Expert Systems with Applications 39 (2012) 2794–2803

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Extraction of decision alternatives in construction management projects: Application and adaptation of NSGA-II and MOPSO

Elahe Fallah-Mehdipour [a,*], Omid Bozorg Haddad [a], Mahmoud M. Rezapour Tabari [b], Miguel A. Mariño [c]

[a] Department of Irrigation & Reclamation Engineering, Faculty of Agricultural Engineering & Technology, College of Agriculture & Natural Resources, University of Tehran, Karaj, Tehran, Iran

[b] Department of Engineering, Shahrekord University, Shahrekord, Iran

[c] Department of Land, Air & Water Resources, Department of Civil & Environmental Engineering, and Department of Biological & Agricultural Engineering, University of California, 139 Veihmeyer Hall, University of California, Davis, CA 95616-8628, USA

ARTICLE INFO

Keywords:
Project management
Time-cost-quality trade-off
NSGA-II
MOPSO

ABSTRACT

The time-cost trade-off problem is a known bi-objective problem in the field of project management. Recently, a new parameter, the quality of the project has been added to previously considered time and cost parameters. The main specification of the time-cost trade-off problem is discretization of the decision space to limited and accountable decision variables. In this situation the efficiency of the traditional methods decrease and applying of the evolutionary algorithms is necessary. In this paper, two evolutionary algorithms that originally search the decision space in a continuous manner including: (1) multi-objective particle swarm optimization (MOPSO) and (2) nondominated sorting genetic algorithm (NSGA)-II, are considered as the optimization tools to solve two construction project management problems. These problems are both in discrete domain including two or tree objectives, separately. In this regard, some procedures has been suggested and then applied to adopt both algorithms capable in solving the problems in a discrete domain. Results show the advantages and effectiveness of the used procedures in reporting the optimal Pareto for the optimization problems. Moreover, the NSGA-II is more successful in determining optimal alternatives in both time-cost trade-off (TCTO) and time-cost-quality trade-off (TCQTO) problems than the MOPSO algorithm.

1. Introduction

Simulation and optimization models can provide solutions to many problems in the field of operations research. Although simulation models are based on trial and test methods followed by engineering judgment, final solutions may not be optimal. In contrast, optimization models can yield optimal/near-optimal solutions by searching a part or an entire decision space. Different single-objective optimization methods such as linear (LP), nonlinear (NLP) and dynamic programming (DP) are capable to move toward optimal solutions. However, difficulties in determining optimal solutions, especially in some discrete or nonlinear problems, as well as the curse of dimensionality in solving the large-scale problems, are disadvantages of those optimization methods.

Evolutionary algorithms are potential candidates to determine optimal/near-optimal solutions in the aforementioned problems. In these types of algorithms, random decision variables are produced as input data for a simulation model. Output data from the simulation model can be used as input data for an optimization model. In such a process, newly-generated decision variables, based on previously calculated ones, can be improved. This process continues up to the maximum number of iterations for determining the best solution. Genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), and simulated annealing (SA) are evolutionary algorithms that have been developed for solving optimization problems.

Multi-objective problems are another type of operations research problems which include a vector of objectives instead of a single objective. The main goal of multiobjective optimization techniques is to determine a set of optimal solutions, especially when objectives are conflicting. In traditional optimization methods, techniques such as the weighting approach are used in LP and NLP to produce just a single optimal point based on the considered weights. However, evolutionary algorithms can yield a set of nondominated solutions, Pareto, as the optimal solutions. Nondominated sorting genetic algorithm (NSGA), multi-colony ACO (MOACO), and multi-objective PSO (MOPSO) are some examples of multi-objective evolutionary optimization algorithms of this type.

The time-cost trade-off (TCTO) problem is a traditional bi-objective problem with a discretized decision space. Two

* Corresponding author.
E-mail addresses: Falah@ut.ac.ir (E. Fallah-Mehdipour), OBHaddad@ut.ac.ir (O.B. Haddad), mrtabari@eng.sku.ac.ir (M.M. Rezapour Tabari), MAMarino@ucdavis.edu (M.A. Mariño).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.08.139

existing objectives are minimization of: (1) the total cost and (2) the total time of project execution with respect to activity orders and relations and other management constraints. Thus there are different alternatives, from the best value of each objective to the worst one, satisfying the constraints and relations.

Solutions to the TCTO problem, especially using evolutionary algorithms as the optimization tool, have been reported by various investigators. Feng, Liu, and Burns (1997) categorized existing techniques for solving the TCTO problem in two mathematical and heuristic (evolutionary algorithms) methods. Li and Love (1997) used an improved GA to facilitate solving the TCTO problem. Hegazy (1999) used the GA to analyze the optimal time and cost of construction. Zheng, Thomas, and Kumaraswamy (2004) proposed adaptive weights derived from a previous generation for producing a search pressure toward the ideal point. Tareghian and Taheri (2006) developed a three interrelated binary integer programming technique for the TCTO problem. Rasmy, Abdelsalam, and Hussein (2008) applied a Pareto SA (PSA) algorithm to determine a set of nondominated solutions for the TCTO problem in the critical chain project.

Recently, maximization of equivalent quality of the project has been added to the time and cost as the third objective. Afshar, Kaveh, and Shoghli (2007) developed a new evolutionary algorithm, multi-colony ACO, for the time-cost-quality trade-off (TCQTO) problem. Rahimi and Iranmanesh (2008) solved the TCQTO problem using the MOPSO algorithm. They proposed a single-objective function based on the goal attainment method:

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

in which: OF = fitness function; f_i = ith objective function; and F_i = ith coordinate value of the ideal point. In fact, the TCQTO problem was transferred to a single-objective one by using Eq. (1) and the Pareto obtained by considering different combinations of weights in each run of the PSO algorithm. This process seems to be time-consuming because the PSO algorithm has to be used for different combinations of weights.

In this paper, two evolutionary algorithms, a multi-objective version of the PSO algorithm (MOPSO) and NSGA-II, are used to solve the TCTO and TCQTO problems. Although in both algorithms the search space is evaluated by a continuous search method, in this paper both algorithms are adapted to search in a discrete domain. Results are then compared to select the best algorithm.

2. Multi-objective optimization problems

Many optimization problems can be presented by the following general mathematical model:

$$Max./Min. \quad f(x) \quad x \in X \tag{2}$$

$$f : X \to R \tag{3}$$

in which: $f(x)$ = objective function; x = decision variables vector; X = decision variables space; and R = set of real numbers.

Fig. 1(a) shows a schematic of transferring data from a two-dimensional decision variable space to a decision space in single-objective problems. In multi-objective optimization problems, optimizing a vector of objectives is considered instead of satisfaction of a single objective. Generally, in mathematical form, multi-objective optimization problems can be defined as:

$$Max./Min. \quad f(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \quad x \in X \tag{4}$$

$$f : X \to R^m, \ m \in \mathrm{N} \tag{5}$$

where m = number of objectives and N = set of natural numbers.

There are techniques such as the weighting method and ε-constraint method which transfer multi-objective problems to a single-objective one using different combinations of a weighting vector and constraints. Thus, each optimal solution can be assigned to a specific combination of weighting vector and constraint. Hence, in each run of the algorithm, a single point (solution) can be achieved. However, multi-objective evolutionary algorithms are capable to find almost all candidate solutions (Pareto) in a single run.

Fig. 1(b) presents a schematic of transferring data from two-dimensional decision variable space to the decision space in a two-objective problem. As it is shown, each set of decision variables has been related to a couple of objectives. Note that none of the solutions dominate the others. In other words, if all objective values of a solution dominate the correspondent values of another solution, the former will be a dominated solution and the latter will be removed. Otherwise, both solutions will be located in the nondominated set.

Fig. 2 shows dominated and nondominated relations between objective values in a bi-objective problem in which both objectives are minimization. In this figure, solutions labeled by 1 or 2 have nondominated conditions individually. Note that the set labeled 1 dominates the set labeled 2. In the optimization procedure, the best set of nondominated solutions is called Pareto-front. Thus, there are two Paretos in Fig. 2, in which the one labeled 1 is the Pareto-front.

3. Multi-objective evolutionary algorithms

Evolutionary algorithms are a subset of evolutionary computations which can perform optimal/near-optimal solutions in all types of problems (linear/nonlinear, discrete/continuous, convex/

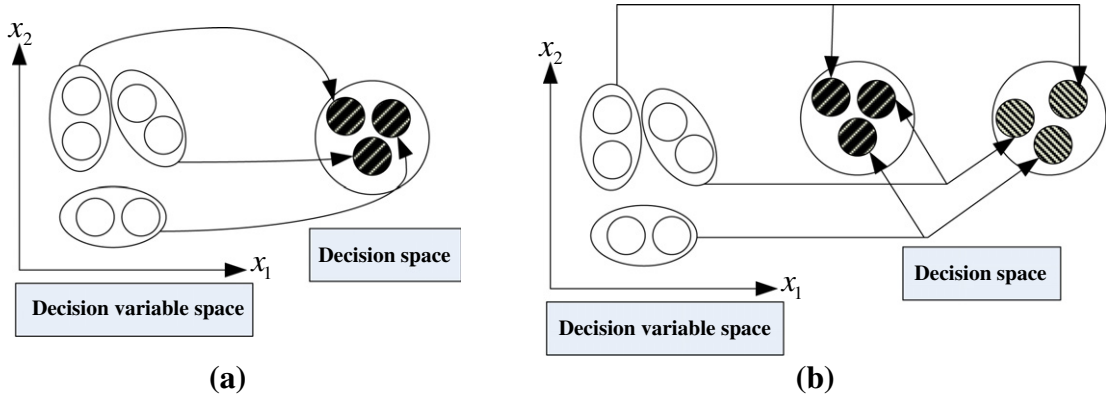

Fig. 1. Schematic of decision and decision variable spaces in: (a) single-objective and (b) multi-objective problems.

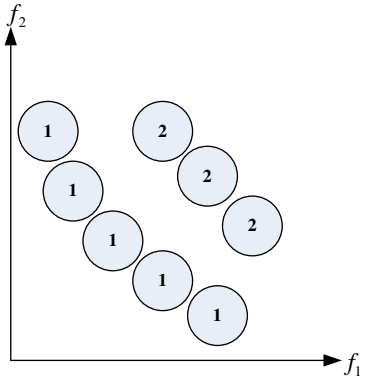

Fig. 2. Schematic of dominated and nondominated conditions of solutions in a bi-objective problem.

nonconvex) using validated experimental theories of biological evolution and natural processes, particularly through activities of different species of animals.

A set of solutions resulting from a program run, without using any techniques such as the weighting approach that are directly related to decision-makers' opinions, is the most important advantage of evolutionary algorithms in the field of multi-objective optimization. Thus, these algorithms have been used as optimization tools in recent multi-objective optimization problems (Deb, 2001; Tan, Lee, & Khor, 2001). In this paper, two different types of multi-objective evolutionary algorithms, the MOPSO algorithm and NSGA-II are used as optimization tools in extraction solution of the TCTO and TCQTO problems.

3.1. Multi-objective particle swarm optimization (MOPSO) algorithm

The PSO algorithm is an optimization technique based on a bird migration pattern. In the real world, the movement of birds towards food can display a regular system in which each bird improves its position in the time dimension. In the PSO algorithm, each bird can be presented as a particle (single solution) and a set of the birds is identified as a swarm (set of solutions). Thus, in an optimization problem, the position of the ith particle (x_i) can be represented by a D-dimensional vector:

$$x_i = [x_{i1}, x_{i2}, \ldots, x_{iD}] \quad \text{for} \quad i = 1, \ldots, N \tag{6}$$

where D = number of decision variables and N = swarm size. Moreover, the best bird (with the smallest distance from the food) is called the global best (*Gbest*), and the best position of a bird ever tried toward the food is the particle best (*Pbest*). Steps of the PSO algorithm are as follows:

In the first step, random solutions (x_i) with a normal distribution of decision variables are produced. In the second step, those solutions are used in the simulation model as input data. The objective function for each particle (solution) is then calculated and stored in the memory of the algorithm. For the next step, *Pbest* and *Gbest* are assigned with respect to the best position of particles and swarm so far discovered, respectively. In the fourth step, the velocity is calculated as:

$$v_{id}^{t+1} = \alpha\left(w^t.v_{id}^t + c_1.r_1^t\left(Pbest_{id}^t - x_{id}^t\right) + c_2.r_2^t\left(Gbest_d^t - x_{id}^t\right)\right)$$
$$\text{for} \quad i = 1, 2, \ldots, N \; d = 1, 2, \ldots, D \tag{7}$$

$$w^t = w_{Max} - \frac{(w_{Max} - w_{Min}) \times t}{iter_{Max}} \tag{8}$$

in which: v_{id}^{t+1} = velocity of ith particle for dth dimension in $(t+1)$th iteration; α = constriction factor which is a fixed pre-specified value and controls the velocity of particles in the decision variables space; w^t = inertia weight parameter in tth iteration which is calculated by Eq. (8). This parameter starts from the maximum value (w_{Max}) in the first iteration to the minimum value in the last iteration ($iter_{Max}$). That is, at the beginning of search process, the effect of velocity (v_{id}^t) is more than that in later iterations. c_1 = cognitive parameter; c_2 = social parameter (c_1 and c_2 assign the proportion of *Pbest* and *Gbest* in the velocity); and r_1^t and r_2^t = uniformly distributed random numbers in [0,1] in the tth iteration. Hence, each particle moves in the decision space by a velocity vector with two elements. Thus, $Pbest_{id}^t$ = best position of the ith particle for the dth dimension in the tth iteration and $Gbest_d^t$ = best position of the swarm for the dth dimension in the tth iteration.

At the next step, resulting velocities are controlled using lower ($v_{\min}$) and upper ($v_{\max}$) bounds of velocity:

$$v_{\min} \leqslant v_{id}^{t+1} \leqslant v_{\max} \tag{9}$$

Finally, the particle position is calculated by using Eq. (10):

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{10}$$

The resulting particle position is used as the new input for the simulation model in the second step and new objective functions are again calculated. This process continues up to the maximum number of iterations. The aforementioned PSO algorithm has a continuous search mechanism to move toward an optimal point.

The movement toward an optimal point in the single-objective PSO (SOPSO) algorithm is different from that in the MOPSO algorithm. In the SOPSO algorithm, each particle follows its objective

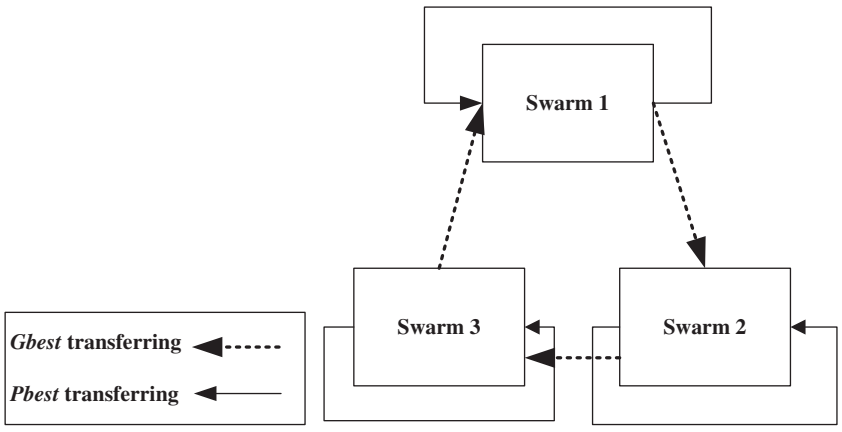

Fig. 3. Schematic of the *Pbest* and *Gbest* transferring in a three-objective problem.

function in the search process. In the MOPSO algorithm, however, the number of objective functions is more than one. Thus, the movement in the decision space toward each objective and consequently the search process should be adopted.

The vector evaluated PSO (VEPSO) is a multi-objective algorithm that uses one swarm for each objective. Thus, each swarm uses its particle position as the *Pbest*, but the *Gbest* of each swarm is replaced by the *Gbest* of other swarms for the next iteration. Fig. 3 shows an example of the transferring *Pbest* and *Gbest* in a three-objective problem.

In this paper, the proposed VEPSO algorithm (Parsopoulos & Vrahatis, 2002) is coupled with a dynamic external archive for transferring the produced particles in each iteration. Thus, the particles are compared and nondominated ones are stored and others are removed at the end of each iteration. In this mechanism, the member of an external archive can be reported in each iteration and the external archive size changes dynamically.

3.2. Nondominated sorting genetic algorithm (NSGA)-II

GA is a type of evolutionary algorithm in which a population set of chromosomes (solutions) moves toward better solutions. The evolution usually starts from a random population in the first generation (iteration). In each generation, the fitness of all chromosomes in the population is evaluated. Chromosomes are then stochastically selected from the current population (based on their fitness), and modified using the GA operators (crossover and mutation) to perform a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm process stops in the maximum number of generations. By increasing the objectives in different optimization problems, applications of the multi-objective GA (MOGA) have been recently developed from concepts borrowed from the single-objective GA (SOGA).

Srinivas and Deb (1994) used the nondominated sorting concept on the GA. Steps of the NSGA-II are as follows:

In the first step, a set of random solutions (chromosomes) with a uniform distribution are produced. The first generation is a $N \times D$ dimensions matrix, in which N and D are identified as the number of chromosomes (solutions) and decision variables (genes), respectively. There are dominated and nondominated solutions in the population that create different Paretos. In the second step, chromosomes are classified to the aforementioned Paretos using Eq. (11):

$$d_{I_j} = \sum_{m=1}^{M} \frac{f_m^{I_{j+1}} - f_m^{I_{j-1}}}{f_m^{Max} - f_m^{Min}} \tag{11}$$

where: d_{I_j} = crowded distance of jth solution; M = number of objectives; $f_m^{I_{j+1}}$ and $f_m^{I_{j-1}}$ = values of mth objective for $(j-1)$th and $(j+1)$th solution; f_m^{Max} = maximum value of mth objective function among solutions of the current population; and f_m^{Min} = minimum value of mth objective function among solutions of the current population.

In the aforementioned equation, index I_j denotes the jth solution in the sorted list and $(j-1)$ and $(j+1)$ are the two nearest neighboring solutions on both sides of I_j. The algorithm then searches mostly near points (solutions) with more value of d_{I_j}. This process will cause the more uniform distribution of the resulting Pareto and a vast range of selections for the decision makers. The Pareto is then ranked from the best to the worst solutions, in which the comparison criterion is the distance from ideal Pareto-front with the assumed location. In the third step, the selection operator which is the crowded tournament is considered. It is used to select appropriate chromosomes (which are called parents) from the previous generation. The crowded tournament operator compares different solutions using two criteria: (1) a nondominated rank and (2) a crowding distance in the population. In this process, if a solution dominates the others, it will be selected as the parent. Otherwise, the solution with the higher value of crowding distance will be selected as the winner.

Deb and Agrawal (1995) developed a simulated binary recombination (crossover) operator, called the SBX operator, to combine two chromosomes and creating two new chromosomes (children). This operator is similar to one-point cut crossover in the binary data. The probability distribution is calculated:

$$P(\beta_i) = \begin{cases} 0.5(\eta_c+1)\beta_i^{\eta_c}, & \text{if } \beta_i \leqslant 1 \\ 0.5(\eta_c+1)\frac{1}{\beta_i^{\eta_c+2}}, & \text{otherwise} \end{cases} \tag{12}$$

$$\beta_i = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}}, & \text{if } u_i \leqslant 0.5 \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \tag{13}$$

in which: $P(\beta_i)$ = crossover probability; β_i = difference between the objective functions of parents and children; η_c = a constant which shows the difference between the objective functions of parents and children (a large value of η_c gives a higher probability for creating near-parent solutions); and u_i = a random value in [0,1]. The aforementioned difference between parents and children is

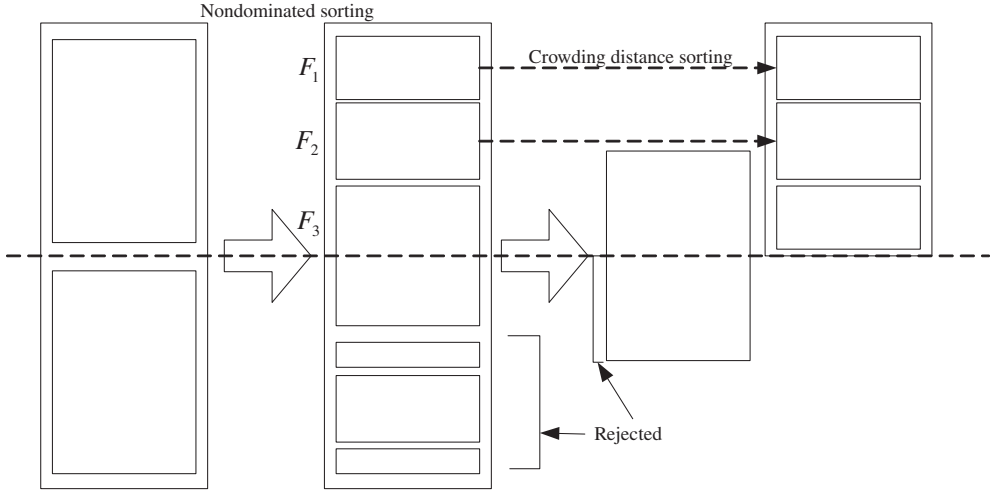

Fig. 4. Schematic of the NSGA-II procedure (Deb, 2001).

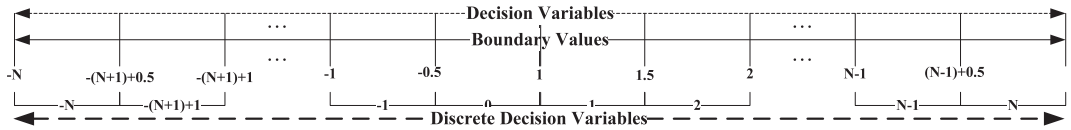

Fig. 5. Schematic of discretization in the adaptive MOPSO.

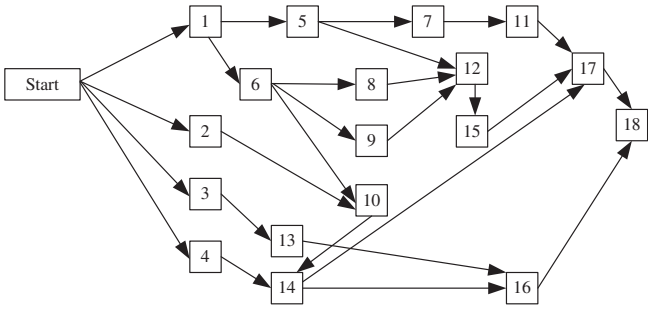

Fig. 6. The first test problem network.

calculated with Eq. (14) and the children's values are produced by Eq. (15):

$$\beta_i = \left| \frac{x_1^{child} - x_2^{child}}{x_1^{parent} - x_2^{parent}} \right| \tag{14}$$

$$\begin{aligned} x_1^{child} &= 0.5\left[(1+\beta_i)x_1^{parent} + (1-\beta_i)x_2^{parent}\right] \\ x_2^{child} &= 0.5\left[(1-\beta_i)x_1^{parent} + (1+\beta_i)x_2^{parent}\right] \end{aligned} \tag{15}$$

where x_1^{child}, x_2^{child} = value of the first and second child's chromosomes and $x_1^{parent}, x_2^{parent}$ = value of the first and second parent chromosomes, respectively.

The other GA operator is mutation. A polynomial mutation operator which is proposed by Deb and Goyal (1996) has been used in this paper:

$$\delta_i = \left\{ \begin{array}{ll} (2r_i)^{1/(\eta_m+1)} - 1 & , \quad if\ r_i < 0.5 \\ 1 - [2(1-r_i)^{1/(\eta_m+1)}] & , \quad if\ r_i \geqslant 0.5 \end{array} \right\} \tag{16}$$

in which δ_i = mutation value; r_i = a random value in [0,1]; and η_m = distribution constant of mutation. The δ parameter is added to the parent gene value, as follows:

$$x^{child} = x^{parent} + \delta \tag{17}$$

A new generation which is a combination of the parents' and children's chromosomes is then produced. In the new generation, different chromosomes are ranked and chromosomes of the first rank are selected for the next generation. If the number of these chromosomes is less than the population size, chromosomes with a lower rank will be added to fulfill the new generation. Fig. 4 shows the NSGA-II procedure.

4. Time-cost-quality trade-off problems

There are three main issues that are most important in a successful project: (1) it has to be on time (time); (2) it has to be within budget (cost); and (3) a project must meet the customer's requirement (quality). Time-cost-quality trade-off is a known and

Table 1
Specification for the first test problem.

Activity	Option	Duration (day)	Cost (10^3 $)	Activity	Option	Duration (day)	Cost (10^3 $)
1	1	14	2.40	9	2	18	0.24
	2	15	2.15		3	20	0.18
	3	16	1.90		4	23	0.15
	4	21	1.50		5	25	0.10
	5	24	1.20	10	1	15	0.45
2	1	15	3.00		2	22	0.35
	2	18	2.40		3	33	0.32
	3	20	1.80	11	1	12	0.45
	4	23	1.50		2	16	0.35
	5	25	1.00		3	20	0.30
3	1	15	4.50	12	1	22	2.00
	2	22	4.00		2	24	1.75
	3	33	3.20		3	28	1.50
4	1	12	45.00		4	30	1.00
	2	16	35.00	13	1	14	4.00
	3	20	30.00		2	18	3.20
5	1	22	20.00		3	24	1.80
	2	24	17.50	14	1	9	3.00
	3	28	15.00		2	15	2.40
	4	30	10.00		3	18	2.20
6	1	14	40.00	15	1	16	3.50
	2	18	32.00	16	1	20	3.00
	3	24	18.00		2	22	2.00
7	1	9	30.00		3	24	1.75
	2	15	24.00		4	28	1.50
	3	18	22.00		5	30	1.00
8	1	14	0.22	17	1	14	4.00
	2	15	0.22		2	18	3.20
	3	16	0.20		3	24	1.80
	4	21	0.21	18	1	9	3.00
	5	24	0.12		2	15	2.40
9	1	15	0.30		3	18	2.20

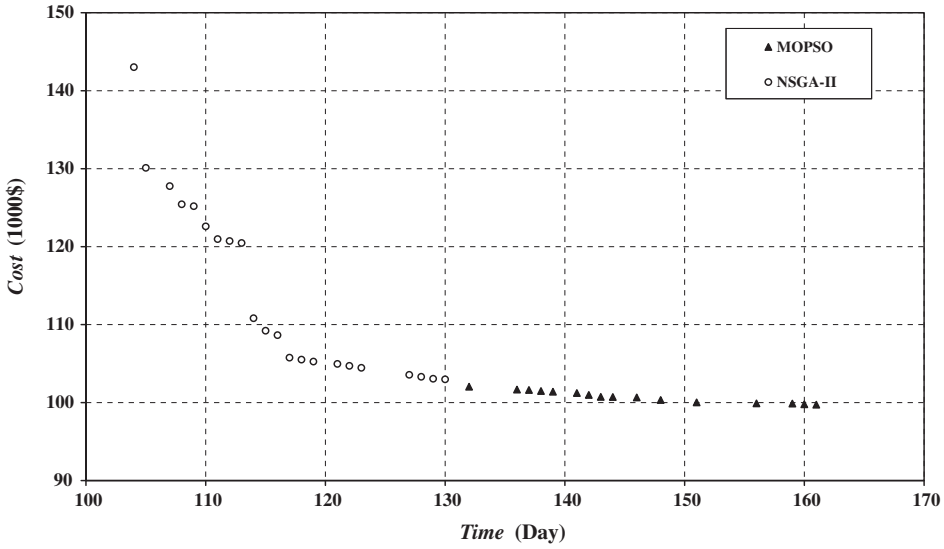

Fig. 7. Resulted Pareto-fronts using the MOPSO algorithm and NSGA-II for the TCTO problem.

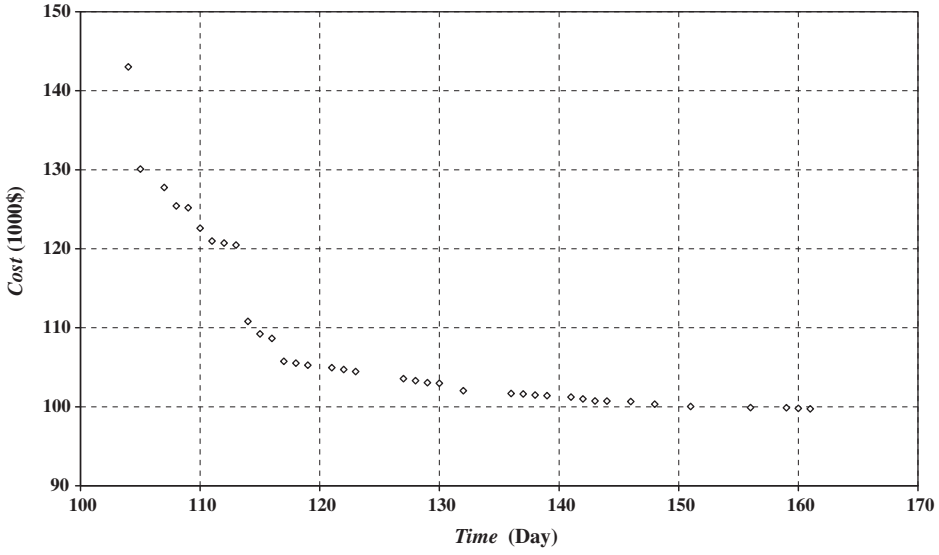

Fig. 8. Final Pareto using Merged of the MOPSO algorithm and NSGA-II for the TCTO problem.

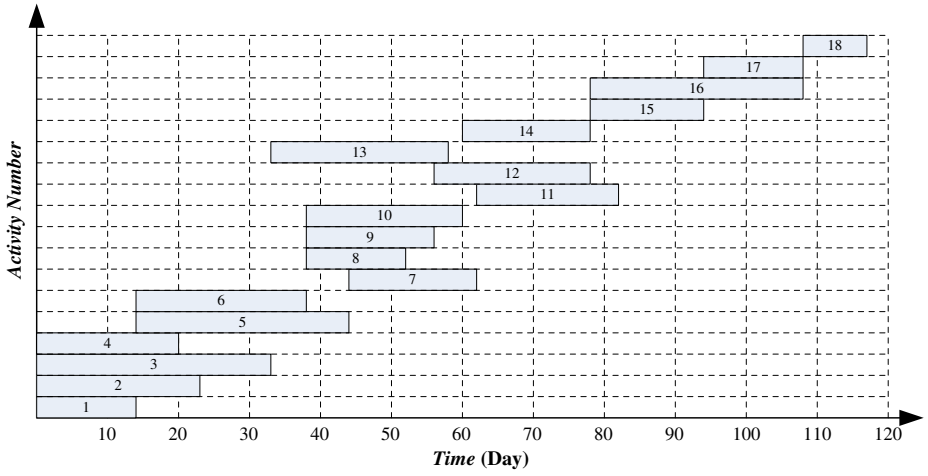

Fig. 9. Duration of different activities for a selected solution in the first test problem.

important problem whose main characteristic is decision-making in a discrete decision space. In this problem, there are three objectives: (1) minimizing the total time of the project; (2) minimizing the total cost of the project; and (3) maximizing the total quality of the project with respect to the order of the project's activities, as follows:

$$Time = Min.\left(Max\left[\sum_{i\in A} t_i^{(k)} a_i^{(k)}\right]\right) \tag{18}$$

$$Cost = Min.\left(\sum_{i\in A} c_i^{(k)} a_i^{(k)}\right) \tag{19}$$

$$Quality = Max.\left(\sum_{i\in A} w_i^{(k)} a_i^{(k)}\right) \tag{20}$$

in which: *Time* = total time of project; A = set of activity; $t_i^{(k)}$ = duration of ith activity for kth option; *Cost* = total cost of project; $a_i^{(k)}$ = index of ith activity for kth option; $c_i^{(k)}$ = cost of ith activity for kth option; *Quality* = total quality of project; and $w_i^{(k)}$ = quality weight of ith activity for kth option.

The TCQTO problem is a three-objective problem in which the value of each objective decreases with an increase in other objectives. Thus, there is no single solution as the optimal point and a set of nondominated solutions/Pareto-front should be identified as the optimal set. Thus, evolutionary algorithms are suitable candidates to produce a set of solutions.

5. Adaptation of the MOPSO algorithm and NSGA-II in TCQTO problems

The TCTO and TCQTO are two- and three-objective problems with a discrete decision space. Any solution in these problems is related to a vector of time, cost, and quality. Thus, to find optimal solutions, evolutionary algorithms have been adapted to search in a discrete decision space.

In this paper, the MOPSO algorithm and NSGA-II have been used as the optimization tools. Both algorithms search the decision space based on a continuous method. To apply theses algorithms in the TCTO and TCQTO problems, a different technique has been proposed for each algorithm in which a continuous searching method is transferred to a discrete one.

5.1. The discrete MOPSO algorithm

In the MOPSO algorithm, to determine optimal solutions, a type of classification has been used which divides a feasible continuous decision space into discrete classes. In this method, integer decision variables are randomly produced in the first iteration. The velocity of each particle is then calculated with Eq. (7). Generally, this parameter may have different positive and negative continuous values. However, those values are considered to be classified to use as integer variables in the TCTO and TCQTO problems. Fig. 5 shows schematic of the discretization of the velocity parameter. As shown, continuous values are related to integer values. Thus, velocity is presented as an integer parameter which is added to the decision variable (x) and produces a new integer decision variable for the next iteration. The new variable is related to the corresponding vectors (time, cost, and quality) to calculate objective functions.

5.2. The discrete NSGA-II

As presented in the NSGA-II identification, this algorithm produces continuous variables using crossover and mutation operators. To adapt the NSGA-II for solving the TCTO and TCQTO problems, another type of classification has been used. In this process, all decision variables are limited in [0,1]. These variables are multiplied by 1000 and their integer part is separated and saved as the value of each variable. Each variable is then divided by the maximum states of related activity and the residual indicates the state of activity. Eqs. (21) and (22) present an example for an activity with three different states:

$$\mathrm{int}(x \times 1000) = x' \rightarrow R = 3^*((x'/3) - \mathrm{int}(x'/3)) \tag{21}$$

$$x'' = \begin{cases} 1 & R = 0 \\ 2 & R = 1 \\ 3 & R = 2 \end{cases} \tag{22}$$

where: x' = integer part of $(x \times 1000)$; R = residual of x' divided by 3; and x'' = new integer value of decision variable (x). According to this process, all values in [0,1] are transferred to an integer value.

6. Case study

In this paper, to apply the MOPSO algorithm and NSGA-II in TCTO and TCQTO problems, two different test problems are considered with two and three objectives, respectively. It should be noted that the time objective of this problem has been calculated considering the critical path method (CPM). It is concerned with two objectives: (1) planning the project schedule in a way that it is completed as quickly as possible and (2) identifying those activities where a delay in their execution is likely to affect the

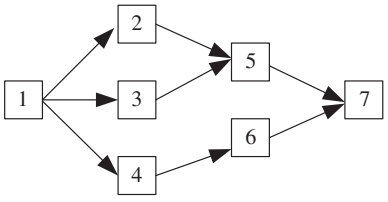

Fig. 10. The second test problem network.

Table 2
Specification for the second test problem.

Activity	Options	Duration (days)	Cost (10^3 dollars)	Activity weight (percent)	Quality (percent)
1	1	14	23	8	98
	2	20	18		89
	3	24	12		84
2	1	15	3	6	99
	2	18	2.4		95
	3	20	1.8		85
	4	30	1.2		70
	5	60	0.6		59
3	1	15	4.6	14	98
	2	22	4		81
	3	33	3.2		63
4	1	12	45	19	94
	2	16	35		76
	3	20	30		64
5	1	22	20	17	99
	2	24	17.5		89
	3	28	15		72
	4	30	10		61
6	1	14	40	19	100
	2	18	32		79
	3	24	18		68
7	1	9	30	17	93
	2	15	24		71
	3	14	22		67

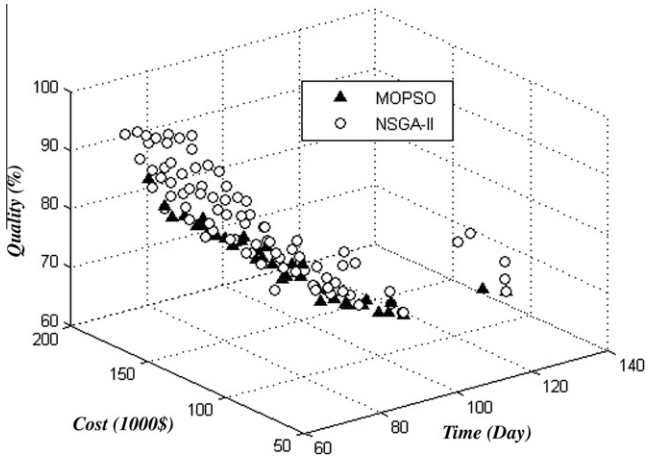

Fig. 11. Resulted Pareto-fronts using the MOPSO algorithm and NSGA-II for the TCQTO problem.

overall end date of the project or later activities start dates. The CPM is used for projects that are made up of a number of individual activities. An activity is a specific task with time duration and a precedence relationship with other activities of the project. If some of the activities require other activities to finish before they can start, then the project becomes a complex set of activities (Bozorg Haddad, Mirmomeni, Zaredeh Mehrizi, & Mariño, 2008).

Moreover, the same function evaluation has been considered for both algorithms. The number of particles and chromosomes are equal to 100 and the generation of NSGA-II is 500. However, two swarms have been used in the MOPSO algorithm and the maximum iteration is equal to 250.

To select an appropriate Pareto-front, a comparison criterion which is called generational distance (*GD*) has been used (Veldhuizen, 1999). This criterion is capable to present an average of distances from an optimal or ideal Pareto-front, as follows:

$$GD = \frac{1}{NS}\sqrt{\sum_{i=1}^{NS} d_i^2} \tag{23}$$

where: *GD* = generational distance; *NS* = number of solutions found; and d_i = euclidean distance (in the objective space) between members of the calculated Pareto and the nearest member of the ideal Pareto-front.

6.1. First test problem

To evaluate the MOPSO algorithm and NSGA-II in a TCTO problem, a test problem including 18 activities which has been initially presented by Feng et al. (1997) has been considered. Fig. 6 shows the network of this problem which includes activities arrangement. Table 1 presents these activities along with different options.

Resulting Pareto using the MOPSO algorithm and NSGA-II are presented in Fig. 7. As shown, both Paretos are approximately in the same location.

Because there is no known Pareto-front for the TCTO problem considered in this paper, an ideal point with the minimum value of each objective has been assumed to be the optimal Pareto-front. In fact, the ideal point is at the intersection of solutions with minimum values of time and cost. Because there are different units for time and cost, values of objective functions have been normalized to values in [0,1]. Numbers 0 and 1 have been assigned to solutions with the minimum value of each objective (best one) and the maximum value of each objective (worst one), respectively. In the current problem, possible minimum values of time and cost are 104 (day) and 99,740($), respectively, which have been replaced by the normalized value of zero for both time and cost.

The *GD* value for the resulting Pareto using the MOPSO algorithm and NSGA-II are 0.466 and 0.375, respectively. Thus, the NSGA-II Pareto is the more appropriate Pareto in the first test problem.

As shown in Fig. 7, minimum values for the time objective for the MOPSO algorithm and NSGA-II are different. Those minimum time values for the NSGA-II and MOPSO algorithm are 104 and 110 (day), respectively.

For comparison purposes, the Pareto of the MOPSO algorithm and NSGA-II have been merged and nondominated solutions have been presented as the final Pareto (Fig. 8). The proportions of nondominated solutions in the final Pareto yielded by each algorithm were then calculated. Results showed that 57.89(%) of nondominated solutions were determined by the NSGA-II. Thus, the

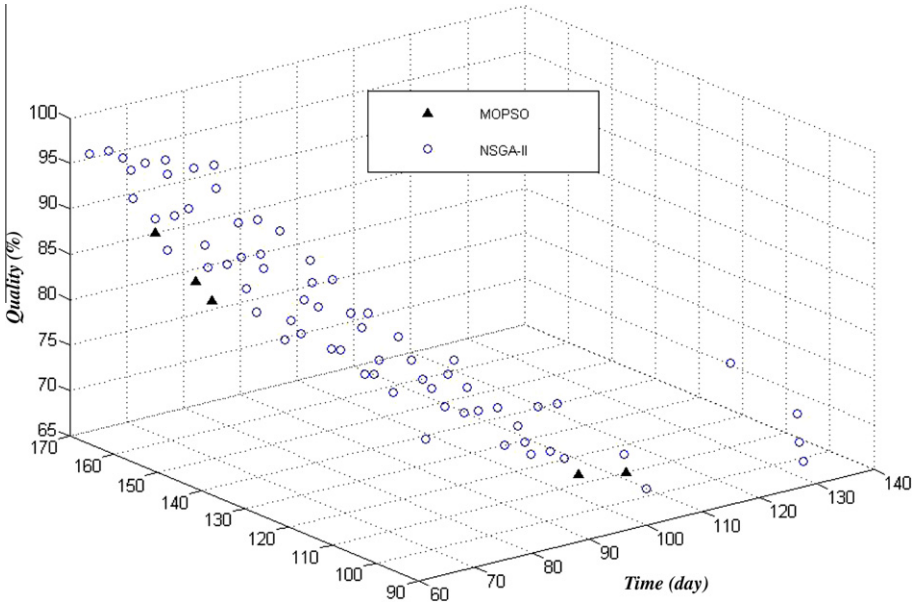

Fig. 12. Final Pareto using Merged of the MOPSO algorithm and NSGA-II for the TCQ problem.

NSGA-II is more capable to determine appropriate solutions than the MOPSO algorithm.

To illustrate the efficiency of the CPM in time managing, the duration of each activity for a solution that has the lowest distance from ideal point (0,0) is shown in Fig. 9. The decision variables of the aforementioned solution are {1,4,3,3,4,3,3,1,2,2,3,1,3,3,1,5,1,1} corresponding to 117 (day) and 105,760 ($). According to related options for each activity, the time of project has been calculated to be 360 days without considering the CPM. Thus, using the CPM improves the total time of the project.

6.2. Second test problem

To apply the MOPSO algorithm and NSGA-II in a three-objective construction problem, the seven-activity TCQTO problem of Zheng and Thomas (2005) is considered. Fig. 10 shows the network of the TCQTO problem. Moreover, different options of each activity are presented in Table 2. According to this table, minimum and maximum options for decision making are 2 and 5, respectively. The best value for the single-objective problem is 60 (day), 95,800 ($), and 97 (%) for the first (*Time*), second (*Cost*) and third (*Quality*) objectives. Thus, the comparison point for calculating the *GD* criterion is equal to (0,0,1) after transferring solutions to the corresponding values between 0 and 1.

Fig. 11 shows the resulting Paretos using the MOPSO algorithm and NSGA-II. To compare these Paretos, the *GD* criterion is equal to 0.792 and 0.768 for the MOPSO algorithm and NSGA-II Pareto-fronts, respectively. Both algorithms achieved the optimal single-objective solution. The merged Pareto, which is called final Pareto, is shown in Fig. 12. According to the final Pareto, 93.24 (%) of the solutions were obtained by the NSGA-II and the rest by the MOPSO procedure. Thus, the NSGA-II is more capable to find nondominated solutions than the MOPSO procedure.

To show the difference between optimization of two and three objectives in the TCQTO problem, different combinations of two objectives were considered. Resulting Paretos using the MOPSO

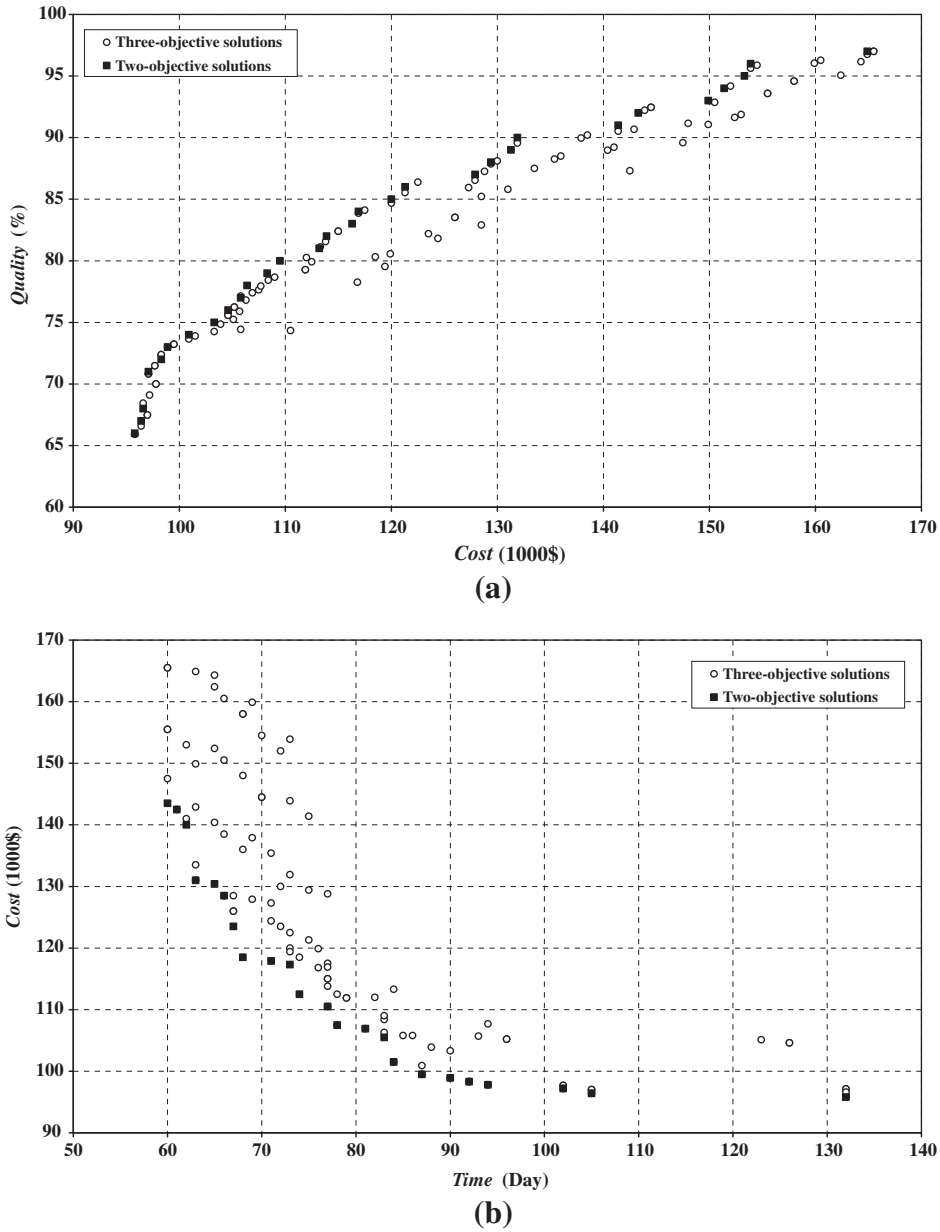

Fig. 13. Solutions for the two and three objective problem for: (a) cost-quality and (b) time-cost.

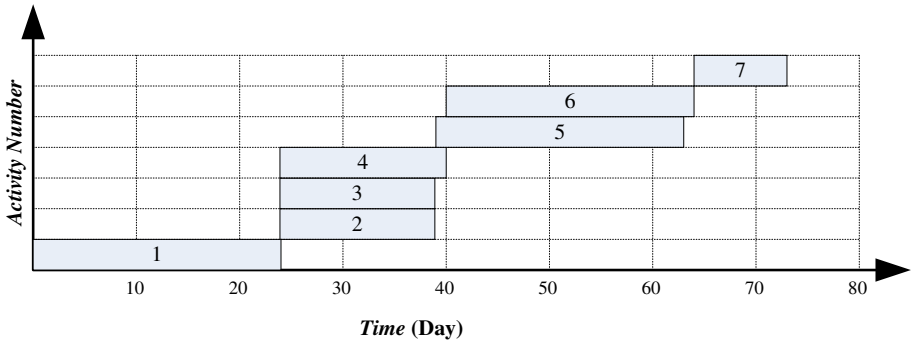

Fig. 14. Duration of different activities for a selected solution in the second test problem.

algorithm with two objectives in the TCQTO problem are shown in Fig. 13. As shown, the number of nondominated solutions in the three-objective problem exceeds that of the bi-objective problem. However, most of the three-objective solutions are dominated solutions. Thus, an increase in the number of objectives adds the number of nondominated solutions to the nondominated solutions found in problems with fewer objectives. Note that there is no conflict between time and quality objectives in this problem. Thus, both objectives achieved the optimal solution using the same alternatives and options ({1,1,1,1,1,1,1}) as decision variables. Thus, there is no Pareto-front for the time and quality optimization problem.

To demonstrate the capability of using the CPM in the TCQTO problem, the duration time of a nondominated solution with the lowest distance from the normalized ideal point (0,0,1) is presented in Fig. 14. The total time of the project for the aforementioned solution ({3,1,1,2,2,3,1}) is 73 (day). The time duration was calculated to be 125 (day) without using the CPM. The cost and quality of this solution are 120,000 ($) and 85 (%), respectively.

7. Concluding remarks

Recently, project management problems have been considered to be multi-objective problems including time, cost, and the quality as objectives. In multi-objective problems, different objectives are commonly conflicting. In time-cost-quality trade-off (TCQTO) problems, the quality of the project has been considered as an objective which is improved by increasing the cost of the project. On the other hand, decreasing the time of the project increases the cost of the project. Thus, time and quality are conflicting with the cost of the project. However, the time and quality of a project usually changes in the same direction. Thus, a project with a less duration time achieves an upper level of quality. Evolutionary algorithms are suitable tools that can yield different solutions by considering existing conflicts.

The other specification of a TCQTO problem is having a discrete decision-variable space. Thus, the used algorithms should be adapted for using in the TCTO and TCQTO problems. In this paper, the MOPSO algorithm and NSGA-II were used to solve TCTO and TCQTO problems. These algorithms search the decision space in a continuous manner. To adapt the MOPSO algorithm and NSGA-II with the TCTO and TCQTO problems for discrete decision-making, two techniques were implemented. In the former method, the continuous decision rounds to the near integer value while the latter method uses a relative integer value. Moreover, to compare the Paretos and select an appropriate one, a comparison criterion called generational distance (*GD*) has been used.

Results showed that both NSGA-II and MOPSO algorithms can yield an acceptable number of nondominated solutions. The NSGA-II with a smaller value of *GD* in both TCTO and TCQTO problems has been nominated. The value of *GD* using the NSGA-II was 27.58 which has been improved (decreased) by 3.03% compared to that of the MOPSO algorithm. To confirm previous results, the resulting Paretos using the aforementioned algorithms were merged and nondominated solutions were considered to be the final Pareto. The proportion of nondominated solutions in the final Pareto was used as the other criterion for selecting an appropriate algorithm. Results showed that the capability of the NSGA-II is superior to that of the MOPSO procedure in determining the final Pareto for TCTO and TCQTO problems (57.89 and 93.24%, respectively).

References

Afshar, A., Kaveh, A., & Shoghli, O. R. (2007). Multi-objective optimization of time-cost-quality using multi-colony ant algorithm. *Asian Journal of Civil Engineering (Building and Housing), 8*(2), 113–124.

Bozorg Haddad, O., Mirmomeni, M., Zaredeh Mehrizi, M., & Mariño, M. A. (2008). Finding the shortest path with honey-bee mating optimization algorithm in project management problems with constrained/unconstrained resources. *Computational Optimization and Application*. doi:10.1007/s10589-008-9210-9.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. England: John Wiley and Sons, Ltd.

Deb, K., & Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems, 9*(2), 115–148.

Deb, K., & Goyal, M. (1996). A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics, 26*(4), 30–45.

Feng, C., Liu, L., & Burns, S. S. (1997). Using genetic algorithms to solver construction time cost trade off problem. *Journal of Computing in Civil Engineering, 11*(3), 184–189.

Hegazy, T. (1999). Optimization of construction time-cost trade-off analysis using genetic algorithms. *Canada Journal of Civil Engineering (NRC Canada), 26*, 685–697.

Li, H., & Love, P. E. D. (1997). Using improved genetic algorithms to facilitate time-cost optimization. *Journal of Construction Engineering and Management – ASCE, 123*(3), 233–237.

Parsopoulos, K. E., & Vrahatis, M. N. (2002). Particle swarm optimization method in multiobjective problems. In *Proceedings of the 2002 ACM symposium on applied computing, March 2002* (pp. 603–607). Madraid, Spain.

Rahimi, M., & Iranmanesh, H. (2008). Multiobjective particle swarm optimization for a discrete time, cost and quality trade-off problem. *World Applied Sciences Journal, 4*(2), 270–276.

Rasmy, M. H., Abdelsalam, H. M. E., & Hussein, R. R. (2008). Multi-objective time-cost trade-off analysis in critical chain project networks using Pareto simulated annealing. In *Proceeding of the sixth international conference on informatics and systems, March 2008 (INFOS2008)*. Egypt, Cairo.

Srinivas, N., & Deb, K. (1994). Multi-objective function optimization using nondominated soting genetic algorithms. *Evolutionary Computation Journal, 2*(3), 221–248.

Tan, K. C., Lee, T. H., & Khor, E. F. (2001). Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. In *Proceeding of IEEE congress on the evolutionary computation* (pp. 979–986). Korea, Seoul.

Tareghian, H. M., & Taheri, S. H. (2006). On discrete time cost quality trade off problem. *Applied Mathematics and Computation, 181*(2), 1305–1312.

Veldhuizen, D. (1999). Multiobjective evolutionary algorithms: Classification, analyses, and new innovations. Ph.D. Thesis. Dayton, Ohio: Department of Electrical and Computer Engineering Air Force Institute of Technology.

Zheng, Daisy, X. M., Thomas, N. G. S., & Kumaraswamy, Mohan, M. (2004). Applying a genetic algorithm-based multiobjective approach for time-cost optimization. *Journal of Construction Engineering and Management – ASCE,* 130(2), 168–176.

Zheng, Daisy, X. M., & Thomas, N. G. S. (2005). Stochastic time-cost optimization model incorporating fuzzy sets theory and nonreplaceable front. *Journal of Construction Engineering and Management – ASCE*, 131(2), 176–186.