World Applied Sciences Journal 4 (2): 270-276, 2008
ISSN 1818-4952

Multi Objective Particle Swarm Optimization for a Discrete Time, Cost and Quality Trade-off Problem

Maryam Rahimi and Hossein Iranmanesh

Department of Industrial Engineering,
Faculty of Engineering, University of Tehran, Tehran, Iran

Abstract: Time-Cost trade-off is a known and important problem in project management, because project managers should decide that whether they want to shorten the time of a project and tolerate its execution costs or minimize the costs and accept delays. It was recently suggested that the quality of a project should also be taken into considerations in this decision-making. In this paper, a meta-heuristic algorithm for the discrete time, cost and quality trade-off problem was applied and multiple alternative were considered for the activities of a project. Advantage of this method over others is that every activity has several different modes offered by various options of time, cost and quality and the best options of project's activities are determined in order to minimize the total cost of the project while maximizing the quality and also meeting a given deadline by assuming that duration and quality of project activities are discrete. Particle Swarm Optimization (PSO) algorithm was proposed to solve this problem. In this algorithm, initially a population of feasible solutions is generated. A number of these solutions are then selected and improved locally thorough the algorithm. The improved solutions are then combined to generate a new set of solutions. Since it was assumed that there were no bounds for three entities, a multi-objective problem has been considered. The whole process is stopped when no significant improvements, through fitness function achieved from multi-objective problem, can be made in the set of solutions that are observed. For validating of the efficiency of proposed multi-objective particle swarm, two types of problems, large size and small size are considered and for both types several problems are considered randomly and proposed algorithm is applied to them. Also a genetic algorithm is applied to those problems to comprise the efficiency of PSO and GA together.

Key words: Particle swarm optimization . Project management . Discrete time-cost and quality trade off problem . Best option . Genetic algorithm

INTRODUCTION

The discrete time, cost trade-off problem, hereafter referred to as DTCTP, is a well known problem in the project management literature. In most projects, the activity durations of some or all activities can be reduced by allocation of more resources. Of course, this causes an increase in the direct cost of these activities. The so called deadline problem aims at minimizing the project duration without exceeding a given budget. There is a tradeoff between cost and time, solutions with shorter duration usually cost more while solutions with low costs usually take longer [1, 2]. Recently, in expedition of a project, quality should also be taken into consideration along with the time and cost tradeoffs. Considering the inter twined effects of time, cost and quality in project management, it seems reasonable to develop a mathematical model, hereafter referred to DTCQTP, which considers project's time, cost and quality simultaneously. In DTCQTP, project's activities are performed in one of several alternatives. For each activity a set of time, cost and quality triplet, referred to as mode, are given.

The literature on DTCQTP is scant. At first Babu and Suresh suggested that the quality of a completed project may be affected by project crashing. They developed a solution procedure which considers the trade off among time, cost and quality in continuous mode [3]. This procedure later used to study an actual cement factory construction project in Thailand by Khang and Myint [4]. Tareghian and Taheri developed three inter related binary integer programming models for DTCQTP [5]. They also presented a new solution procedure based on scatter search that utilizes attraction of repulsion mechanisms borrowed from electromagnetism theory to move the sample point towards the optimality [6].

Corresponding Author: Dr. Hossein Iranmanesh, Department of Industrial Engineering, Faculty of Engineering, University of Tehran, P.O. Box 11115/4563, Tehran, Iran

In this paper, a multi objective method has been addressed in DTCQTP and with a meta-heuristic algorithm, Particle Swarm Optimization, for considering a large size problem solved, that assist project managers make tradeoff decision. In the following section, the problem is described formally. In section 3, a solution procedure based on PSO is described and in section 4, assumptions of Genetic algorithm is written for being comprised with PSO. In the section 5 computational results from proposed solution procedure when applied to two types of problems, large size and small size and for both types several problems that are considered randomly, are discussed. The final section presents the conclusion.

PROBLEM STATEMENT

One of the most important problems in construction project management is determining the best alternative of each activity, while minimizing total cost and time of project and maximizing the total quality of project. This problem could be considered in two categories:

- Decision maker or project manager determines limits or bounds for all time, cost and quality and then he is looking for the best alternatives for them, so in this way we are facing with a single objective problem.
- Decision maker or project manager couldn't determine a bound and in this way we are facing with a multi objective problem.

In this paper the second category has considered and it supposed that finding an upper or lower bound and also having any estimation for time, cost and quality is difficult.

In this problem, it is supposed that each activity has several alternatives to be done and each alternative has different time, cost and quality. For example, Table 1 shows a project with 3 activities that have 2 alternatives.

For each activity ij, a set M_{ij} of modes is given. For each mode, $r \in M_{ij}$, let t_{ijr}, c_{ijr} and q_{ijr} denote the duration, the cost and the quality attained by performing activity ij in mode r, respectively. The nature of each activity determines the quality level ($0 \leq q_{ijr} \leq 1$) assigned to it. For example number of errors or numbers of withdrawal outputs in doing the activity show the quality level. The objective is to construct the complete and efficient time, cost and quality profile to offer decision support in crashing a project. There would be a constraint to ensure that one and only one execution mode is assigned to each activity. This constraint can be shown as follows:

Table 1: Example of problem

	Alternative 1			Alternative 2		
Activity	Cost	Time	Quality	Cost	Time	Quality
12	2000	10	0.65	1800	12	0.70
23	1700	10	0.45	1500	15	0.50
24	1500	15	0.60	2100	10	0.55

$$\sum_{r \in M_{ij}} x_{ijr} = 1 \qquad ij = 1,2,...,n$$

The first objective considered is minimization of total project duration. This objective can be calculated by the following expression:

$$\text{Min} f_1 = \sum_{ij=1}^{n} \sum_{r \in M_{ij}} t_{ijr} x_{ijr}$$

When number of activity and also x_{ijr} is a zero or one variable presents that the mode of activity is chosen or not.

Another objective considered is minimization of total project cost. This objective can be calculated by the subsequent expression:

$$\text{Min } f_2 = \sum_{ij=1}^{n} \sum_{r \in M_{ij}} c_{ijr} x_{ijr}$$

Parameters of x_{ijr} and n are the same as explained in last objective.

The third and last objective considered is maximization of project's overall quality. To calculate the value of this objective, the following expression is used:

$$\text{Max } f_3 = \frac{1}{n} \sum_{ij=1}^{n} \sum_{r \in M_{ij}} q_{ijr} x_{ijr}$$

Parameters of x_{ijr} and n are the same as explained in first objective.

It can be easily noticed that the objectives considered are inherently contradicting. To illustrate the point, one should take into account that the optimization of the first objective in a single objective problem is performed regardless of project's cost and quality. Hence, the resulting sequences may have large cost or small quality, thus imposing large penalties to the system.

SOLUTION PROCEDURE

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by

Dr. Eberhart and Dr. Kennedy in 1995 [7], inspired by social behavior of bird flocking or fish schooling. According to what scientists have found, in order to search for food, each member in a flock of birds determines its velocity based on its personal experience as well as information gained through interaction with other members of the flock. This idea was the main principle for PSO. Each bird, called particle, flies through the solution space of the optimization problem searching for the optimum solution and thus its position represents a potential solution for the problem. In Particle Swarm terminology, the available solutions in each iteration are called the "swarm" which is equivalent to the "population" in genetic algorithms.

Solution representation: In order to construct a direct relationship between the problem domain and the PSO particles for this problem, every particle is in project and n numbers of dimensions of each particle are presented for n number of activities of a project. Each dimension represents a typical activity and the particle $X^t_i = [x_{i1}^t, x_{i2}^t, \ldots, x_{in}^t]$ corresponds to the continuous position values for n number of activities in this problem and also it does not present the number of modes. Instead, a heuristic like the SPV rule is used to determine the number of modes.

For each dimension, the distance between upper bound and lower bound is divided by number of modes, so we have several ranges and the continuous number that is calculated for each dimensions gives the position of range that it is in there, in this way we can convert the continuous position to a discrete one which is the key to enable the continuous PSO algorithm to be applied to this problem.

Ideal point: Ideal point is a virtual point with coordinates which are obtained by separately optimizing each objective function.

Since the problem in question is non-linear, even optimizing it and considering only one objective at a time is a demanding task. To overcome this obstacle, the problem in hand is first linearized so that each of the objective functions can be solved to optimality with available optimization software such as LINGO 8. In the process of finding the ideal point there is another problem, even after linearization, that is the NP-hardness of the large-size linearized problems due to their large feasible space and our inability to find the global optimum (even a strong local optimum) in a reasonable time. The following approach is adopted to solve this problem: when finding the exact ideal point is not easy, an approximation of it called the Dynamic Ideal Point (DIP) is used instead. The approximation requires interrupting the optimization software (LINGO 8) after ξ seconds after the first feasible solution is found and report the best solution found up to that time as the respective coordinate of the ideal point. The value of ξ is determined after running various test problems. To improve this approximation and to prevent it from reducing the quality of our algorithm, DIP must be updated at the end of each iteration of the proposed particle swarm.

Fitness function: In order to simultaneously maintain suitable intensification and diversification, a new function based on Goal Attainment method is considered. This Function can be shown as follows:

$$OF = \sum_{i=1}^{3} \frac{|f_i - F_i|}{F_i}$$

Where f_i is the i-th objective function value of the solution, F_i is the i-th coordinate value of the ideal point.

Initialization

- Set iteration counter as t = 0 and Population size as NP, twice the number of dimensions. A population of particles is constructed randomly for the PSO algorithm of this problem. The continuous values of positions are established randomly. The following formula is used to construct the initial continuous position values of the particle uniformly:

$$x^0_{ij} = x_{min} + (x_{max} - x_{min}) * r_1$$

  Where $x_{min} = 0.0$, $x_{max} = 4.0$ and r_1 is a uniform random number between 0 and 1. Generate NP particles randomly as explained above, $\{x^0_i, i=1, 2,\ldots, NP\}$ where $x^0_i = [x^0_{i1}, x^0_{i2},\ldots, x^0_{in}]$.
- Initial velocities are generated by a similar formula as follows:

$$v^0_{ij} = v_{min} + (v_{max} - v_{min}) * r_2$$

  Where $v_{min} = -4.0$, $v_{max} = 4.0$ and r_2 is a uniform random number between 0 and 1. Generate the initial velocities for each particle randomly, $\{V^0_i, i=1, 2,\ldots, NP\}$ where $V^0_i = [V^0_{i1}, V^0_{i2},\ldots, V^0_{in}]$
- Apply the heuristic rule to find the order of modes $Y^0_i = [Y^0_{i1}, Y^0_{i2},\ldots, Y^0_{in}]$ of particle X^0_i for i=1, 2,..., NP.
- From linear programming solution, Consider Dynamic Ideal Point (DIP) as

$$DIP^0 = [DIP^0_1, DIP^0_2,\ldots, DIP^0_n]$$

- Evaluate each particle in the swarm using the objective function OF^0_i for i = 1, 2,..., NP.
- For each particle in the swarm, set Personal best as $P^0_i = X^0_i$, where $P^0_i = [P^0_{i1} = X^0_{i1}, P^0_{i2} = X^0_{i2}..., P^0_{in} = X^0_{in}]$ together with its best fitness value, OF_i^{pb} for i = 1, 2,..., NP.
- Find the best fitness value among the whole swarm such that $OF_l = \min\{OF_i^0\}$ for i = 1,2,..., NP with its corresponding positions X^0_l. Set global best to $G^0 = X^0_l$ such that $G^0 = [g_1 = x_{l,1}, g_2 = x_{l,2},..., g_n = x_{l,n}]$ with its fitness value $Of^{gb} = OF_l$.

Updating the procedure

- Iteration counter: t = t + 1.
- Inertia weight: $w^t = w^{t-1} * \beta$ where β is decrement factor.
- Velocity: $v^t_{ij} = w^{t-1} v^{t-1}_{ij} + c_1 r_1(p_{ij}^{t-1} - x_{ij}^{t-1}) + c_2 r_2(g_j^{t-1} - x_{ij}^{t-1})$, Where c_1 and c_2 are acceleration coefficients and r_1 and r_2 are uniform random numbers between (0, 1).
- Position: $x^t_{ij} = x_{ij}^{t-1} + v_{ij}^t$.
- Order of modes: Apply the heuristic rule to find the permutation $Y^0_i = [Y^0_{i1}, Y^0_{i2},..., Y^0_{in}]$ for i = 1, 2,..., NP.
- Dynamic ideal point: At the end of each iteration of the proposed particle swarm, if the minimum values of each objective function for all the particles of the swarm is smaller than its related coordinate of the dynamic ideal point; this coordinate is replaced by that value. According to this procedure, the value of the dynamic ideal point will progressively be improved during the optimization process.
- Personal best:
  Each particle is evaluated by using the permutation to see if the personal best will improve. That is, if $OF^t_i < OF_i^{pb}$ for i = 1, 2,..., NP then personal best is updated as $P^t_i = X^t_i$ and $OF_i^{pb} = OF_i^t$.
- Global best:
  Find the minimum value of personal best. That is, $OF_l^t = \min\{OF_i^{pb}\}$, i= 1, 2,..., NP.
  If $OF_l^t < OF^{gb}$, then the global best is updated as $G^t = X_l^t$ and $OF^{gb} = OF_l^t$

Stopping criterion: If the number of iterations exceeds the maximum number of iterations, or maximum CPU time, then stop. To construct the set of N initial solutions, those N solutions are selected that have the shortest distance to the dynamic ideal point.

Regarding the PSO parameters, the acceleration coefficients were taken as $c_1 = c_2 = 2$ consistent with the literature. Initial inertia weight was set to $w_0 = 0.9$ and never decreased below 0.40. Finally, the decrement factor ß was taken as 0.975. The population size was twice the number of dimensions [8, 9].

GENETIC ALGORITHM FOR DTCQTP

In the last decade, genetic algorithms have been successfully applied to optimize problems in diverse fields. Genetic algorithms differ from other search techniques which rely on analogies to natural genetic and biological evolution processes, start with an initial set of random solutions called "population" and use a process similar to biological evolution to improve upon them. Each individual in the population is a string of symbols, usually but not necessarily, a binary bit string, called a "chromosome" or "individual", representing a solution to the optimization problem. Chromosomes are made of units, "genes"[10].

To make a relationship between this problem and genetic algorithm, n numbers of genes are presented for n number of activities of project. Each gene represents a activity and the chromosome shows a project. Other considerations in genetic algorithm for this problem are following:

- Population size: population size is considered as twice the number of units.
- Crossover operator: several crossover operators have been proposed for permutation presentation, such as partially mapped crossover (PMX), order crossover (OX) and cycle cross-over (CX). Order crossover (OX) as crossover operator by probability of 0.8 is considered.
- Mutation operator: The mutation operator is designed with a neighbor search technique in order to find an improved offspring. Mutation operator of inversion by probability of 0.05 is considered.
- Elitist selection: Randomly removing one chromosome and adding the best one into the new generation if it is not selected through the process of roulette wheel selection. Elite probability of 0.1 is considered in this study.
- Fitness function of minimization of scheduling function is considered.
- Termination criteria: 100 number of generation

COMPUTATIONAL EXPERIMENTS

The number of execution modes per activity is randomly chosen from $|M_{ij}| \varepsilon [2, 10]$. For each activity, the execution mode is generated as follows: first the number of modes for activity ij is generated, i.e. $|M_{ij}| \sim (2, 10)$. Then the duration of each mode is randomly sampled from (10, 100). The corresponding cost for

Table 2: Problem set s for the small sized problems

Problem	Activity	Mode	OF_{MOPS}	OF_{MOGA}
1	3	2	0.05	0.05
2	4	2	0.11	0.13
3	5	2	0.07	0.12
4	4	3	0.09	0.08
5	5	3	0.11	0.15
6	6	3	0.11	0.17
7	7	3	0.11	0.15
8	5	4	0.14	0.13
9	6	4	0.12	0.15
10	7	4	0.14	0.23

each mode is sampled randomly from (50, 80). It is assumed that under normal circumstances, the quality attained by each activity is 99%. To generate quality for other modes, we have assumed that the activity duration compressions do not adversely affect the activity qualities by the same magnitude. We have therefore, classified activities into five different categories. The quality for each mode is sampled randomly from (0.50, 0.99).

The proposed solution procedure is tested with a number of instances of the problem for which the optimal solution by LINGO could obtain. This algorithm has been coded in Visual Basic 6.

Small sized problems: The first experiment is carried out on a set of the small sized problems. This experiment contains 10 test problems of different sizes from PSO and GA depicted in Table 2.

Large sized problems: Another set of experiment is implemented for the large sized problems. To consider desired test problem, 14 test problems of different sizes are generated and with the results from PSO and GA are illustrated in Table 3.

Solution distribution according to frequency for 12*10 problems is represented in Fig. 1 and 2. As it can be seen all of solutions of PSO are less than 0.5 but

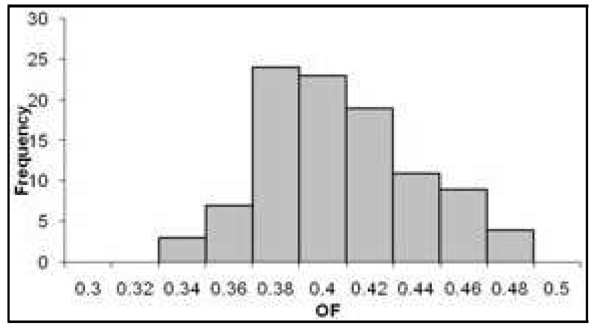

Fig. 1: Solutions distribution for PSO

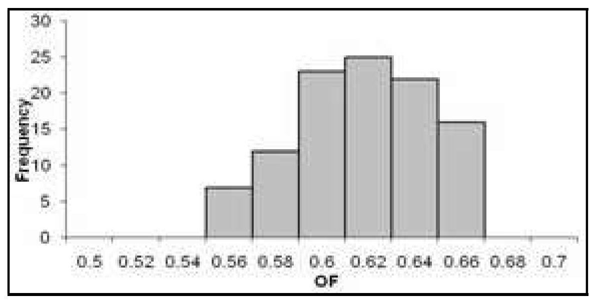

Fig. 2: Solutions distribution for GA

Table 3: Problem sets for the large sized problems with best solution

			MOPS				MOGA			
Problem	Activity	Mode	Time	Cost	Quality	OF	Time	Cost	Quality	OF
1	6	5	280	403	0.84	0.32	314	395	0.86	0.40
2	7	5	293	511	0.92	0.20	303	430	0.93	0.24
3	8	5	310	632	0.82	0.22	287	707	0.79	0.29
4	7	6	290	567	0.87	0.27	274	693	0.87	0.30
5	8	6	340	668	0.90	0.31	385	640	0.91	0.37
6	9	6	407	731	0.91	0.19	399	753	0.91	0.20
7	9	7	386	745	0.85	0.18	306	805	0.84	0.20
8	10	7	504	904	0.81	0.25	545	880	0.81	0.27
9	9	8	492	883	0.94	0.22	580	850	0.95	0.30
10	10	8	481	954	0.85	0.27	434	1126	0.82	0.41
11	10	9	583	1003	0.90	0.26	540	1405	0.88	0.35
12	11	9	622	1138	0.79	0.19	654	1086	0.80	0.25
13	11	10	654	1250	0.75	0.19	668	1024	0.75	0.32
14	12	10	713	1436	0.72	0.34	672	1745	0.71	0.56

Table 4: Computational results of MOPS for large size problems

			Time				Cost				Quality			
Problem	Activity	Mode	Best	Max	Average	stdev	Best	Max	Average	stdev	Best	Min	Average	stdev
1	6	5	280	402	304	14.3	403	718	584	20.4	0.84	0.74	0.78	0.04
2	7	5	293	414	314	14.9	511	626	570	21.5	0.92	0.79	0.83	0.04
3	8	5	310	428	352	15.2	632	744	705	23.6	0.82	0.66	0.72	0.05
4	7	6	290	395	314	15.1	567	782	624	25.3	0.87	0.68	0.78	0.05
5	8	6	340	474	402	17.3	668	797	745	27.4	0.90	0.61	0.71	0.05
6	9	6	407	510	489	19.1	731	847	796	30.3	0.91	0.68	0.74	0.05
7	9	7	386	498	403	20.8	745	920	843	32.5	0.85	0.59	0.68	0.05
8	10	7	504	618	593	22.5	904	1286	992	35.2	0.81	0.63	0.72	0.04
9	9	8	492	607	567	23.2	883	1120	979	35.9	0.94	0.71	0.82	0.06
10	10	8	481	622	563	22.7	954	1732	1126	38.3	0.85	0.67	0.73	0.06
11	10	9	583	713	677	25.5	1003	2039	1640	40.1	0.90	0.64	0.77	0.06
12	11	9	622	795	731	25.9	1138	1982	1783	39.4	0.79	0.58	0.69	0.05
13	11	10	654	808	746	26.4	1250	1790	1537	40.5	0.75	0.61	0.69	0.08
14	12	10	713	911	817	31.3	1436	2045	1803	42.8	0.72	0.59	0.64	0.10

the solutions is greater than 0.5. This concludes the performance of PSO for this problem is significantly better than GA.

Best solution, maximum (for minimizing objective) or minimum (for maximizing objective), average and standard deviation of results from PSO are represented in Table 4 as well. These tables show that the performance of PSO rather than GA in the same condition is better and PSO for DTCQTP is more efficient.

CONCLUSION

In this paper a multi-objective particle swarm (MOPS) for determining the best alternatives of a project's activities in discrete time, cost and quality trade-off problem with the objective of the minimizing of the total time and total cost and also maximizing the total quality of the project, was studied. A similar concept for Ideal Point in multi-objective optimization problems (Dynamic Ideal Point) was introduced and used in the initialization phase and in the main algorithm. In the initialization phase, the DIP was approximated using linear programming when finding the exact Ideal Point was difficult and this approximation was improved with regard to the better values found for each of the objective functions throughout the main algorithm.

The efficiency of proposed multi-objective particle swarm was validated by considering two types of problems, large size and small size and applying particle swarm optimization algorithm and genetic algorithm to several problems of each type. The experimental results indicated that the proposed MOPS was able to improve the quality of the obtained solutions in the same condition, especially for the large-sized problems and in five cases in the small sized problems, all of available non-dominated solutions of the searching space obtained by enumeration was detected by MOPS.

REFERENCES

1. De, P., E.J. Dunne, J.B. Gosh and C.E. Wells, 1995. The discrete time-cost tradeoff problem revisited. European Journal of Operational Research, 81: 225-238.
2. Deineko, V.G. and G.J. Woeginger, 2001. Hardness of approximation of the discrete time-cost tradeoff problem. Operations Research Letters, 29: 207-210.
3. Babu, A.J.G. and N. Suresh, 1996. Project management with time, cost and quality considerations. European Journal of Operational research, 88: 320-327.
4. Khang, D.B. and Y.M. Myint, 1999. Time, cost, quality trade-off in project management: A case study. International Journal of project Management, 17 (4): 249-256.
5. Tareghian, H.R. and S.H. Taheri, 2006. On discrete time, cost and quality trade-off problem. Applied Mathematics and Computation, 181: 1305-1312.
6. Tareghian, H.R. and S.H. Taheri, 2007. A solution procedure for the discrete time, cost and quality trade-off problem using electromagnetic scatter search. Applied Mathematics and Computation, 190: 1136-1145.

7. Eberhart, R.C. and J. Kennedy, 1995. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science. Nagoya, Japan, pp: 39-43.
8. Tasgetiren, M.F., M. Sevkli, Y.C. Liang and G. Gencyilmaz, 2007. Particle swarm optimization algorithm for makespan and total flowtime minimization in permutation flowshop sequencing problem. European Journal of Operational research, 177: 1930-1947.
9. Rahimi-Vahed, A.R. and S.M. Mirghorbani, 2007. A multi-objective particle swarm for a flowshop scheduling problem. Journal of Combination Optimization, 13: 79-102.
10. Cheng, R. and M. Gen, 1998. Loop layout design problem in flexible manufacturing systems using genetic algorithms. Computers and Industrial Engineering, 34: 53-61.