1	11	45	6	1	31
21	1	2	11	46	2	1	31
22	3	2	12	47	1	2	32
23	2	1	12	48	2	1	32
24	1	3	12	49	1	2	33
25	2	1	13	50	2	1	45

Table 2. Results using the method

λ	Makespan	Floating time
1	46	436
0.995	46	1011
0.990	46	1039
0.985	46	1043
0.980	46	1072
0.975	47	1092
0.970	47	1104
0.965	47	1128
0.960	47	1155

Table 3. Results using MS project 2016

Makespan	Floating time
49	358

6. Conclusion

This paper has proposed a solution to RCPSP as a significant scheduling problem in project management. This paper has improved the robustness of RCPSP and developed a Robust RCPSP (RRCSP). The proposed solution to the scheduling problem could minimize the makespan while maximizing the robustness. This created more robustness and stability in the project completing time. The proposed solution to the problem contributes to pave a new line for future research. The applica-

Table 4. Results of the study by Abbasi et al. [10]

λ	Makespan	Floating time
1	47	82
0.995	47	115
0.990	47	115
0.985	47	115
0.980	51	343
0.975	52	389
0.970	52	389
0.960	66	841
0.950	66	841

bility of the proposed model was examined through a numerical example.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

References

[1] C. Fang, R. Kolisch, L.Wang, and C. Mu, "An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem," *Flexible Services and Manufacturing Journal*, vol. 27, no. 4, pp. 585-605, 2015.

[2] W. Ma, Y. Che, H. Huang, and H. Ke, "Resource-constrained project scheduling problem with uncertain durations and renewable resources," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 4, pp. 613-621, 2016.

[3] M. A. Al-Fawzan and M. Haouari, "A bi-objective model for robust resource-constrained project scheduling," *International Journal of Production Economics*, vol. 96, no. 2, pp. 175-187, 2005. https://doi.org/10.1016/j.ijpe.2004.04.002

[4] C. Artigues, P. Michelon, and S. Reusser, "Insertion techniques for static and dynamic resource-constrained project scheduling," *European Journal of Operational Research*, vol. 149, no. 2, pp. 249-267, 2003. https://doi.org/10.1016/S0377-2217(02)00758-0

[5] M. R. Asadabadi, "A developed slope order index (SOI) for bottlenecks in projects and production lines," *Computational Management Science*, vol. 14, no. 2, pp. 281-291, 2017.

[6] D. Debels, B. De Reyck, R. Leus, and M. Vanhoucke, "A hybrid scatter search/electromagnetism meta-heuristic for project scheduling," *European Journal of Operational Research*, vol. 169, no. 2, pp. 638-653, 2006. https://doi.org/10.1016/j.ejor.2004.08.020

[7] M. P. Hansen, "Tabu search for multiobjective optimization: MOTS," in *Proceedings of the 13th International Conference on Multiple Criteria Decision Making*, Cape Town, South Africa, 1997, pp. 574-586.

[8] M. Hapke, A. Jaszkiewicz, and R. Slowinski, "Interactive analysis of multiple-criteria project scheduling problems," *European Journal of Operational Research*, vol. 107, no. 2, pp. 315-324, 1998. https://doi.org/10.1016/S0377-2217(97)00336-6

[9] A. Nagar, J. Haddock, and S. Heragu, "Multiple and bicriteria scheduling: a literature survey," *European Journal of Operational Research*, vol. 81, no. 1, pp. 88-104, 1995. https://doi.org/10.1016/0377-2217(93)E0140-S

[10] B. Abbasi, S. Shadrokh, and J. Arkat, "Bi-objective resource-constrained project scheduling with robustness and makespan criteria," *Applied Mathematics and Computation*, vol. 180, no. 1, pp. 146-152, 2006. https://doi.org/10.1016/j.amc.2005.11.160

[11] S. K. Goudos, M. Deruyck, D. Plets, L. Martens, and W. Joseph, "Optimization of power consumption in 4G LTE networks using a Novel Barebones self-adaptive differential evolution algorithm," *Telecommunication Systems*, vol. 66, no. 1, pp. 109-120, 2017.

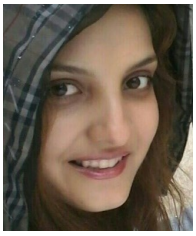
[12] O. Icmeli-Tukel and W. O. Rom, "Ensuring quality in resource constrained project scheduling," *European Journal of Operational Research*, vol. 103, no. 3, pp. 483-496, 1997. https://doi.org/10.1016/S0377-2217(96)00305-0

[13] R. Storn and K. Price, "Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.

Milad Keshavarz Kord Mahalleh is student of Industrial Engineering at Tafresh University. He obtained his BS in 2015 from Tafresh University. His MS has started in 2015 and his topic of thesis is multi objective resource allocation project scheduling by meta heuristic method.

Behzad Ashjari is Professor of Industrial Engineering at Tafresh University. He obtained his Ph.D. in 2000 from AmirKabir University of Technology for work on Resource Allocation in Multi-Project Planning. He has published several papers on mathematical modeling and simulation in International Journal of Project Management. His research interests are in mathematical modeling of man power planning and application of operational research in maintenance and SCM.

Morteza Saberi has an outstanding research records and significant capabilities in in area of Business Intelligence, Data Mining and applied machine learning. He has published more than 140 papers in reputable academic journals and conference proceedings. His Google Scholar citations and h-index are 1400 and 18 respectively. He was a Lecturer at the Department of Industrial Engineering at University of Tafresh. He is also the recipient of the 2006-2012 Best Researcher of Young researcher Club, Islamic Azad University (Tafresh Branch). He is also the recipient of National Eminent Researcher Award among Young researcher Club, Islamic Azad University members.



Fereshteh Yousefi is student of Industrial Engineering at University of science and technology. She obtained her BS in 2015 from Tafresh University. Her MS has started in 2015 and his thesis is about resource constraint project scheduling problem with consideration multi skill manpower.