In project scheduling, it is feasible to reduce the duration of a project by allocating additional resources to its activities. However, crashing the project schedule will impose additional costs. Numerous research has focused on optimizing the trade-off between time and cost to achieve a set of non-dominated solutions. However, the majority of the research on time-cost trade-off problem developed methods for relatively simple problems including up to eighteen activities, which are not representing the complexity of real-life construction projects. In this work a Particle Swarm Optimization (PSO) technique is presented for Pareto oriented optimization of the complex discrete time-cost trade-off problems. The proposed PSO engages novel principles for representation and position-updating of the particles. The performance of the PSO is compared to the existing methods using a well-known 18-activity benchmark problem. A 63-activity problem is also included in computational experiments to validate the efficiency and effectiveness of the proposed PSO for a more complex problem. The results indicate that the proposed method provides a powerful alternative for the Pareto front optimization of DTCTPs.

Keywords: Pareto front, particle swarm optimization, project scheduling, time-cost trade-off problem.

INTRODUCTION

Opposed to other industries, transient nature of the construction projects imposes heavy burden on decision makers regarding unequivocal optimal devotions of time, cost, and resources (Hegazy 1999). Either at the planning stage or in case the project is running behind the scheduled plan, the contractor and the client, as the main parties to a construction project normally strive to complete the project within shorter durations at the minimum cost possible. However, crashing the project schedule imposes additional costs and might be profitable up to a certain limit (Zheng et al. 2005). Accordingly, analyses of the trade-off between time and cost in view of obtaining a set of non-dominated solutions is a significant aspect of the construction management. Classical network analyses like critical path method (CPM), in essence, merely incorporate the cost and project deadline aspects (Aminbakhsh 2013). Such methods attempt to minimize the project duration regardless of the availability of resources (both money and physical resources). Any reduction in project duration is facilitated by crashing the project schedule. Decision makers speed up the project by using additional labour and machinery or by adopting alternative construction techniques (Hegazy 1999). The best combination of crashing alternatives is facilitated...
by optimizing the balance between the direct and indirect costs of a project. The trade-off between time and cost of a project is known as the time-cost trade-off problem (TCTP) (Feng et al. 1997), discrete version of which (DTCTP) considers discrete sets of time-cost options for the activities. Such a concern is imperative to TCT analyses as real-life projects comprise resources of discrete units. In Pareto oriented optimization of DTCTPs, non-dominated optimal total costs are mapped to feasible completion times to generate Pareto fronts of the problems.

Commercial scheduling software packages, in general, do not bear any strategies for the DTCTP. The methods proposed for the DTCTP can be classified into three categories: exact methods, heuristics, and meta-heuristics. Exact methods based on mixed integer programming (De et al. 1995; Szmerekovsky and Venkateshan 2012), branch and bound (Demeulemeester et al. 1998; Vanhoucke 2005), and Benders decomposition (Hazır et al. 2010) can solve small scale problems. Few research work has focused on the development of heuristic methods for the time-cost trade-off problem (Prager 1963, Siemens 1971; Goyal 1975, Moselhi 1993). The most recent heuristic method for DTCTP is implemented by Hegazy (1999) which is rooted upon Siemens's (1971) method. In recent years the significant developments in meta-heuristic algorithms has enabled researchers to shift focus from heuristics to meta-heuristic methods for DTCTP. Genetic algorithms (Feng et al. 1997; Hegazy 1999; Zheng et al. 2005; Kandil and El-Rayes 2006; Eshtehardian et al. 2008; Fallah-Mehdipour et al. 2012; Sonmez and Bettemir 2012), ant colony optimization (Ng and Zhang 2008; Xiong and Kuang 2008; Afshar et al. 2009), particle swarm optimization (Yang 2007; Zhang and Xing 2010; Fallah-Mehdipour et al. 2012), shuffled frog leaping (Elbeltagi et al. 2007), and Electimize (Abdel-Raheem and Khalafallah 2011) are among the meta-heuristics implemented for the TCTP. Despite Pareto front optimization is reckoned to be the ultimate resolution of TCT analyses (e.g. Zheng et al. 2005; Yang 2007; Eshtehardian et al. 2008), relatively scarce devotions are made toward this end. The majority of the existing research in their computational experiments has focused on solving relatively simple problems that include up to only 18 activities with 5.9E09 number of possible resource utilization options. The main objective of this study is to present an efficient PSO model incorporating novel principles for binary representation and position-updating of the particles which is capable of exerting the Pareto front optimization for complex DTCTP. The practiced complex instance includes 63 activities with 1.37E42 number of possible combinations of time-cost alternatives.

**PARETO FRONT AND OPTIMIZATION MODEL**

Pareto front optimization is a multi-objective decision making problem that any of its objectives might reach their optimal amounts at miscellaneous points called the Pareto Front. Obtaining the Pareto front for TCT problem, in essence, engages concurrent optimization of two classical TCT extensions, viz., the budget and the deadline problems. The bi-criterion TCT optimization problem can be formulated as follows:

\[
\text{Minimize } y = (D_i, C_i)
\]

\[
D_i = \max \left[ \sum_{j=1}^{S} \sum_{k=1}^{m} d_{jk}^{(t)} x_{jk}^{(t)} \right]
\]

\[
C_i = \sum_{j=1}^{S} \sum_{k=1}^{m} c_{jk}^{(t)} x_{jk}^{(t)} + D_i \times ic
\]
Pareto oriented optimization

where \( D_i \) and \( C_i \) denote total duration and total cost of \( i \)th solution, respectively; \( d_{jk} \) and \( dc_{jk} \) represent duration and direct cost of the \( k \)th options of the \( j \)th activity, respectively; and \( ic \) denotes the daily indirect cost.

**PROPOSED PARTICLE SWARM OPTIMIZATION METHOD**

The PSO method, imitating bird flocks that forage and fly in unison, was first introduced by Kennedy and Eberhart (Eberhart and Kennedy 1995; Kennedy and Eberhart 1995) who later developed a binary paradigm of their model for problems incorporating discrete objective functions (Kennedy and Eberhart 1997). In the proposed PSO method, each particle represents a solution in a \( S \) -dimensional solution space. The position and velocity of \( i \)th particle for the \( k \)th option of the \( j \)th activity, in the time step \( t \) is represented by \( x_{ijk}^{(t)} \) and \( v_{ijk}^{(t)} \), respectively. Moreover, the proposed PSO involves binary values, that is, each particle \( i \), for its \( j \)th activity can only have a single \( k \) set to one, with all the remaining components of the \( j \)th activity holding zero (viz., for a solution generated by PSO, at time-step \( t \): \( \sum_{k=1}^{m(j)} x_{ijk}^{(t)} = 1 \)). An external archive, \( O \) has been dedicated to the PSO model, so as to store all the non-dominated solutions found by this algorithm. A controller, Eq. (4), is implemented to carry out judgments regarding particles’ qualification to enter the external archive (Aminbakhsh 2013). For any decision vector \( x \), this controller engages the following criteria with respect to the measured \( D_x \) and \( C_x \):

\[
\begin{align*}
\text{Accept} & \quad \text{if } \quad D_x \neq D_y \\
\text{or} & \quad \left\{ \begin{array}{l}
D_x = D_y \\
C_x \leq C_y
\end{array} \right. \\
\text{Reject} & \quad \text{otherwise}
\end{align*}
\]

where; \( D_y \) and \( C_y \), respectively, represent duration and cost of particle \( y \) – an existing solution stored in the archive \( O \). The quality of the solutions are compared to determine the personal best positions, \( P_i \)'s, experienced by any of the particles.

According to Eq. (5), for decision vectors \( u \) and \( v \):

\[
(5) \quad u > v \quad \text{if} \quad \begin{cases} 
C_u < C_v \quad \text{or} \\
C_u = C_v \quad \text{and} \quad D_u < D_v
\end{cases}
\]

Since all the archived solutions are non-dominated with equal qualities, in order to provide dynamic exploitation of the archived solutions, PSO utilizes an effective multi-objective approach in measuring global best positions, \( P_g \)'s, of the particles. Throughout each iteration, PSO randomly selects non-dominated solutions as the \( P_g \)'s of the particles. Particles fly to their new positions using the velocity vector given in Eq. (6),

\[
(6) \quad v_{ijk}^{(t+1)} = w^{(t)} v_{ijk}^{(t)} + c_1 r_1 \left( P_{ijk}^{(t)} - x_{ijk}^{(t)} \right) + c_2 r_2 \left( P_g^{(t)} - x_{ijk}^{(t)} \right)
\]
where; $w$ is the inertia weight; $r_1$ and $r_2$ are uniformly distributed random vectors between [0,1]; and the constants $c_1$ and $c_2$ are the cognitive and social parameters, respectively.

In the proposed PSO, velocity vectors are transformed into probabilities (Aminbakhsh 2013) and are normalized to the range [0,1] using a logistic transformation function given in Eq. (7). Then, subject to the probabilistic condition specified in Eq. (8), each particle moves to a new position in the solution space.

(7) \[
\text{sig}(v_{ijk}^{(r)}) = \frac{1}{1 + \exp(-v_{ijk}^{(r)})}
\]

(8) \[
X_{ijk}^{(r+1)} = \begin{cases} 
1 & \text{if } \text{sig}(v_{ijk}^{(r+1)}) = \max \left\{ \text{sig}(v_{ijk}^{(r+1)}) \right\} \\
0 & \text{otherwise}
\end{cases}
\]

The optimization process is reiterated until the stipulated number of iterations is reached. Subsequently, PSO returns the ultimate archived non-dominated solutions.

**TEST INSTANCES**

For performance evaluation of the proposed PSO method, a small-scale problem, as well as a more complex medium scale instance is implemented.

1) Small-scale test instance

The first test problem involves the 18-activity network, details of which can be derived from Feng et al. (1997) incorporating the time-cost alternatives defined in Hegazy (1999). The majority of the previous research (Zheng et al. 2005; Ng and Zhang 2008; Afshar et al. 2009; Zhang and Ng 2012) used this problem to evaluate the performances of the proposed multi-objective meta-heuristics. This problem with a total of 5.9E09 possible schedules is examined with a daily indirect cost of $1,500.

2) Medium-scale test instance

In the course of the analyses, Sonmez and Bettemir’s (2012) hypothetical 63-activity problem is used as the second test instance. This medium scale complex problem consists of 1.37E42 different time-cost alternatives and a daily indirect cost of $2,300.

**COMPUTATIONAL EXPERIMENTS AND COMPARISONS**

Computational experiments were conducted to evaluate the performance of the proposed PSO method for Pareto front optimization of DTCTP using benchmark instances. The proposed algorithm was coded in C++ and compiled in Visual Studio 2013. All of the tests were carried out on a computer with an Intel Core i7-3.40 GHz CPU. Pilot experiments were conducted to determine an adequate set of parameter values for the PSO algorithm. The pilot experiments revealed that the set of parameters that are summarized in Table 1 provides an adequate combination for the PSO. 200,000 and 1,000,000 schedules (objective function evaluations) were used as the stopping criteria in experiments for the first and second problems, respectively.
Table 1: Parameter settings of the PSO

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Small-Scale Problem</th>
<th>Medium-Scale Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>No. of Birds</td>
<td>1,000</td>
<td>5,000</td>
</tr>
<tr>
<td>$c_1$</td>
<td>Cognitive Parameter</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>$c_2$</td>
<td>Social Parameter</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$w_{\text{max}}$</td>
<td>Max. Inertia Weight</td>
<td>1.2</td>
<td>1.2</td>
</tr>
<tr>
<td>$w_{\text{min}}$</td>
<td>Min. Inertia Weight</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>$v_{\text{max}}$</td>
<td>Max. Velocity</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

1) Small-scale test instance

Table 2 summarizes the results of the PSO along with the performance of four previous meta-heuristics for the 18-activity problem. Although none of the existing studies reveal the elapsed CPU times, the proposed PSO requires an acceptable CPU time of approximately 1.9 seconds to unravel the first instance by searching a mere 3.39E-05 fraction of the solution space. Solutions obtained by Zheng et al. (2005) are of inferior quality compared to the results of PSO, since, MAWA-GA’s solutions cost 0.9% to 1.55% more that PSO’s results. For $D = 110$ days, ACS-TCO of Ng and Zhang (2008) and ACS of Zhang and Ng (2012) provide a solution which costs more than the proposed PSO’s result. The Pareto front solutions reported for NA-ACO of Afshar et al. (2009) are identical to the results obtained by the PSO method. The comparison of PSO with the state-of-art methods reveal that proposed PSO is among the top performing algorithms for Pareto oriented optimization of the small-scale DTCTPs.

Table 2: Comparison of Pareto Fronts located for Small-Scale problem

<table>
<thead>
<tr>
<th>Duration (days)</th>
<th>Zheng et al. (2005)</th>
<th>Ng and Zhang (2008)</th>
<th>Afshar et al. (2009)</th>
<th>Zhang and Ng (2012)</th>
<th>PSO (This Study)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>287,720</td>
<td>283,320</td>
<td>283,320</td>
<td>283,320</td>
<td>283,320</td>
</tr>
<tr>
<td>101</td>
<td>284,020</td>
<td>279,820</td>
<td>279,820</td>
<td>279,820</td>
<td>279,820</td>
</tr>
<tr>
<td>104</td>
<td>280,020</td>
<td>276,320</td>
<td>276,320</td>
<td>276,320</td>
<td>276,320</td>
</tr>
<tr>
<td>110</td>
<td>273,720</td>
<td>271,320</td>
<td>271,270</td>
<td>271,320</td>
<td>271,270</td>
</tr>
</tbody>
</table>

2) Medium-scale test instance

Sonmez and Bettemir (2012) presented a medium-scale 63-activity DTCTP for the single objective cost optimization problem. To our best knowledge, the problem is first solved for Pareto front optimization in this work. Computational experiments of the proposed PSO model involving this instance revealed promising results since PSO was able to locate 38 non-dominated solutions by searching only 7.30E-37 fraction of the solution space. Pareto front of this larger instance was achieved within a reasonable CPU time of 37.5 seconds, for the first time. Lack of earlier attempts for Pareto oriented optimization of this problem ruled out a comparative presentation of the results obtained.
CONCLUSIONS

A discrete PSO method for the Pareto front optimization of discrete time-cost tradeoff problem is presented in this study. Novel principles for binary representation, and position-updating of the particles are implemented in the proposed PSO method. It is shown that PSO operates to unravel the multi-criteria DTCTP by searching a very small fraction of the search space. The results of the computational experiments reveal that PSO can achieve high quality solutions for a small benchmark DTCTP. For a more complex instance, including 63 activities, PSO was able to obtain 38 non-dominated solutions within seconds, for the first time. The results indicate that the proposed method is among the top performing algorithms, providing a powerful alternative for the DTCTP Pareto front optimization. However, research focusing on generation of large-scale complex DTCTP instances would enable a better understanding of the performance of heuristics and meta-heuristics for the real-life projects. Development of efficient discrete optimization methods for Pareto oriented optimization of large-scale real-life projects appears to be another promising area for the future research.

ACKNOWLEDGMENTS

This research is funded by The Scientific and Technological Research Council of Turkey (TÜBİTAK), under grant #213M253.

REFERENCES


