

# Using Multi-Objective Particle Swarm Optimization for Bi-objective Multi-Mode Resource-Constrained Project Scheduling Problem

FatemehAzimi<sup>1</sup>, RazeehSadatAboutalebi<sup>2</sup>, Amir Abbas Najafi<sup>3</sup>

**Abstract**—In this paper the multi-mode resource-constrained project scheduling problem with discounted cash flows is considered. Minimizing the makespan and maximization the net present value (NPV) are the two common objectives that have been investigated in the literature. We apply one evolutionary algorithm named multi-objective particle swarm optimization (MOPSO) to find Pareto front solutions. We used standard sets of instances from the project scheduling problem library (PSPLIB). The results are computationally compared respect to different metrics taken from the literature on evolutionary multi-objective optimization.

**Keywords**—Evolutionary multi-objective optimization Makespan, Multi-mode, Resource constraint, Net present value

## I. INTRODUCTION AND LITERATURE REVIEW

**R**ESOURCE constraint project scheduling problem (RCPSp) is a class of project scheduling that activities should be scheduled subject to precedence and resource constraints and it is proven to be NP-hard [1]. It has widely studied in the literature and various extensions of basic RCPSp have been developed. For an overview of these extensions refer to [2, 3].

The multi-mode resource-constrained project scheduling problem which is known as MRCPSp is a generalization of single-mode version which is RCPSp and is more realistic model.

Several authors have used exact and heuristic procedures to solve this problem. According to Jozefowska et al. [4] exact methods are unable to find optimal solution in reasonable computation time. Different heuristic approaches have been used to deal with this problem. Jozefowska et al. [4] and Bouleimen et al. [5] used Simulated Annealing algorithm to solve MRCPSp. Alcaez et al [6], Hartmann [7] and Mori,

Teseng [8] proposed different genetic algorithms which Mori and Teseng considered only renewable resources. Jarboui et al. [9] also used PSO algorithm to solve this problem. In this paper, we present a new heuristic solution procedure to solve MRCPSp. We consider solving the problem with two objectives. The first objective is minimizing the makespan and the second one is maximizing the net present value. Minimizing the makespan is the most popular objectives in the literature [10]. Recent surveys on is maximizing the net present value in MRCPSp are [11], [12] and [13]. Ulsoy et al. [12] used GA approach to solve four payment models for solving MRCPSpDCF which we use their delayed scheduling in our model to maximize the net present value.

Taking this problem into account as the multi-objective problem has not been well studied. Several authors have used different ways to cope with multiple objectives. Nudtasomboon and Randhawa [14] define one overall objective as the weighted sum of all performance measures considered. They used various objectives such as makespan, weighted tardiness, resource leveling and usage of nonrenewable resources. Vob and Witt [15] define an objective that contains makespan, weighted tardiness and set up costs. Using generation of Pareto-optimal schedules is another way to deal with multiple objectives. This is done by Slowinski et al. [16] Nabrinski and Welgarz [17] present a knowledge based approach to a project scheduling problem with multiple modes.

The use of evolutionary algorithms for multi objective optimization has significantly grown in last few years [18]. We used Multi objective particle swarm optimization to solve this problem. The rest of this research is organized as follow:

In section two we will consider the generalized formulation of MRCPSp and we mathematically formulated the problem. In Section three and four you will see the application of the proposed meta-heuristic: MOPSO. Computational results and the analyses for the results are given in section five and finally some conclusions are given in section six.

<sup>1</sup> Department of Mathematics, Islamic Azad University, Qazvin Branch, Qazvin, Iran (Phone: +98 2813665275; Email: ff\_azimi@yahoo.com).

<sup>2</sup> Faculty of Industrial and Mechanical Engineering, Islamic Azad University, Qazvin Branch, Qazvin, Iran (Phone: +98 2813665275; Email: razeeh.abutalebi@gmail.com).

<sup>3</sup> Faculty of Industrial Engineering, K.N. Toosi University of Technology, Tehran, Iran (Phone: +98 2184063378; Fax: +98 2188674858; Email: aanajafi@kntu.ac.ir).

## II. MODEL

Consider an activity-on-node project which consists of  $V=\{1, 2, \dots, n\}$  activities. Activities 1 and  $n$  are dummy activities and represent the start and completion of the project.

Each activity  $j \in V$  can be performed in one of its mode given by the set  $\{1, \dots, M_j\}$ . Each mode represents resource requirement and activity duration. The duration of activity  $j$  when executed in mode  $m$  is  $d_{jm}$  and all of the activities must be done without interruption. For each activity  $j$ , the precedent activities set is denoted as  $P(j)$ .

Activities 1,  $n$  have duration equal to zero and don't need any resources in their unique modes. The set of renewable resources is denoted as  $R_k$  and activity  $j$  performed in mode  $m$  requires  $r_{jkm}$  units of renewable resource  $k$ . The cash flow of each activity is represented by  $CF_j$ . Cash outflows are included by the execution of activities and usage of resources while cash inflows result from payments due to completion of specified parts of project. Furthermore, we assume that the discount factor is  $\alpha$ . Now the model can be formulated as follow:

$$\text{Min} \sum_{t=ES_n}^{LS_n} t \cdot X_{nt} \quad (1)$$

$$\text{Max} \sum_{i=2}^{n-1} \sum_{m=1}^{M_i} \sum_{t=ES_i}^{LS_i} CF_i \cdot e^{-\alpha t} \cdot X_{imt} \quad (2)$$

Subject to:

$$\sum_{m=1}^{M_i} \sum_{t=ES_i}^{LS_i} X_{imt} = 1 \quad ; i=1, 2, \dots, n \quad (3)$$

$$\sum_{m=1}^{M_i} \sum_{t=ES_i}^{LS_i} (t + d_{im}) X_{imt} \leq \sum_{m=1}^{M_j} \sum_{t=ES_j}^{LS_j} t \cdot X_{jmt} \quad ; \forall i \in P(j) \quad (4)$$

$$\sum_{i=2}^{n-1} \sum_{m=1}^{M_i} r_{ikm} \sum_{s=\max\{t-d_{im}, ES_i\}}^{\min\{t, LS_i\}} X_{ims} \leq R_k \quad ; k=1, \dots, K \quad ; t=1, \dots, T \quad (5)$$

$$X_{imt} \in \{0, 1\} \quad ; i=1, \dots, n \quad ; m=1, \dots, M_i \quad ; t=ES_i, \dots, LS_i \quad (6)$$

Where  $ES_i$  and  $LS_i$  are the earliest and latest start time of activity  $i$  based on the modes. The objective is to determine a schedule  $(M, S)$  that minimize the project completion time and maximize the net present value and both the precedence relation between activities and resource constrained are satisfied.

The objective function (1) minimizes the completion time of the dummy end activity and thus the project duration. The objective function (2) maximizes the NPV of the project and it includes the net present value of positive and negative cash flows. Equation (3) assures that each activity can be done in exactly one mode and one start time. Constraint (4) represents precedence relations. Constraint (5) guarantees the availability of renewable resources and (6) states binary values on the decision variables which is equal to 1 if the activity  $i$  in mode  $m$  starts in time instant  $t$  and to be 0 otherwise. It is obvious that if each activity has only one execution mode, we obtain

the standard resource-constrained project scheduling problem. According to Schirmer [19] the multi-mode problem itself is NP-hard in the strong sense.

## III. SOLUTION ENCODING AND INITIAL POPULATION GENERATION

We have extended the representation designed by Alcaraz et al [6]. Each individual is composed of precedence list, mode assignment and scheduling mode. Component of every solution are as follows:

- $f$  is an ordered list activity. This list is a permutation of all the activities, activities are not allowed to repeat in this list.
- $m$  represents the mode assignment.
- $s$  indicates the scheduling mode which is a binary variable.

This coding structure named activity list with scheduling mode and mode assignment representation.

If the  $f$  is 0, we use non-delayed scheduling and each activity can appear in the list after all its predecessors. If the  $f$  is 1, we choose delayed scheduling that proposed by Ulsoy et al. [12]. In this scheduling no activity starts earlier than any of the activities residing at earlier loci on the chromosome. So start time never decrease with increasing position on the chromosome.

Now we describe a new procedure for our meta-heuristic method. This procedure works as follows:

For  $i=1$  to  $n_{pop}$  we generate different permutations of our activities. For each activity we produce random modes from its available modes and random scheduling modes.

For each chromosome we use parse solution procedure that works as follows: For each activity in the list if all its predecessors have scheduled, according to  $f$ , we start scheduling and choose feasible start times that satisfies the precedence relations and also resource constraints. We do this procedure for all the chromosomes. In this way the related schedule will always be feasible.

## IV. THE MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION (MOPSO)

Particle swarm optimization (PSO) is a heuristic search technique that was proposed in 1995 [20]. This algorithm inspired by choreography of a bird flock. The position of each particle changes according to its own experience based on social-psychological tendency to emulate success of other individuals. A swarm consists of a set of particles and each particle represents a potential solution.  $x^i[t]$  is the position of each particle that is defined by adding a velocity to a current position :

$$x^i[t+1] = x^i[t] + v^i[t+1] \quad (7)$$

That the velocity vector is defined as follow:

$$v^i[t+1] = wv^i[t] + c_1r_1(x^{i.best}[t] - x^i[t]) + c_2r_2(x^{g.best}[t] - x^i[t]) \quad (8)$$

Where  $x^{i.best}[t]$  is position of the best particle member of the neighborhood of the given particle,  $x^{g.best}$  is the best position of the best particle member of the entire swarm (leader),  $w$  is inertia weight,  $c_1$  is the cognitive learning factor and  $c_2$  is the social learning factor (usually defined as constants) and  $r_1, r_2 \in [0,1]$  are random values. In this work we use  $w=0.7$  and  $c_1, c_2 = 1.5$ .

#### Main algorithm

In case of the relative simplicity of PSO, Multi objective particle swarm optimization allows PSO algorithm to solve multi objective problems [21]. This algorithm is based on Pareto dominance and it considers every non dominated solution as new leader. Also this approach uses a crowding factor to filter out the list of available leaders. This algorithm works as follows [22]: First a swarm is initialized. Then a set of leaders is also initialized with the non-dominated particles from the swarm. This set is usually stored in an external archive. Then some sort of quality measure is calculated for all the leaders in order to select one leader for each particle of the swarm. At each generation for each particle a leader is selected and a flight is performed. Then the particle is evaluated and its corresponding  $x^{i.best}$  is updated. A new particle replaces its  $x^{i.best}$  particle usually when this particle is dominated or if both are nondominated with respect to each other. After all the particles have been updated, the set of leader is updated too. Finally the quality measure of the set of leaders is recalculated and this procedure is repeated for a certain number of criterions.

#### External repository

The main objective the external repository is to keep a record of non-dominated vectors found during the search process. The external repository consists of two components: the archive controller and the grid which are discussed in more details in [23]. The function of archive controller is to decide whether a certain solution should be added to archive or not and the mechanism of the grid is to produce well-distributed Pareto fronts.

### V. EXPERIMENTAL RESULTS

In this section, in order to have quantitative performance of our algorithm, first we consider two metrics for spacing and diversity of nondominated Pareto solutions and then we report the results of our algorithm.

#### Performance metric

##### 1. Spacing (sp)

Scott [24] proposed this metric to measure how well the solutions are distributed. This metric measures the (distance)

variance of neighboring vectors in non dominated vectors and it is defined as:

$$S = \frac{1}{n-1} \sqrt{\sum_{i=1}^n (\bar{d} - d_i)^2} \quad (9)$$

Where  $d_i = \min_j \left\{ \sum_{k=1}^m |f_m^i - f_m^j| \right\}$ ;  $i, j = 1, \dots, n$  and  $n$  is the number of non-dominated solutions generated by the algorithm;  $m$  is the number of objectives and  $\bar{d}$  is the mean of all  $d_i$ . Smaller  $S$  value corresponds to better performance of algorithm and a value of zero for this metric shows that all the non-dominated solutions found are equidistantly spaced.

#### 2. Maximum spread

According to Zitzler et al. [25], maximum spread is used to measure the diversity of the obtained non-dominated front. This metric is defined as:

$$D = \sqrt{\sum_{i=1}^m (f_i^{\max} - f_i^{\min})^2} \quad (10)$$

Where  $f_i^{\max}$  and  $f_i^{\min}$  represent the maximum respectively minimum value for the objective functions.

#### Computational experiments

We used a set of standard test problems from the project scheduling problem library PSPLIB from the University of Kiel [26]. We have generated cash flows of all activities from the interval [-1000; 1000] with the uniform distribution and the discount rate is assumed as  $\alpha=0.1$ . In the following examples MOPSO used a population of 100 particles, a repository size of 100 particles and 20 subdivisions of adaptive grid. All these values were determined after performing extensive set of experiments. The problems that we have selected from PSPLIB are in different categories of N (N=10,12,18,20,30),  $M_i$  ( $M_i=2,4,5$ ) and K ( $K=3,4,5$ ). In the following examples 220 instances have been selected and we report the average of performing 20 independent runs for each algorithm and each instance is replicated 10 times. We have coded the algorithm in MATLAB 7.9 and run on a Pentium 4 with 1 GB Ram. Tables 1, 2 and 3 show performance of the algorithm considering the metrics previously described. In Fig.1, Fig.2 and Fig.3, we can see the Pareto fronts obtained by the MOPSO in three typical examples with different sizes. In Fig.1 There are 27 non-dominated solutions produced by the algorithm in an example with small size. The spacing is 7.94 and the maximum spread is 563.94. We can see 16 non-dominated solutions for an example with medium size in Fig.2. The spacing is 12.45 and maximum spread of 381. Fig.3 represents 8 non-dominated solutions for a large size example while the spacing is 0.1 and the maximum spread is equal to 53.

TABLE I

RESULTS OF NO. OF NON-DOMINATED SOLUTIONS, SET NO (PSPLIB)

| Instance set | No. of Problems | No. of non-dominated solutions |
|--------------|-----------------|--------------------------------|
| J10          | 20              | 14                             |
| J12          | 20              | 16                             |
| J18          | 20              | 19                             |
| J20          | 20              | 24                             |
| J30          | 20              | 27                             |
| R3           | 20              | 22                             |
| R4           | 20              | 22                             |
| R5           | 20              | 22                             |
| M2           | 20              | 19                             |
| M4           | 20              | 24                             |
| M5           | 20              | 21                             |

TABLE II

RESULTS OF THE SPACING METRIC, SET NO (PSPLIB)

| Instance set | No. of Problems | Spacing |
|--------------|-----------------|---------|
| J10          | 20              | 9.87    |
| J12          | 20              | 10.7    |
| J18          | 20              | 10.95   |
| J20          | 20              | 8.07    |
| J30          | 20              | 14.24   |
| R3           | 20              | 11.07   |
| R4           | 20              | 10.66   |
| R5           | 20              | 13.73   |
| M2           | 20              | 11.49   |
| M4           | 20              | 11.95   |
| M5           | 20              | 7.72    |

TABLE III

RESULTS OF THE MAXIMUM SPREAD METRIC, SET NO (PSPLIB)

| Instance set | No. of Problems | Maximum spread |
|--------------|-----------------|----------------|
| J10          | 20              | 260            |
| J12          | 20              | 410            |
| J18          | 20              | 403.73         |
| J20          | 20              | 657.13         |
| J30          | 20              | 1029.5         |
| R3           | 20              | 552            |
| R4           | 20              | 429.8          |
| R5           | 20              | 413.44         |
| M2           | 20              | 443.63         |
| M4           | 20              | 551.6          |
| M5           | 20              | 470.27         |

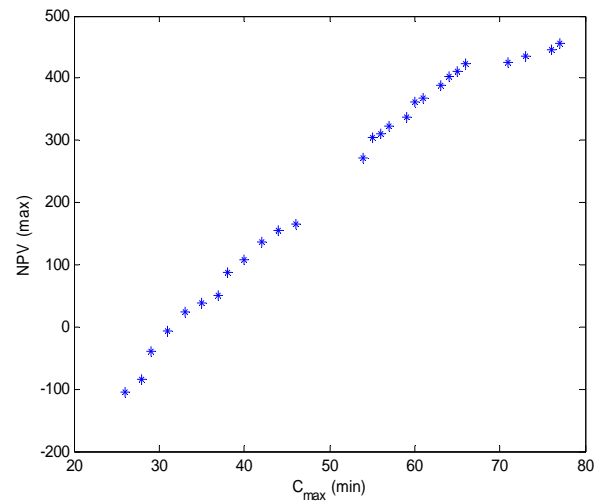


Fig. 1 Pareto front obtained by the MOPSO in an example with small size

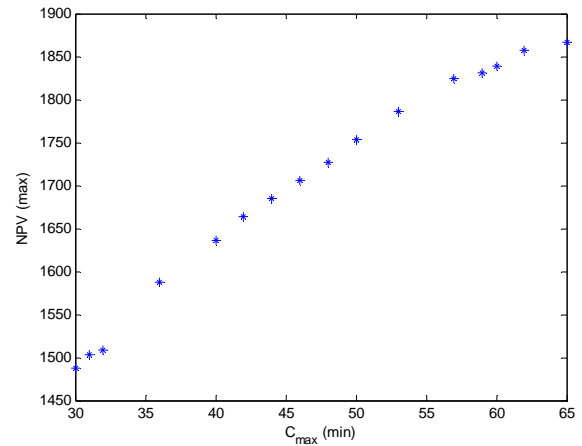


Fig. 2 Pareto front obtained by the MOPSO in an example with medium size

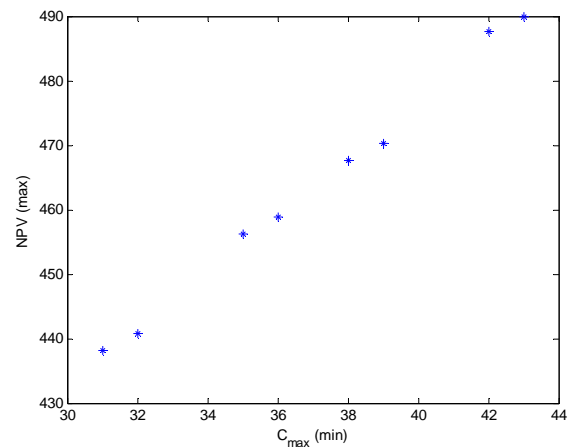


Fig. 3 Pareto front obtained by the MOPSO in an example with medium size

## VI.CONCLUSION

In this paper we consider multi-mode resource constraint project scheduling with the objectives of maximization the net present value and minimization of makespan. We used one evolutionary algorithm, MOPSO, for solving this problem. This metaheuristic implanted on 220 instances in variable categories obtained from PSPLIB where cash flows were generated randomly with the uniform distribution.

We have applied different metrics for comparing the non-dominated solutions and reported the results.

A further task on further research could be done in different scopes such as:

- Comparison the results of MOPSO with the other algorithms such as NSGA-II, PSA, MOTS, SPGA-II.
- Considering other objectives such as minimization of tardiness, maximizing of robustness and so on.
- Considering more constraints such as non-renewable resources or generalized precedence relations between activities.

## REFERENCES

- [1] J. Blazewicz, J.K. Lenstra and A.H.G RinnooyKan, "Scheduling Subject to resource constraints: classification and complexity", *Discrete Applied Mathematics*, vol. 5, pp. 11-24, 1983.
- [2] P. Bricker, A. Drexl, R. Mohring, K. Neumann and E. Pesch, "Resource-constrained project scheduling: Notation, classification, models and methods", *European Journal of Operational Research*, vol. 112, pp.3-41,1999.
- [3] S. Hartmann and D. Briskorn, "A Survey of variants and extensions of the resource-constrained project scheduling problem", *European Journal of Operational Research*, to be published.
- [4] J. Jozefowska, M. Mika, R. Rozycki, G. Waligora, and J. Weglarz, "Simulated annealing for multi-mode recourse-constrained project scheduling", *Annals of Operational Research*, vol. 102, pp.137-155, 2001.
- [5] K. Bouleimen and H. Lecocq, "A new efficient simulated annealing for the recourse-constrained project scheduling problem and its multiple mode version", *European Journal of Operational Research*, vol.149, pp.268-281, 2003.
- [6] J. Alcaraz, C. Maroto and R. Ruiz, "Solving the multi-mode recourse-constrained project scheduling problem with genetic algorithms". *Journal of the Operational Research Society*, vol. 54, pp.614-626, 2003.
- [7] S. Hartmann and A. Drexl, "Project scheduling with multi modes: A comparison of exact algorithms", *Network*, vol. 32, pp. 283-297, 1998.
- [8] M. Mori and CC. Tseng, "A genetic algorithm for multi-mode recourse-constrained project scheduling problem". *European Journal of Operational Research*, vol. 100, pp.134-141, 1997.
- [9] B. Jarboui, N. Damak, P. siarry and A. Rebai, "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problem", *Applied Mathematics and Computation*, vol. 115, pp.195-299, 2008.
- [10] R. Kolisch and S. Hartmann, "Experimental investigation of heuristics for recourse-constrained project scheduling: An update". *European Journal of Operational Research*, vol.174, pp.23-37, 2006.
- [11] M. Mika, G. Waligora and J. Weglarz, "Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models", *European Journal of Operational Research*, vol. 164, pp. 639-668, 2005.
- [12] G. Ulusoy, F. Sivrikaya and Ş. Şahin, "Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows". *Annals of Operations research*, vol. 102, pp. 237-261, 2001.
- [13] G. Waligora, "Discrete-continuous project scheduling with discounted Cash Flows-a tabu search approach". *Computers & Operations Research*, vol.35, pp. 2141- 2153,2008.
- [14] N. Nudtasomboon, S.U. Randhawa, "Resource-constrained project scheduling with renewable and non-renewable resources and time-resource tradeoffs", *Computers and Industrial Engineering*, vol.32, pp.227-242,1997.
- [15] S. Vob, A. Witt, "Hybrid flow shop scheduling as multi-mode multi-project scheduling problem with batching requirements: A real-world application", *International Journal of Production Economics*, vol.105,pp. 445-458, 2007.
- [16] R.Slowinski, B.Soniewicki, J.Weglarz,"DSS for multi-objective project scheduling", *European journal of operational research*, vol.79, pp. 220-229,1994.
- [17] J. Nabrzyunski, J. Weglarz, *Knowledge-based multi-objective project scheduling problems*, In: J. Weglarz, editor. Project scheduling: recent models, algorithms and applications. Kluwer academic Publishers, pp. 383-411,1999.
- [18] C.A. CoelloCoello, G.B. Lamont and D.A. Van Veldhuizen.and G.B. Lamont, *Evolutionary algorithms for solving multi-objective problems*. Norwell, MA: Kluwer, 2002.
- [19] A. Schirmer, "New insight on the complexity of resource-constrained Project scheduling-two cases of multi-mode scheduling", *University of Kiel*, Germany, 1996.
- [20] J. Kennedy and R. Eberhart, "Particle swarm optimization". In *IEEE International Conference on Neural Networks*, Pert, Australia, 1995, pp. 1942-1498.
- [21] C.A.CoelloCoello and M.S. Lechuga, "A proposal for Multi-objective particle swarm optimization". In *Congress on Evolutionary Computation*, Piscataway, New Jersey, 2002, pp.1051-1056
- [22] M. Reyes and C.A.CoelloCoello, "Multi-objective particle Swarm optimizers: A survey of the state-of-the-art". *International Journal of Computational Intelligence Research*, vol.2, pp. 287-308, 2006.
- [23] C.A. CoelloCoello, G.T.Pulido and M.S.Lechuga, "Handling multi-objectives with particle swarm optimization". *IEEE Transactions on Evolutionary Computation*, vol. 8, pp.256-279,2004.
- [24] J. R. Scott, "Fault tolerant design using single and multi-criteria genetic algorithm optimization". Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, USA, 1995.
- [25] E. Zitzler and L. Thiele, "Multi objective optimization using evolutionary algorithms, a comparative case study". In *Fifth International Conference on Parallel Problem Solving*, Berlin, Germany, 1998, pp. 292-301.
- [26] R. Kolisch and A. Sprecher, "PSPLIB- a project scheduling problem library", *European Journal of Operational Research*, vol.96, pp.205-216, 1996.
- [27] J. Weglarz, editor. *Project scheduling: recent models, algorithms and applications*. Kluwer academic Publishers, 1999.