Solving a generalised precedence multi-objective multi-mode time-cost-quality trade-off project scheduling problem using a modified NSGA-II algorithm

Maghsoud Amiri and Amir-Reza Abtahi*

Department of Industrial Management,
Faculty of Management and Accounting,
Allameh Tabataba’i University,
Postal Box 6479-14155, Haft-Peikar Alley,
Nezami Ganjavi Ave., Tavanir St., Tehran, Iran
Fax: +98-021-44453640
E-mail: mg_amiri@yahoo.com
E-mail: amir_abtahi@yahoo.com
*Corresponding author

Kaveh Khalili-Damghani

Department of Industrial Engineering,
South Tehran Branch,
Islamic Azad University,
No. 148, 134 Ave., 111 St., Tehran-Pars, Tehran, Iran
E-mail: kaveh.khalili@gmail.com

Abstract: In this paper, a new mathematical formulation is proposed to model a generalised precedence multi-objective multi-mode time-cost-quality trade-off project scheduling problem (GPDTTCQTP). Afterwards, a modified NSGA-II algorithm is developed to solve the proposed GPDTTCQTP. The modified NSGA-II utilises a dynamic parameter tuning and a heuristic self-adaptive constraint handling strategy. These properties result in proper performance in regenerating the Pareto front of the GPDTTCQTP. Investigating the efficiency of proposed algorithm several benchmark instances are systematically generated and solved. The proposed procedure is straightforward and results are promising.

Keywords: multi-objective genetic algorithm; NSGA-II; project scheduling; time-cost-quality trade-off; generalised precedence relations.


Biographical notes: Maghsoud Amiri is with Department of Industrial Management, Faculty of Management and Accounting, Allameh Tabataba’i University, Tehran, Iran since 2004. He is currently an Associate Professor of Industrial Management at ATU. He has several publications in international

Amir-Reza Abtahi is with Department of Industrial Management, Faculty of Management and Accounting, Allameh Tabataba'i University, Tehran, Iran since 2008. He has some publications in international conferences and journals such as International Journal of Advanced Operations Management, International Journal of Modeling in Operation Management, International Journal of Strategic Decision Sciences, International Journal of Applied Decision Sciences, Management Studies Journal, International Conference of Management, and Knowledge of Management Journal. His research interests are meta-heuristic methods, DEA, operations management, project management, supply chain modelling, artificial neural network, and system dynamics.


1 Introduction

Managing projects successfully is both an art and a science and attempts to control corporate resources within the constraints of time, cost, and quality (Kerzner, 2009). Time, cost, and quality which can be defined as the competing objectives of a project have to balance through project scheduling which is an integral phase of project management (Iranmanesh et al., 2008). In many projects there is a need to compress the overall project completion time. To this aim, project manager has to allocate more resources and therefore much money to the project which leads to increasing the total direct cost of the project. Instead, it is obvious that applying less expensive resources would lead to longer time to complete each project’s activity and then longer overall project completion time. For example, using more productive resources such as using more efficient equipment or hiring more workers, etc., may compress the duration of completing an activity, but may increase the direct cost of the project (Zhang and Xing, 2010). On the other hand, compressing project time and decreasing its cost
Solving a generalised precedence multi-objective

may affect the quality of the project and decrease its level (Babu and Suresh, 1996; Kim et al., 2012).

So, considering the aforementioned trade-off between time, cost, and quality in scheduling a project is a serious problem for a project manager. This problem can be studied in continuous and discrete variants. In continuous variant of the problem, it has been assumed that there are general continuous functions that relate time, cost, and quality to each other. In discrete variant of time-cost-quality trade-off problem (TCQTP), it has been considered that the relationship between time, cost, and quality of a project are defined at discrete points (Demeulemeester and Herroelen, 2002). Practical and theoretical importance of this problem persuaded researchers to develop different approaches for problems in this category (Weglarz et al., 2011; Warren Liao et al., 2011).

Though each activity can be executed in one of its possible modes while each mode has its own time, cost and quality amounts. Hence, In order to balance the objectives time, cost and quality for improving the efficiency of a project, some methodologies have been proposed to solve TCQTP by deciding an optimal combination of execution modes for all project activities.

In this paper, a new variant of generalised precedence relations discrete time-cost-quality trade-off problem (GPDTCQTP) represents. Due to our best knowledge, there exists no similar research work in literature of trade-off problems which contributed to generate non-dominated solutions in presence of no articulation of preference relations between conflictive objectives. The activities of the GPDTCQTP are assumed to have several executing modes with specified time-cost-quality in each mode. The PGPDTCQTP serves several real and practical issues as follows. The GPDTCQTP have three conflictive objectives and generalised precedencies are considered between activities. The proper upper and lower bounds are assumed for time, cost, and quality of the project. All these assumptions are practical and are to be faced in real projects. Due to our best knowledge, there is no unique trade-off model which considered all aforementioned properties, concurrently.

A new mixed-integer mathematical formulation is proposed for GPDTCQTP. As it is hard to represent the preference articulation of decision makers (DMs) in form of relative importance of objective functions so, it is interesting to generate set of non-dominated solutions for GPDTCQTP. Hence, a dynamic self-adaptive version of NSGA-II algorithm is proposed to generate set of non-dominated solutions on Pareto front of instances of GPDTCQTP.

The next sections of this paper are arranged as follows. In Section 2, a review of literature of TCQTP, concepts of multiple-objective optimisation and genetic algorithm are represented. In Section 3 the problem formulation and notations have been proposed. In Section 4 proposed modified NSGA-II has been introduced. In Section 5 the experimental results are represented. Finally, conclusion remarks are summarised in Section 6.

2 Literature review

In this section, a literature review of TCQTP, multiple-objective decision making concepts, and genetic algorithms (GAs) are represented.
2.1 Literature of TCQTP

The interactions between conflictive objectives of a project such as time, cost, and quality is called trade-off problem. Several continuous and discrete variants of trade-off problems, has widely been studied (Demeulemeester and Herroelen, 2002). In continuous trade-off problem, there are functions which correlate time, cost, and quality objectives. In discrete variant, the relationships between objectives of a project are defined at discrete points (Demeulemeester and Herroelen, 2002). In this case, each activity can be executed in several modes. The time, cost and quality of each mode is predetermined. Hence, the feasible solution space of the problem extensively increases for medium and large size instances. Trade-off problem is known to be NP-Hard (De et al., 1997).

### Table 1 Summary of studies on the TCQT problem

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Method</th>
<th>Main contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Babu and Suresh (1996)</td>
<td>Linear programming</td>
<td>Using three inter-related linear programming model simply extendable to non linear models</td>
</tr>
<tr>
<td>Khang and Myint (1999)</td>
<td>Linear programming</td>
<td>Applying Babu and Suresh method to an actual cement factory; finding the method’s applicability, assumptions and limitations</td>
</tr>
<tr>
<td>El-Rayes and Kandil (2005)</td>
<td>Genetic algorithm</td>
<td>Appyling their model in highway construction projects; quantifying quality by some quality indices; calculating the project quality by addetive weighting of activities quality</td>
</tr>
<tr>
<td>Tareghian and Taheri (2006)</td>
<td>Integer programming</td>
<td>Developing a method to pruning the activities executions modes</td>
</tr>
<tr>
<td>Pollack-Johnson and Liberatore (2006)</td>
<td>Goal programming</td>
<td>Conceptualisation of quality in projects; quantifying quality of each activity execution mode by AHP; developing a goal programming model with four objectives namely time, cost, minimum quality and mean of quality</td>
</tr>
<tr>
<td>Tareghian and Taheri (2007)</td>
<td>Electromagnetic scatter search</td>
<td>Validating and checking the applicability of their algorithm by solving a randomly generated large scale problem with 19,900 activities</td>
</tr>
<tr>
<td>Afshar et al. (2007)</td>
<td>Multi-colony ant algorithm</td>
<td>Solving an example and comparing their algorithm’s results with some other algorithms</td>
</tr>
<tr>
<td>Zhang and Xing (2010)</td>
<td>Particle swarm optimisation</td>
<td>Considering construction methods instead of execution modes for each activity; Using Fuzzy numbers to describe time, cost, and quality; fuzzy multi-attribute utility methodology and constrained fuzzy arithmetic operators to evaluate each construction method; demonstrating the effectiveness of their algorithm by solving a bridge construction project TCQT problem</td>
</tr>
<tr>
<td>Kim et al. (2012)</td>
<td>Mixed integer linear programming</td>
<td>Focusing on minimising quality loss cost instead of maximising the individual activity quality of projects; validating their model by applying it to a robot type palletising system installation project</td>
</tr>
</tbody>
</table>

Several exact and approximation procedures were developed for small-size instances of trade-off problem (Skutella, 1998; Sunde and Lichtenberg, 1995; Burns et al., 1996; Demeulemeester et al., 2000). Heuristic and Meta-heuristic algorithms represented better...
results for medium and large-size instances (Rahimi and Iranmanesh, 2008). Meta-heuristics have also represented proper results for multi-objective versions of trade-off problems. Other assumptions such as time-switch constraints have been introduced in the literature of trade-off problems (Vanhoucke, 2005). Table 1 represents some related studies about trade-off problems.

2.2 Brief literature of MODM

Formally, an MODM model considers a vector of decision variables, objective functions, and constraints. DMs attempt to optimise the objective functions. Since this problem has rarely a unique solution, DMs are expected to choose a solution from the set of efficient solutions. Generally, the MODM problem with minimum objective functions can be formulated as follows.

\[
\begin{align*}
\text{min} & \quad f(x) \\
\text{s.t.} & \quad x \in S = \{ x \in \mathbb{R}^n \mid g(x) \leq b, x \geq 0 \}
\end{align*}
\]

where \(f(x)\) represents \(k\) conflicting objective functions, \(\theta(x) \leq b\) represents \(m\) constraints, \(S\) is feasible solution space, and \(x\) is an \(n\)-vector of decision variables, \(x \in \mathbb{R}^n\). The following definitions have worth to be mentioned for a MODM problem (Hwang and Masud, 1979).

**Definition 1**

\(x^*\) is said to be a complete optimal solution, if and only if there exists \(x^* \in X\) such that \(f_i(x^*) \leq f_i(x), i = 1, \ldots, k\), for all \(x \in X\). Also, ideal solution, superior solution, or utopia point are equivalent terms indicating a complete optimal solution. In general, such a complete optimal solution that simultaneously minimises all objective functions does not always exist when the objective functions conflict with each other.

**Definition 2**

\(x^*\) is said to be a Pareto optimal solution, if and only if there does not exist another \(x \in X\) such that \(f_i(x) \leq f_i(x^*)\) for all \(i\) and \(f_j(x) < f_j(x^*)\) for at least one \(j\). The Pareto optimal solution is also named differently by different disciplines: non-dominated solution, non-inferior solution, efficient solution, and non-dominate solution.

**Definition 3**

\(x^*\) is said to be a weak Pareto optimal solution, if and only if there does not exist another \(x \in X\) such that \(f_i(x) \leq f_i(x^*), i = 1, \ldots, k\).

Here, let \(X^{CO}, X^{P}, X^{WP}\) denote complete optimal, Pareto optimal, and weak Pareto optimal solution sets, respectively. Then from above definitions, we can easily get \(X^{CO} \subseteq X^{P} \subseteq X^{WP}\).

A satisfactory solution is a reduced subset of the feasible set that exceeds all of the aspiration levels of DM. A set of satisfactory solutions is composed of acceptable alternatives. Satisfactory solutions do not need to be non-dominated. A preferred solution is a non-dominated solution selected as the final choice through DMs’ involvement in the information processing.

During the process of decision making, some preference articulation from DM may be required. Based on type and time of given information by DM, the methods for
solving MODM problems have been systematically categorised into four classes by (Hwang and Masud, 1979). In all aforementioned categories, several exact procedures (Chankong and Haimes, 1983) and heuristic methods (Yu and Gen, 2010; Coello Coello et al., 2007) have been proposed.

In one of the aforementioned classes, there is a posterior articulation of preference information on priority of objective functions. In this case generating non-dominated solutions on Pareto front of MODM problem is desirable. The methods in this class strictly deal with constraints and do not consider the preference of DMs. The desired outcome, however, is to narrow the possible courses of actions and select the preferred course of action easier. They are also called non-dominated solutions generation methods.

2.3 Literature of GAs and its applications

GAs are stochastic search and optimisation techniques which are the most known types of evolutionary computations (ECs). GAs have been widely studied and applied to solve a variety of combinatorial optimisation problems. GAs has five basic components as genetic representation, genetic operators, evaluation function, construction of initial population, and values of parameters (Gen and Cheng, 2000).

Some of recent applications of GAs in several engineering and management areas are represented as follows. Sangwan and Kodali (2011) have proposed a mathematical model for the integrated design of cellular manufacturing systems problem. They have used an intelligent hybrid procedure, based on genetic algorithm and simulated annealing. The model has been validated using some instances from the literature. Ashoka Varthanan et al. (2010) have suggested an integer nonlinear programming model for aggregate production-distribution plan problem and solved it using an exact procedure as well as a genetic algorithm. The results showed that genetic algorithm outperforms the exact procedure. Ramasubramaniam et al. (2010) have proposed an integer linear programming model for a scheduling problem in steel casting industry. The objective of the model is minimising the maximum completion time of all the jobs. They have used a number of heuristic algorithms as well as a genetic algorithm to solve the problem. The computational experiment results have showed that the genetic algorithm can provide a more efficient solution in comparison with heuristic algorithms. Sivakumar and Shahabudeen (2008) have designed an adaptive Kanban system which its parameters set by genetic algorithm. Results indicate using genetic algorithm leads to better solution with improved computational efficiency in comparison with traditional Kanban system design methods. Raja et al. (2008) have studied a parallel-machine scheduling problem. They have proposed a genetic algorithm with fuzzy approach to select the optimal solution. The results have been compared with a known genetic algorithm with partially mapped crossover as well as a genetic algorithm with multi-component uniform order-based crossover. The comparison indicated that their proposed fuzzy-genetic algorithm generates better results than two aforementioned GAs. Ganesh and Narendan (2007) have studied the operational decision-making in vehicle routing problem with delivery, pick-up and time windows in the distribution of blood. They use a heuristic method in assigning vehicles to routes and a genetic algorithm with innovative crossover and mutation to solve the problem. Computational results revealed that their heuristic genetic algorithm can solve problems effectively.
Among meta-heuristic methods, evolutionary computation procedures reported proper performance for solving multi-objective problems during last decades (Coello Coello et al., 2007). Since introduction of GAs, different types of GAs have proposed for multi-objective optimisation problems. Among them, NSGA-II is one of the most applicable types of multi-objective evolutionary algorithms (MOEAs). As NSGA-II contains proper properties, such as elitism, fast non-dominated sorting and diversity maintenance along the Pareto-optimal front, so it has been successfully applied in a wide range of engineering, management, and combinatorial optimisation problems. NSGA-II has also been used as a validated comparison reference for other new developed/extended meta-heuristics. Recently, Zhou et al. (2011) have accomplished a comprehensive survey of the state of the art of MOEAs.

3 Problem formulation and notation

Consider a project defined as an acyclic directed graph \( G = (V, E) \), where \( V \) denotes the set of nodes representing activities and \( E \) denotes the set of arcs representing relations in the AON type of project representation. Each activity, \( ij \in V \), can be executed in \( r(i) \) different modes which mode \( k \) has its own time \( (t_{ik}) \), cost \( (c_{ik}) \) and quality \( (q_{ik}) \). Generalised precedence relations, in which start and finish of sequential activities may relate, are occurred in real life projects. To consider generalised precedence constraints in the model, the set of arcs, \( E \), partitions into four subsets ESS, ESF, EFS, and EFF, which respectively denote the sets of start-to-start, start-to-finish, finish-to-start, and finish-to-finish precedence relations. The notations used to formulate the GPDTCP demonstrated in Table 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Statement</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>Number of project activities</td>
<td>( i = 1, 2, \ldots, n )</td>
</tr>
<tr>
<td>( C )</td>
<td>Project cost threshold</td>
<td></td>
</tr>
<tr>
<td>( T )</td>
<td>Project deadline</td>
<td></td>
</tr>
<tr>
<td>( Q )</td>
<td>Project minimum desired level of quality</td>
<td></td>
</tr>
<tr>
<td>( r(i) )</td>
<td>Number of execution modes for activity ( t )</td>
<td>( i = 1, 2, \ldots, n )</td>
</tr>
<tr>
<td>( t_{ik} )</td>
<td>Time of activity ( i ) in mode ( k )</td>
<td>( i = 1, 2, \ldots, n; k = 1, 2, \ldots, r(i) )</td>
</tr>
<tr>
<td>( c_{ik} )</td>
<td>Cost of activity ( i ) in mode ( k )</td>
<td>( i = 1, 2, \ldots, n; k = 1, 2, \ldots, r(i) )</td>
</tr>
<tr>
<td>( q_{ik} )</td>
<td>Quality of activity ( i ) in mode ( k )</td>
<td>( i = 1, 2, \ldots, n; k = 1, 2, \ldots, r(i) )</td>
</tr>
<tr>
<td>( d_{ij} )</td>
<td>Time lag between precedent activity ( i ) and activity ( j )</td>
<td>( i = 1, 2, \ldots, n; j = 1, 2, \ldots, n )</td>
</tr>
<tr>
<td>( w_{i} )</td>
<td>Weight of contribution in project quality for activity ( i )</td>
<td>( i = 1, 2, \ldots, n );</td>
</tr>
<tr>
<td>( s_{i} )</td>
<td>Positive decision variable: start time of activity ( i )</td>
<td>( i = 1, 2, \ldots, n );</td>
</tr>
<tr>
<td>( x_{ij} )</td>
<td>Binary decision variable: equals 1 if activity ( i ) executes in mode ( k ) and is 0, otherwise</td>
<td>( i = 1, 2, \ldots, n; j = 1, 2, \ldots, r(i) )</td>
</tr>
</tbody>
</table>
The objectives are to minimise the overall time and cost and maximise the overall quality of the project, by selecting a right mode for all of project activities. We consider a multi-objective mixed-integer mathematical model for generalised precedence time-cost-quality trade-off problem (GPDTCQP). The objectives are conflictive and there exists no prior articulation of DM preferences on priority of different objectives. Hence, generating a set of non-dominated solution on Pareto front of the problem is preferred.

The model (1) to (13) represents the GPDTCQP problem. (1) to (3) are allocated to describe time, cost and quality objective functions, respectively. Equation (1) tries to minimise the start time of the last activity of the project. Hence, minimisation of decision variable $S_n$ means minimising duration of project. Equation (2) seeks the minimum value of cost of all activities of the project in all execution modes. Equation (3) seeks the maximum weighted average value of quality of all activities of the project in all execution modes. Constraint (4) guarantees the selection of one and only one mode for each activity. Constraints (5) to (8) preserve the generalised relations between project activities. Constraint (5), which should be written for all arcs such as $(i, j)$ which belong to sub-set of finish-to-start, ensures that the start time of activity $j$ should be greater than or equal to finish time of activity $i$ plus a predetermined time lag $d_{ij}$. Constraint (6), which should be written for all arcs such as $(i, j)$ which belong to sub-set of start-to-finish, ensures that the start time of activity $i$ plus a predetermined time lag $d_{ij}$ should be less than or equal to finish time of activity $j$. Constraint (7), which should be written for all arcs such as $(i, j)$ which belong to sub-set of start-to-start, ensures that the start time of activity $i$ plus a predetermined time lag $d_{ij}$ should be less than or equal to start time of activity $j$. Constraint (8), which should be written for all arcs such as $(i, j)$ which belong to sub-set of finish-to-finish, ensures that the finish time of activity $i$ plus a predetermined time lag $d_{ij}$ should be less than or equal to finish time of activity $j$. Constraint (9) defines the upper bound of cost of the project for TCQTP. Constraint (10) defines the maximum allowed time of the project for TCQTP. Constraint (11) defines the minimum required quality level of the project for TCQTP. Constraint (12) ensures each activity not to start earlier than zero. Constraint (13) represents binary type of decision variables.

$$\begin{align*}
\text{Min Time} & = S_n \\
\text{Min Cost} & = \sum_{k=1}^{r(l)} \sum_{i=1}^{x_{ik}} x_{ik} c_{ik} \\
\text{Max Quality} & = \left( \sum_{k=1}^{r(l)} \sum_{i=1}^{x_{ik}} w_i x_{ik} q_{ik} \right) / \sum_{i=1}^{r(l)} w_i \\
\text{s.t.} & \\
\sum_{k=1}^{r(l)} x_{ik} & = 1 \quad \forall i \in V \\
s_i \sum_{k=1}^{r(l)} t_{ik} x_{ik} + d_{ij} & \leq s_j \quad \forall (i, j) \in E_{FS}
\end{align*}$$
Solving a generalised precedence multi-objective

\[ s_i + d_{ij} \leq s_j + \sum_{k=1}^{\ell(i)} t_{ik} x_{jk} \quad \forall (i, j) \in E_{\text{EQ}} \]  
(6)

\[ s_i + d_{ij} \leq s_j \quad \forall (i, j) \in E_{\text{SS}} \]  
(7)

\[ s_i + \sum_{k=1}^{\ell(i)} t_{ik} x_{ik} + d_{ij} \leq s_j + \sum_{k=1}^{\ell(i)} t_{jk} x_{jk} \quad \forall (i, j) \in E_{\text{FF}} \]  
(8)

\[ \sum_{i \in V} \sum_{k=1}^{\ell(i)} x_{ik} c_{ik} \leq C \quad \forall (i, j) \in V \]  
(9)

\[ s_a \leq T \]  
(10)

\[ \sum_{i \in V} \sum_{k=1}^{\ell(i)} x_{ik} q_{ik} \geq Q \quad \forall (i, j) \in V \]  
(11)

\[ s_i \geq 0 \quad \forall i \in V \]  
(12)

\[ x_{ij} \in \{0, 1\} \quad \forall (i, j) \in V \]  
(13)

4 Proposed customised NSGA-II algorithm

In the GPDTCPQP problem the non-dominated solutions can be described as different execution modes devoting to the project activities which result to non-preferred amounts in project overall time, cost and quality. NSGA-II consists of assigning rank 1 to the set of non-dominated solutions, removing them from solution pool, and then recognising a new set of non-dominated solutions, ranked 2, and so on. Each solution of initial population is sorted based on non-domination into each front. The first front contains only non-dominant solutions among all solutions, while the solutions of the second front being dominated by the solutions in the first front only and the fronts goes so on (Deb et al., 2002).

Solutions in each front are assigned rank (fitness) values based on front where they belong to. Solutions in first front are given a fitness value of 1 and solutions in the second are assigned fitness value as 2 and so on. In addition to fitness value a parameter, called crowding distance, is calculated for each individual of a population. The crowding distance is a measure of how close an individual is to its neighbours. The large average crowding distance will result in a better diversity in the population.

Parents are selected from the population by using binary tournament selection based on the rank and crowding distance. The fitness of an individual, which belongs to a given rank, is lesser than the others in the same rank with a greater crowding distance measurement. The selected population generates offspring using crossover and mutation operators. The current population and the current offspring are sorted again based on non-domination and crowding distance. Only the best \( N \) individuals are selected, in which \( N \) is the population size (Deb et al., 2002).

As the GPDTCPQP problem has three objectives including minimising time and cost while maximising quality, NSGA-II has been customised and supplied to generate...
different sets of non-dominated solutions. Because the GPDTCPQ problem is a constraint multi-objective optimisation problem, so we supply the customised NSGA-II with a constraint handling strategy. The operators of NSGA-II such as crossover, mutation, crowding distance, fast ranking, and selection has widely been discussed (Deb et al., 2002). So, the main customisation and constraint handling strategy have been described here.

4.1 Main customisation of proposed NSGA-II algorithm for GPDTCPQ

Considering a project with \( n \) activities and \( r(r = 1, 2, \ldots, k) \) different execution modes for each activity, a chromosome in which each cell of it containing the number of selected mode for corresponding activity have been constructed.

A single point cut has been selected as crossover operator. According to a crossover probability, called \( P_c \), two parents are selected and the crossover is accomplished. Value of a randomly selected cell in a given chromosome is changed based on mutation probability; called \( P_m \). The fitness function is calculated based on objective functions of model (1) to (13) for a given chromosome.

The \( P_c \) and \( P_m \) are determined dynamically using linear equations which relate the crossover and mutation rate to iteration number of the algorithm. The \( P_c \) is set close to 1 at first iterations of the algorithm and is decreased as the iterations go on. The \( P_m \) is configured equal to a small value at first iteration of the algorithm and is increased as the last iterations. Dynamic parameter tuning causes better exploration and exploitation.

For handling the constraints of model (1) to (13), we supplied our customised NSGA-II algorithm with combination of penalty and elimination approaches. The time window, cost window and quality window of model (1) to (13) have been satisfied with a heuristic self-adaptive penalty approach. The generalised precedence constraints have been handled through elimination approach.

The fitness of each chromosome (candidate solution) may violate some constraints specifically the time window, cost window, and quality window constraints. In these situations the value of violation by each chromosome in the population calculated with relations (14) to (16).

\[
\begin{align*}
  v_{i1} &= \text{Max}\{0, \text{Time} - T\} \\
  v_{i2} &= \text{Max}\{0, \text{Cost} - C\} \\
  v_{i3} &= \text{Max}\{0, Q - \text{Quality}\}
\end{align*}
\]

where \( v_{i1}, v_{i2}, \) and \( v_{i3} \) are violation values of time, cost, and quality dimension of chromosome \( i \) in the population, respectively.

Based on calculated violation values of each chromosome using (14) to (16), the minimum value of violations in whole population, the iteration number which algorithm is in, and a dynamic self-adaptive penalty approach has been supplied. In this approach a self-adaptive penalty function prepared to calculate the penalty value that has to apply to the NSGA-II fitness functions. The penalty is calculated as (17) to (19) and applied to the fitness functions as (20) to (22).

\[
\text{TimePenalty} = \left( \frac{v_{i1}}{v_{i1}^\text{min}} \right)^\alpha \times t^\beta
\]
Solving a generalised precedence multi-objective

\[
\text{CostPenalty} = \left( \frac{v_{12}}{v_{12}^{\text{min}}} \right)^{\alpha} \times t^{\beta} \quad (18)
\]

\[
\text{QualityPenalty} = \left( \frac{v_{13}}{v_{13}^{\text{min}}} \right)^{\alpha} \times t^{\beta} \quad (19)
\]

\[
\text{Time}'_i = \text{Time}_i + \text{TimePenalty} \quad (20)
\]

\[
\text{Cost}'_i = \text{Cost}_i + \text{CostPenalty} \quad (21)
\]

\[
\text{Quality}'_i = \text{Quality}_i + \text{QualityPenalty} \quad (22)
\]

where \( \text{Time}'_i, \text{Cost}'_i, \) and \( \text{Quality}'_i \) are new values of time, cost, and quality objectives of violated chromosome \( i \) after penalise them. \( v_{12}, v_{13}, \) and \( v_{13} \) are violation values of time, cost, and quality, respectively. \( v^{\text{min}}_{12} = \min_i \{ \varepsilon + v_{12} \}, \) \( v^{\text{min}}_{13} = \min_i \{ \varepsilon + v_{13} \}, \) and \( v^{\text{min}}_{13} = \min_i \{ \varepsilon + v_{13} \} \) are minimum violation values of time, cost, and quality dimension of chromosomes in population, respectively. \( \varepsilon \) is a little positive value to avoid division by zero, \( t \) is iteration numbers, and \( \alpha \) and \( \beta \) are control parameters.

Each chromosome which does not satisfy the generalised precedence constraints eliminates during population initialisation and evolutionary operators.

5 Computational experiments

To investigate the efficiency of modified NSGA-II algorithm, it has been required to generate some problems. We have used a three phase procedure in problem generation including network generation phase, generalised precedence relations generation phase, and activities’ execution modes generation phase. For the first phase of problem generation, we have used RanGen which proposed by Demeulemeester et al. (2003) to generate instances of project networks. Table 3 illustrates the characteristics of generated instances.

<table>
<thead>
<tr>
<th>Instance no.</th>
<th>Number of activities</th>
<th>Number of arcs</th>
<th>Order strength*</th>
<th>Complexity index**</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>45</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>1,225</td>
<td>1</td>
<td>42</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>4,950</td>
<td>1</td>
<td>87</td>
</tr>
</tbody>
</table>

Notes: *The number of precedence relations divided by the theoretical maximum number of precedence relations in the network. **Measure of the closeness of a network to a series-parallel directed graph

The modified NSGA-II algorithm was coded using VBA for MS-Excel 12.0. All codes were run on a Pentium IV PC with MS-Windows 7 Home edition, 2 GB of RAM, and 2.0 GHz Core 2 Due CPU. The most fitted parameter amounts have been categorised based on problem instances and illustrated in Table 4.
Table 4  Fitted parameters of algorithm

<table>
<thead>
<tr>
<th></th>
<th>Problem instance 1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NSGA-II parameters</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chromosome no.</td>
<td>200</td>
<td>Archive size</td>
<td>30</td>
<td>Maximum iteration no.</td>
<td>100</td>
<td>Cross rate</td>
</tr>
<tr>
<td>Alpha</td>
<td>1</td>
<td>Beta</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Problem instance 2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NSGA-II parameters</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chromosome no.</td>
<td>200</td>
<td>Archive size</td>
<td>30</td>
<td>Maximum iteration no.</td>
<td>100</td>
<td>Cross rate</td>
</tr>
<tr>
<td>Alpha</td>
<td>1</td>
<td>Beta</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Problem instance 3</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NSGA-II parameters</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chromosome no.</td>
<td>200</td>
<td>Archive size</td>
<td>30</td>
<td>Maximum iteration no.</td>
<td>100</td>
<td>Cross rate</td>
</tr>
<tr>
<td>Alpha</td>
<td>1</td>
<td>Beta</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The modified NSGA-II was implemented on three different problem instances (i.e., small scale, medium scale, and large scale) as illustrated in Table 4. The results are to be discussed in this sub-section. Figure 1 represents the generated non-dominate solutions. As it is clear the number of non-dominated solutions in all instances is promising. Moreover, the variance of non-dominated solutions over the several instances is a small value which means the robustness of the proposed algorithm in each of single run.

As shown in Figure 1 the following notes are interesting to be mentioned. The CPU time of NSGA-II algorithm was 40, 130, and 400 seconds for small, medium, and large instances, respectively. Considering the structure, hardness, and complexity of the test problems, this amount of CPU time is promising.
Solving a generalised precedence multi-objective

Figure 1  Generated non-dominated solutions. (a) modified-NSGA-II (small size instance) (b) modified-NSGA-II (medium size instance) (c) modified-NSGA-II (large size instance) (see online version for colours)
Figure 2  The reference sets of different benchmark instances, (a) RS for small size instance (b) RS for medium size instance (c) RS for large size instance (see online version for colours)
As the real Pareto front of the benchmark instances are not known in advance so, a reference set (RS) was defined for each benchmark instances. A RS is the best known solutions of a benchmark instance throughout the ten different runs. On the other hand RS is assumed as all of non-dominated solutions of an instance during all of ten runs. Figure 2 represents the RSs for three categories of benchmark instances. Comparatively, it is clear that there is no significant difference between the generated Pareto solutions of a single run and RS of all runs. Hence, the stability of the proposed procedure is perceived.

**Figure 3** Achieved Objective functions of NSGA-II vs. RS for small scale instance, (a) time (b) cost (c) quality (see online version for colours)
Figure 3 represents achieved objective functions of a sample run of NSGA-II against RS for small scale instance. This may help to recognise the relative proportional of a sample run in construction of RS. Plotting the achieved solutions against RS in medium and large size instances has no practical use as the RS are approximately the same generated solutions by a single run of NSGA-II algorithm.

As it is clear, the distance of a single run of the proposed procedure to RS is small. this means that the quality of re-generated Pareto front is promising in a single run of the proposed procedure.

6 Conclusions

In real world of project scheduling the activities of a project can perform in several ways with certain time, cost and quality amounts so called multi-mode execution activities. Also there are several types of precedence relations between the activities of a project. Considering these and the fact that in practice the resources are in discrete units, we model a problem namely generalised precedence discrete time-cost-quality trade-off (GPDTCQTP) using mixed integer programming.

Due to our best knowledge, there exists no similar research work in literature of trade-off problems which contributed to generate non-dominated solutions in presence of no articulation of preference relations between conflictive objectives. The activities of the GPDTCQTP are assumed to have several executing modes with specified time-cost-quality in each mode. The GPDTCQTP serves several real and practical issues as follows. The GPDTCQTP have three conflictive objectives and generalised precedencies are considered between activities. The proper upper and lower bounds are assumed for time, cost, and quality of the project. All these assumptions are practical and are to be faced in real projects. Due to our best knowledge, there is no unique trade-off model which considered all aforementioned properties, concurrently.

A new mixed-integer mathematical formulation is proposed for GPDTCQTP. As it is hard to represents the preference articulation of DMs in form of relative importance of objective functions so, it is interesting to generate set of non-dominated solutions for GPDTCQTP. Hence, a dynamic self-adaptive version of NSGA-II algorithm is represented to generate set of non-dominate solutions on Pareto front of instances of GPDTCQTP.

Since this problem is multi-objective in its nature, we have proposed a well-known multi-objective evolutionary algorithm namely non-dominated solution genetic algorithm (NSGA-II), which customised and modified for this problem. The NSGA-II algorithm was supplied by a dynamic self-adaptive penalty function and dynamic parameter tuning function. The dynamic self-adaptive penalty function penalised the violated chromosomes according to status of the chromosome, status of the entire population and iteration of the algorithm. By dynamic parameter tuning, the cross over, mutation and penalty parameters were modified based on iteration number. To investigate the efficiency and applicability of the method, we generated several series of problems (i.e., small scale, medium scale, and large scale) through a problem generation algorithm. The procedure was straightforward and the results were promising.

The proposed procedure can be developed in form of a decision support system (DSS) in order to handle the real life cases. Also, the proposed procedure of this paper can be compared with other exact and meta-heuristic methods. Moreover, some
performance metrics as well as a well-defined statistical analysis may be provided in order to compare several solution procedures.

There were some limitations in our proposed procedure. Relaxation of them can result in further researches. Some of them are as follows. In real life projects, activities may pre-empt during implementation. The assumption of pre-emption can be supplied to the proposed model of this paper. In real cases, the exact value of time, cost, and quality of activities may not be achievable, so this type of uncertainty may be modelled using fuzzy sets.

Acknowledgements

The authors would like to thank the great comments of the anonymous reviewers which really improved quality of the paper.

References


