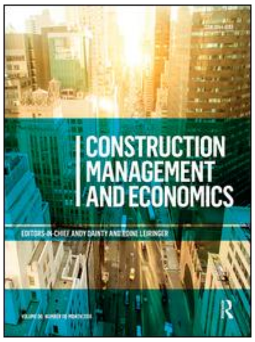

Routledge
Taylor & Francis Group

Construction Management and Economics

ISSN: 0144-6193 (Print) 1466-433X (Online) Journal homepage: https://www.tandfonline.com/loi/rcme20

Multi-objective particle swarm optimization for construction time-cost tradeoff problems

Hong Zhang & Heng Li

Construction Management and Economics (Janaury 2010) **28**, 75–88

Routledge
Taylor & Francis Group

Multi-objective particle swarm optimization for construction time–cost tradeoff problems

HONG ZHANG[1]* and HENG LI[2]

[1]*School of Civil Engineering, Shenzhen University, Nanhai Ave 3688, Shenzhen, China*
[2]*Department of Building and Real Estate, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*

Received 3 April 2009; accepted 11 October 2009

Reduction in construction project duration is generally linked to additional cost due to more expensive resources required. Hence tradeoff between time and cost is crucial to the efficiency of a construction project. The time-cost tradeoff (TCT) issue has been studied through various multi-objective optimization methodologies to determine an optimal set of activity methods with the objectives of minimizing project duration and total cost. A multi-objective particle swarm optimization that adopts a combined scheme for determining the global best of each particle is presented for solving the TCT problem. The candidate TCT solutions in terms of a set of construction methods for activities are represented through the multidimensional particles. The framework of the combined scheme-based multi-objective particle swarm optimization (CSMOPSO) is developed. Computational analyses are conducted to investigate the performance of the CSMOPSO, including comparison with other methods. This study is expected to provide an alternative solving methodology for the TCT problem and help contractors or engineers plan construction methods with optimal time–cost tradeoff.

Keywords: Time–cost tradeoff, multi-objective optimization, particle swarm optimization (PSO), construction methods.

Introduction

Being the commonly used methods for construction project scheduling, the critical path method (CPM) and resource-constrained project scheduling (RCPS) approach have serious limitations in reflecting the competitive environment of the construction industry. The CPM assumes unlimited resources and is unable to consider project duration reduction. Both the CPM and RCPS approaches could not handle the issue of cost reduction. In addition to reducing project duration to meet a specific deadline, reduction of cost in the pursuit of higher profit is also desirable. Hence, reduction of both time and cost has become the major criterion for evaluating the performance of a construction project.

In general, the less expensive resources or technologies would lead to longer duration to complete an activity. For example, using more productive resources or technologies (e.g. using more efficient equipment, hiring more workers or adopting overtime) may save time, but the cost may rise. An appropriate balance or tradeoff between duration and cost is able to enhance the efficiency of a construction project. Therefore, the issue of minimizing both project duration and cost has become a time–cost tradeoff (TCT) problem that addresses how to determine a set of optimal solutions, i.e. construction methods for all activities, which correspond to various projects' duration and cost. To search for optimal solutions for TCT problems is difficult and time-consuming when the numbers of activities and construction methods are very large. What this paper addresses is a new methodology to search for solutions of the TCT problems.

Many attempts have been made to solve the TCT problem. These attempts adopt, in general, three types of techniques, i.e. heuristic methods, mathematical methods and evolutionary techniques such as genetic algorithm (GA). The heuristic methods depend on the rules of thumb, other than the exact mathematical models. The examples of the heuristic methods for

*Author for correspondence. E-mail: sdzhangh@szu.edu.cn

Construction Management and Economics
ISSN 0144-6193 print/ISSN 1466-433X online
http://www.informaworld.com
DOI: 10.1080/01446190903406170

solving the TCT problem include the work of Prager (1963), Siemens (1971) and Moselhi (1993). The heuristic methods are easy to understand and implement. The limitations of the heuristic methods include the inability to find the global or exact optima and to locate a series of optima, as well as consideration of a linear relationship rather than a discrete time–cost relationship for completing an activity (Feng *et al.*, 1997). The mathematical methods for solving the TCT problem include linear programming (Kelly, 1961; Hendrickson and Au, 1989; Pagnoni, 1990), integer programming (Meyer and Shaffer, 1963; Patterson and Huber, 1974), combination of linear programming and integer programming (Burns *et al.*, 1996) and dynamic programming (Robinson, 1975; Elmaghraby, 1993; De *et al.*, 1995). The mathematical methods are able to find the exact optima, but are criticized for the resultant computational burdens, complex formulations, and inability to consider a discrete time–cost relationship for completing an activity (Feng *et al.*, 1997; Hegazy, 1999).

Genetic algorithm (GA), an evolutionary technique, has been widely applied to solve the TCT problems. Based on the relationships between duration and cost, the TCT problem was transformed into a single objective optimization so as to apply GA (Li and Love, 1997; Hegazy, 1999; Li *et al.*, 1999). According to the weighting approach (Gen and Cheng, 2000), the two objectives were combined into a single objective by assigning weights to objectives and a modified adaptive weighting GA method was developed (Zheng *et al.*, 2004, 2005). Nevertheless, transformation to a single objective optimization could not minimize project duration and cost simultaneously. Though the modified adaptive weighting GA method could avoid some problems such as weighting sensitivity and a single optimum instead of a set of optimal solutions, a few non-dominated solutions could be identified (Zheng *et al.*, 2004). Feng *et al.* (1997) developed a multi-objective GA that could provide a set of optimal solutions for the TCT problem. The multi-objective GA is able to overcome the abovementioned deficiencies caused by other methods, such as the inability of finding the exact optima, heavy computation burden, difficulty in weighting objectives and determining only a few optima (Feng *et al.*, 1997).

Particle swarm optimization (PSO) developed by Kennedy and Eberhart (1995) is another kind of evolutionary technique, which simulates the social behaviour of bird flocking to a desired place. Similar to GAs, PSO starts with a set of initial solutions and repetitively updates them. In PSO each particle adjusts its flying according to the experience of its own and others. In addition to the similar advantages of GAs, including computational feasibility and effectiveness, PSO shows its uniqueness such as easy implementation and consistency in performance (Eberhart and Shi, 1998; Robinson *et al.*, 2002; Salman *et al.*, 2002). PSO has been utilized to solve the resource-constrained project scheduling (RCPS) problem (Zhang and Li, 2005; Zhang *et al.*, 2005). However, PSO has seldom been applied for solving the TCT problems.

In this paper, the mechanism of PSO is adopted to explore a new methodology to search for Pareto optimal solutions for the TCT problems. A combined scheme-based multi-objective particle swarm optimization (CSMOPSO) is proposed, in which sparse-degree and roulette-wheel selection are combined to determine the global best for guiding particle-flying. The candidate TCT solutions in terms of a set of construction methods associated with different duration and cost for all activities are represented using the multidimensional particles. A discrete time–cost relationship for completing each activity is considered. This study aims to propose an alternative solving methodology for the construction TCT problems by means of PSO, thus helping industry practitioners plan construction methods with optimal balance between time and cost.

Description of TCT problems

In a construction project, the duration of each activity is related to the adopted construction method composed of use policies of resources and construction technologies. An activity may require additional or more efficient resources or overtime to increase productivity or shorten completion time. For example, high-skilled manpower, more efficient concrete pump and even overtime may be adopted to speed up the progress of an activity of concreting, which should lead to additional cost. Further, the activity of concreting can be carried out using the following methods: (1) cast on site using a local batch plant, crane, bucket and crew; (2) cast on site using ready mix concrete transported from a nearby plant and pipe pumping; and (3) installing prefabricated elements transported from a plant. The three technologies also lead to different cost including direct cost for manpower, equipment and materials as well as indirect cost for transformation, installing and management.

Various construction methods correspond to different project duration and total cost. Normally, more productive construction methods are adopted to reduce duration while additional cost is required for more productive resources or technologies. However, there are some activities that by their very nature cannot be shortened even if additional cost is incurred. In any case, each relationship between time and cost actually reflects the advantage or disadvantage of the construction method to complete an activity. Because of the optional construction methods for an activity, it is assumed that there exists a discrete relationship between duration and cost for completing an activity. In this study, a set of construction methods corresponding to various extents of duration and cost for each activity is considered to be selected through the proposed multi-objective optimization methodology.

Assuming that the time and cost for various construction methods to complete an activity are known, the contractors or engineers need to select appropriate construction methods for all activities in a project with balanced time–cost tradeoff (TCT). The construction TCT problem is a multi-objective optimization with objectives of minimizing project duration and total cost. However, the magnitude of the search space for the multi-objective optimization makes it difficult or time-consuming for general practitioners to search for the optimal solution.

Concept of Pareto-oriented multi-objective optimization

In most of the multi-objective optimization problems, the multiple objectives may conflict with each other; for example, reduction in the project duration is achieved through additional cost for more productive resources or technologies. It is normally impossible to achieve one objective without sacrificing at least one or more objectives. The competing objectives have to be traded off against one another while seeking a solution. For instance, there exists tradeoff between project duration and cost when searching for the highest efficiency. The multi-objective optimization problem should not have a unique solution that makes all objective functions be the optimum at the same time. There exist a series of solutions that are non-dominated by each other. In other words, these solutions cannot be further optimized for one or several objectives and cannot be further worsened for other objectives. For example, a decision vector $\boldsymbol{X} = (x_1 \ldots, x_N)$ in the search space is said to be a non-dominated solution if there does not exist another decision vector $\boldsymbol{X}^* = (x^*_1 \ldots, x^*_N)$ that makes $f_i(^*\boldsymbol{X}) < f_i(\boldsymbol{X})$. Herein $f_i(\ )$ $(i = 1$ to $D)$ denotes one objective function and D denotes the number of objectives. All such non-dominated solutions make up Pareto front $\{\boldsymbol{S}\}$ of a multi-objective optimization problem (Coello, 1999). No member of the Pareto front is dominated by any other member. The goal of multi-objective optimization is to locate the Pareto front of the non-dominated solutions.

For the TCT problems, there are two objectives: minimization of project duration and minimization of cost. The objective functions $f_1(\boldsymbol{X})$ and $f_2(\boldsymbol{X})$ mean the mechanism to obtain the fitness (i.e. project duration and cost) for the TCT problems based on the decision vector $\boldsymbol{X} = (x_1 \ldots, x_N)$. Solving the TCT problem is to determine the Pareto front composed of a series of optimal solutions, i.e. an optimal set of construction methods for all activities.

The heuristic methods, the mathematical methods and the weighting methods that transform multiple objectives into a single objective could not provide a set of non-dominated solutions that make up the Pareto front for the multi-objective optimization problem. On the other hand, the evolutionary techniques such as GA and PSO that adopt the Pareto concept are able to determine the Pareto front composed of the non-dominated solutions (Coello, 1999). A multi-objective particle swarm optimization method for determining the Pareto front is applied in this paper.

Introduction to multi-objective particle swarm optimization (MOPSO)

With the successful application of PSO to single-objective optimization problems, multi-objective particle swarm optimization (MOPSO) has been developed to solve the multi-objective optimization problems.

Basic concept of PSO

PSO simulates a social behaviour such as bird flocking to a promising position for certain objectives in a multi-dimensional space (Kennedy and Eberhart, 1995; Eberhart and Shi, 2001). Like an evolutionary algorithm, PSO conducts a search using a population (called swarm) of individuals (called particles), the positions of which are updated from iteration to iteration. A particle is treated as a point in a multidimensional space and the status of a particle is characterized by its position and velocity (Kennedy and Eberhart, 1995). The position of each particle represents a candidate solution to the problem at hand, resembling the chromosome of GA. A swarm of particles flies along the trajectory that is updated based on the following formulas (Kennedy and Eberhart, 1995):

$$\boldsymbol{V}_i(t) = w(t)\boldsymbol{V}_i(t-1) + c_1 r_1(\boldsymbol{X}_i^L - \boldsymbol{X}_i(t-1)) + c_2 r_2 (\boldsymbol{X}^G - \boldsymbol{X}_i(t-1)) \tag{1}$$

$$\boldsymbol{X}_i(t) = \boldsymbol{V}_i(t) + \boldsymbol{X}_i(t-1) \tag{2}$$

where $\boldsymbol{X}_i(t) = \{x_{i1}(t), \ldots, x_{iN}(t)\}$ denotes the N-dimension position (a N-dimension vector) for particle i in iteration t, while $\boldsymbol{V}_i(t) = \{v_{i1}(t) \ldots, v_{iN}(t)\}$ denotes the velocity for particle i in iteration t. $i = 1, \ldots, P$ and P means the total number of the particles in a swarm, called population size; $t = 1, \ldots, T$ and T means the maximum iteration; $\boldsymbol{X}_i^L = \left\{x_{i1}^L, \ldots, x_{iN}^L\right\}$ denotes the individual best (*ibest*) position the particle i finds so far, while $\boldsymbol{X}^G = \left\{x_1^G, \ldots, x_N^G\right\}$ denotes the global best (*gbest*) position the swarm finds so far. c_1 and c_2 are the positive constants (namely learning factors), while r_1 and r_2 are random numbers between 0 and 1; $w(t)$ is the inertia weight for controlling the impact of the previous velocities on the current velocity.

Equation (1) determines a particle's new velocity according to its previous velocity and the distances from its current position to its *ibest* and the *gbest*. Equation (2) determines a particle's new position by utilizing its *ibest* and the *gbest*. Equations (1) and (2) also reflect the information-sharing mechanism. PSO shares some characteristics with a general evolutionary method such as GA, including population initialization, repetitive search process and fitness evaluating. The updating mechanism of PSO is analogous to the function of crossover for GA. However, PSO allows individuals to benefit from their past experiences whereas GA normally uses the current populations of survivals. In addition, the PSO updating mechanism can be formulated to facilitate the ease of implementation (Eberhart and Shi, 1998).

General multi-objective particle swarm optimization

Owing to the high speed of convergence, PSO has obtained success in solving single-objective optimization problems. PSO is also considered to be particularly suitable for the multi-objective optimization problems (Kennedy and Eberhart, 2001). Many attempts have been made to modify the single-objective PSO to suit multi-objective optimization problems.

The simplest category of approaches for multi-objective PSO is the aggregating methodology that assigns a weight to each objective and combines the multiple objectives into a single one (Parsopoulos and Vrahatis, 2002; Baumgartner *et al.*, 2004). In other words, the multi-objective PSO is transformed to the single PSO. However, the disadvantage of this approach is the lack of a mechanism for assigning the weights to reflect different levels of importance and scaling magnitude for various objectives.

Another category of approach for multi-objective PSO is the Pareto-oriented methodology that focuses on determination of the individual (*ibest*) and global best (*gbest*) based on the Pareto concept. The particle-flying for PSO is guided by the *ibest* for each particle as well as the *gbest* of the entire swarm. The *ibest* and *gbest* need to be determined repetitively during PSO. Because the optimal solution for multi-objective optimization is a set of non-dominated solutions instead of a single solution, determination of the *ibest* and *gbest* from the set of non-dominated solutions is crucial to the Pareto-oriented multi-objective PSO.

Determination of the *ibest* for each particle is straightforward. The *ibest* for particle i, denoted as $\boldsymbol{X}_i^L$, should be updated only when the new position of particle i is non-dominated and it dominates its previous *ibest* ($\boldsymbol{X}_i^L$). Determination of the *gbest* for each particle from the current Pareto front (i.e. the set of non-dominated solutions) is relatively complex. Considering that the *gbest* may be selected differently for each particle, the *gbest* should be denoted separately for each particle, such as $\boldsymbol{X}_{i_i}^G$ for particle i instead of $\boldsymbol{X}^G$ for the entire swarm in single-objective PSO. Various possible schemes have been proposed to determine the *gbest* from the set of non-dominated solutions. The most common schemes include random selection of *gbest* (Coello and Lechuga, 2002; Alvarez-Benitez *et al.*, 2005; Cagnina *et al.*, 2005) and information-based selection of *gbest* (Ray and Liew, 2002; Raquel and Naval, 2005; Sierra and Coello, 2005).

The random scheme gives all non-dominated solutions the same probability to be selected as the *gbest* for a particle. The random scheme is able to promote convergence so as to avoid being trapped at local optima (Coello *et al.*, 2004). On the other hand, the information-based scheme considers additional information such as density or crowing distance when determining the *gbest*. The information-based scheme is able to promote diversity of the solutions by attracting the swarm towards sparsely populated regions (Alvarez-Benitez *et al.*, 2005). The density, or alternatively called sparse-degree in this paper, in terms of the perimeter of the rectangle formed by the nearest neighbours as the vertices (see Figure 1) is often adopted; the non-dominated solution (e.g. $\boldsymbol{X}_k$) with a larger value of

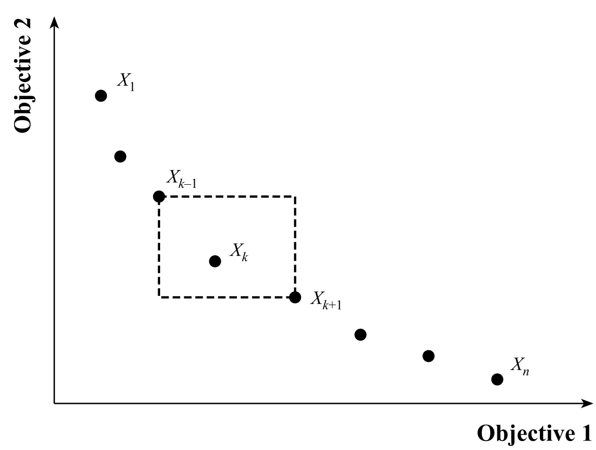

Figure 1 Sparse-degree in terms of perimeter of a rectangle

the sparse-degree has the priority to be selected as the *gbest*.

It is noted that the general multi-objective particle swarm optimization (MOPSO) methods individually adopt a random scheme or an information-based scheme to promote convergence or diversity (sparse-degree) respectively. The general MOPSO methods make it difficult to keep convergence and diversity simultaneously. So a combined scheme that takes advantage of the information-based scheme and the random scheme is expected to be developed.

A combined scheme to determine global best

The sparse-degree is herein considered to be measured in a straightforward manner, that is, in terms of the average Euclidean distance instead of perimeter (Figure 2). For instance, $d_{k,\,k-1}$ and $d_{k,\,k+1}$ respectively represent the Euclidean distance from the non-dominated solution $\boldsymbol{X}_k$ to the two adjacent neighbours (i.e. $\boldsymbol{X}_{k-1}$ and $\boldsymbol{X}_{k+1}$) on both sides. The average distance $d_k = 0.5\ (d_{k,\,k-1} + d_{k,\,k+1})$ serves as the sparse-degree around the solution $\boldsymbol{X}_k$. For the first and last solutions $\boldsymbol{X}_1$ and $\boldsymbol{X}_n$, the sparse-degree is just the distance from them to the only one adjacent solution. Therefore $d_1 = d_{1,2}$ and $d_n = d_{n,\,n-1}$. The distance is computed as follows:

$$d_{k,k-1} = \sqrt{(f_1(\boldsymbol{X}_k) - f_1(\boldsymbol{X}_{k-1}))^2 + (f_2(\boldsymbol{X}_k) - f_2(\boldsymbol{X}_{k-1}))^2}$$

where $f_1\ (\boldsymbol{X}_k)$ and $f_2\ (\boldsymbol{X}_k)$ respectively mean the fitness values for objective 1 (e.g. project duration) and objective 2 (e.g. cost) based on the solution $\boldsymbol{X}_k$.

Such a sparse-degree estimation can be used by the information-based scheme to determine the *gbest* from a set of non-dominated solutions, thus promoting diversity. In order to promote both convergence and diversity, a roulette-wheel selection is applied to determine the *gbest* by regarding the sparse-degree of each solution as the probability to be selected. Unlike the random selection method that randomly determines the *gbest*, the roulette-wheel selection determines the *gbest* based on the probability of each solution. The sparse-degree for each non-dominated solution should be normalized and the total sum of the normalized sparse-degree is equal to 1.0. The size of the section in the roulette wheel is proportional to the sparse-degree for every non-dominated solution; the bigger the sparse-degree is, the larger the section is, and the more likely the corresponding solution is selected as the *gbest*.

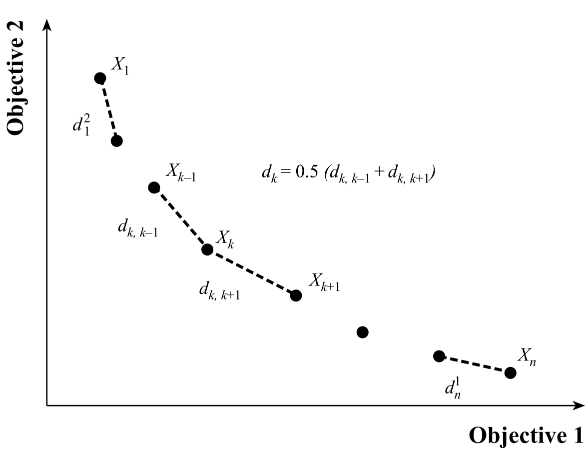

Figure 2 Sparse-degree in terms of Euclidean distance for a solution

Moreover, unlike some schemes considering parts of the non-dominated solutions to be recorded so as to reduce the time of scanning all solutions (Coello *et al.*, 2004), all non-dominated solutions are herein recorded and referred for determining the *gbest* in consideration of the property of the current computer. According to the sparse-degree estimation and the roulette-wheel selection, the procedure of the combined scheme to determine the *gbest* from a set of non-dominated solutions is presented as follows:

- For i =1 to P
  - Update the individual best (*ibest*) $\boldsymbol{X}_i^L$ for the particle i only when its new position is non-dominated and dominates the particle i's current individual best $\boldsymbol{X}_i^L$.
  - Update the Pareto front stored as $\{\boldsymbol{S}\}$ if the new *ibest* $\boldsymbol{X}_i^L$ for particle i is dominated by any existing optimal solutions in the Pareto front.
  - If $\boldsymbol{X}_i^L$ is non-dominated, then add the new non-dominated solution to the Pareto front $\{\boldsymbol{S}\}$.
  - Then find out the existing solutions that are dominated by this new solution and remove them from the stored Pareto front $\{\boldsymbol{S}\}$.
- End.
- For i =1 to P:
  - Determine the global best (*gbest*) $\boldsymbol{X}_{i_i^i}^G$ for particle i.
  - Navigate the stored Pareto front $\{\boldsymbol{S}\}$ for the optimal particle-represented solutions (i.e. positions), and then calculate the sparse-degree (i.e. d_k) for each optimal particle-represented solution.
  - Regarding the sparse-degree as the fitness of each particle-represented solution, select the global best $\boldsymbol{X}_{i_i^i}^G$ for the particle i according to the roulette-wheel selection method.
- End.

At each iteration, determination of the global best *gbest* for each particle will begin only after the *ibest* of every particle has been specified so that updating of the Pareto front is complete for the current iteration.

The proposed combined scheme combines the sparse-degree estimation and roulette-wheel selection to determine the *gbest* so as to promote both convergence and diversity for MOPSO. The advantages of the information-based scheme as well as the roulette-wheel selection that outperforms the random scheme are taken. In addition, all non-dominated solutions are to be considered when determining the *gbest*.

Combined scheme-based MOPSO (CSMOPSO) for TCT problems

Based on the combined scheme to determine the *gbest* from a set of non-dominated solutions (i.e. a Pareto front), a combined scheme-based MOPSO (CSMOPSO) for solving the TCT problems will be proposed in this section.

Mapping of TCT solutions to particles

A candidate solution to the TCT problem is a combination of methods for all activities, that is, a set of activities associated with construction methods, such as $\boldsymbol{X}_i = \{(1, m_1), (2, m_2), (3, m_3),\ldots, (j, m_j),\ldots, (N, m_N)\}$. Herein $j = 1$ to N represents one activity j among the N activities in the construction project, while m_j represents one method for activity j. The method m_j is one of the corresponding options in the range of $[1, M_j]$ and M_j is the maximum number of the method options.

As one point in a multidimensional space, each element of the multidimension position of the particle i at iteration t (i.e. $\boldsymbol{X}_i(t) = \{x_{i1}(t), x_{i2}(t),\ldots, x_{iN}(t)\}$) can stand for one activity among the N activities. Accordingly each element of the particle position can have the value that represents a method option for an activity. So a TCT solution in terms of a combination of activity-methods can be represented through a particle, with a matrix defining exactly its multidimension position $\boldsymbol{X}_i(t) = \{x_{i1}(t), \ldots, x_{iN}(t)\}$. Mapping between one TCT solution and the particle representation is described in Figure 3. The goal of solving the TCT problem, i.e. finding the optimal combination of activity-methods so as to minimize both project duration and cost, can be achieved through searching a particle position with a fitness of optimal time–cost tradeoff. The fitness for the two objectives, i.e. project duration and total cost, is obtained from the scheduling (rather than objective functions $f_1(\boldsymbol{X}_i)$ and $f_2(\boldsymbol{X}_i)$ by adopting the particle-represented solution (i.e. a set of activity-methods) and the precedence relationships among activities.

The multidimension velocity of such a particle can be denoted as $\boldsymbol{V}_i(t) = \{v_{i1}(t), \ldots, v_{iN}(t)\}$. The individual best *ibest* for particle i is represented as $\boldsymbol{X}_i^L = \left\{x_{i1}^L,\ldots,x_{iN}^L\right\}$, while the global best *gbest* that may be different for each particle is represented as $\boldsymbol{X}_i^G = \left\{x_{i1}^G,\ldots,x_{iN}^G\right\}$. Considering the method option range for each activity, the search constraint (i.e. $[x_j^{\min}, x_j^{\max}]\,(j=1 \text{ to } N)$) of the particle-represented solution $X_i(t)$ corresponds to the method option range $[1, M_j]$ ($j = 1$ to N), that is, $x_j^{\min} = 1$ and $x_j^{\max} = M_j$. The particle-represented solutions outside $[1, M_j]$ are infeasible, so the particles that fly outside the search constraint should be adjusted as follows:

For $j = 1$ to N
if $x_{ij}(t) > M_j$ then $x_{ij}(t) = M_j$
else if $x_{ij}(t) < 1$ then $x_{ij}(t) = 1$
End

The velocity of a particle, i.e. $\boldsymbol{v}_i(t) = \{v_{i1}(t), \ldots, v_{iN}(t)\}$, should be also subject to a limit $[v_j^{\min}, v_j^{\max}]\,(j=1 \text{ to } N)$ so as to prevent explosion (Clerc and Kennedy, 2002). Based on the search constraint $[x_j^{\min}, x_j^{\max}]\,(j=1 \text{ to } N)$ for the particle-represented solutions, the velocity constraint can be defined as:

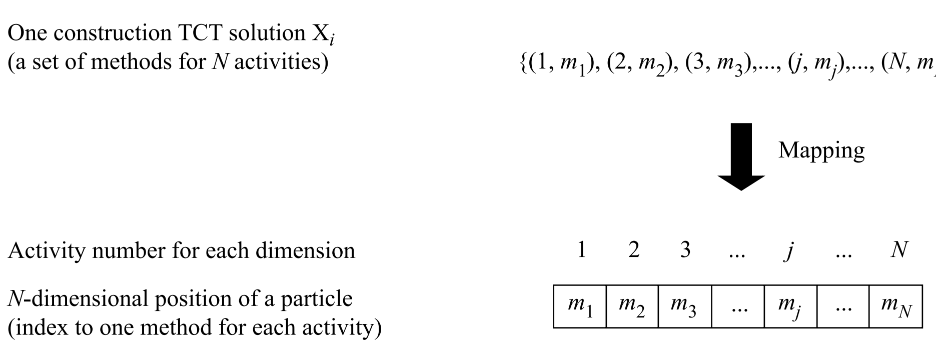

Figure 3 Particle-represented solution for the TCT problem

$v_j^{\min} = -x_j^{\min}$ and $v_j^{\max} = x_j^{\max}$. So the particle velocity should be subject to the constraint $[-M_j, M_j]$ ($j = 1$ to N). The particle velocity that is beyond this constraint should be adjusted as follows:

For $j = 1$ to N
if $v_{ij}(t) > M_j$ then $v_{ij}(t) = M_j$
else if $v_{ij}(t) < -M_j$ then $v_{ij}(t) = -M_j$
End

Flowchart of CSMOPSO

According to the mapping between the TCT solutions and the particles as well as the combined scheme-based multi-objective PSO, the flowchart of the CSMOPSO is presented in Figure 4, which can help implement the CSMOPSO.

Please note that each element (i.e. $x_{ij}(t)$, $I = 1$ to P and $j = 1$ to N) of the particle-represented TCT solution (i.e. $\boldsymbol{X}_i(t)$) should be rounded off to an integer to reflect the method index for each activity. P denotes the population size, that is, the number of particles in the swarm. N denotes the total number of activities of the construction project under study. t denotes the index number of the iteration. M_j ($j = 1$ to N) means the maximum number of the method options for activity j. Updating the particle velocity $\boldsymbol{V}_i(t)$ and position $\boldsymbol{X}_i(t)$ is achieved through Equations (1) and (2). The particle velocity $\boldsymbol{V}_i(t)$ and position $\boldsymbol{X}_i(t)$ that are beyond the corresponding constraints should be adjusted based on the abovementioned method. Determination of the individual best (*ibest*) and global best (*gbest*) for each particle from a set of non-dominated solutions is achieved through the proposed combined scheme. The 'stop criteria' include: (1) maximum number of iterations since last updating of the non-dominated solutions; and (2) maximum total number of iterations. According to the flowchart, the proposed CSMOPSO has been implemented in Visual C++. No additional information for the prototype system is provided here in the consideration of the length limitation of the paper.

Computational experiments

In order to verify the performance of the proposed CSMOPSO, computational experiments as well as comparisons with a random selection-based multi-objective PSO (RSMOPSO) and a multi-objective genetic algorithm (MOGA) are carried out. The RSMOPSO resembles the proposed CSMOPSO except that the global best (*gbest*) is randomly selected. The MOGA is based on the methodology of Feng *et al.* (1997). The RSMOPSO and MOGA are also implemented in Visual C++ just for the purpose of comparisons.

Example description

To verify the performance of the proposed CSMOPSO, exact multi-objective optimal solutions (i.e. true Pareto front) for the examples are required. Therefore, two examples that have been used by Burns *et al.* (1996) and Feng *et al.* (1997) and have the exact solutions are adopted.

Example 1 has seven activities (Figure 5) and the optional methods associated with various durations and costs are shown in Table 1. Example 2 has 18 activities (Figure 6) and the optional methods associated with various durations and costs are shown in Table 2.

Some parameters for experimenting MSMOPSO need to be specified. The inertia weight $w(t)$ is given a constant value of 0.8 according to the conclusion (Eberhart and Shi, 1998) that 0.8 is suitable to the case $v_j^{\max} > 3$ and the current situation that the option range is normally more than 3 so that $v_j^{\max} (= x_j^{\max}) > 3$. Considering that different learning factors c_1 and c_2 cause a little bit of difference (Trelea, 2003), a value of 1.0 is assigned as usual. The population size (i.e. the number of particles in the swarm) P is usually taken as the value close to the number of non-dummy activities (i.e. N) in the project under study, considering that more particles may increase searching success but similarly require more evaluation runs (Trelea, 2003). Therefore the P is given the values of 10 and 20 respectively for the two examples. The termination signals, i.e. maximum number of iterations since last updating of the non-dominated solutions and maximum total number of iterations, are given the values of 50 and 400 respectively for the two examples.

The parameters for the RSMOPSO are the same as those for the CSMOPSO. The parameters for the MOGA are specified as follows: the population size, crossover probability, and mutation probability are given the values of 100, 0.4 and 0.02, respectively. The termination signals are also given the same values as those for the CSMOPSO. Fifty runs of experiments for each method and each example are conducted so as to average the qualitative results.

Performance analysis

In order to investigate the performance of a multi-objective optimization algorithm, three quantitative metrics are usually used (Zitzler *et al.*, 2000). The three metrics respectively assess the convergence degree, diversity (sparse-degree of the non-dominated solutions) and speed of convergence.

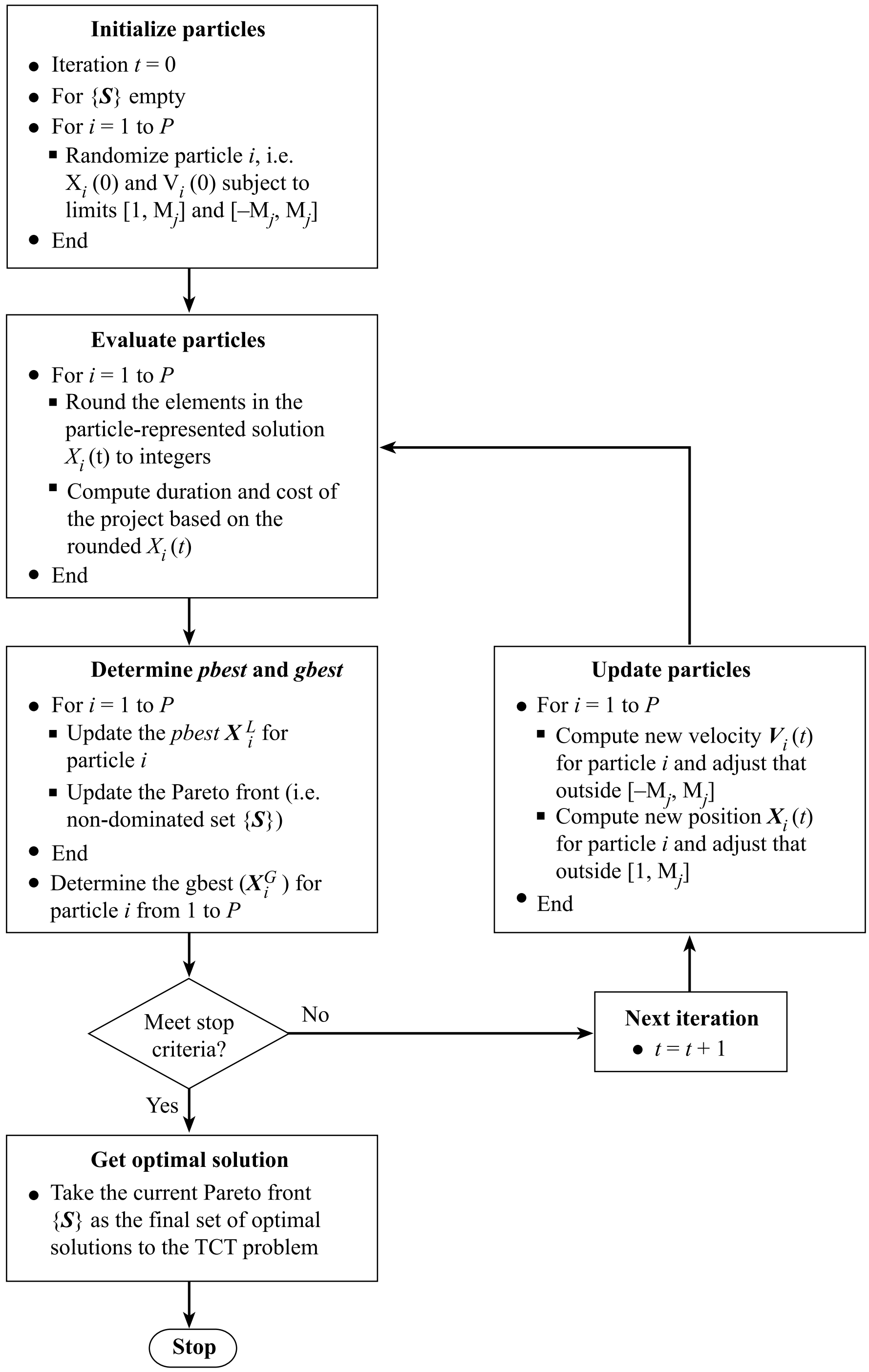

Figure 4 Flowchart of the proposed CSMOPSO

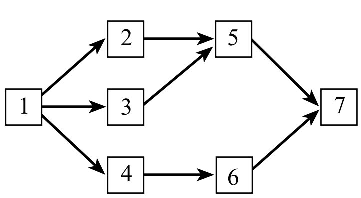

Figure 5 Activity-on-node network for example 1

The first metric considers the generational distance (GD) (Zitzler *et al.*, 2000) and is defined as:

$$GD = \frac{1}{n}\sqrt{\sum_{i=1}^{n} d_i^2} \tag{3}$$

where n is the number of solutions in the obtained Pareto front and d_i is the Euclidean distance (in the

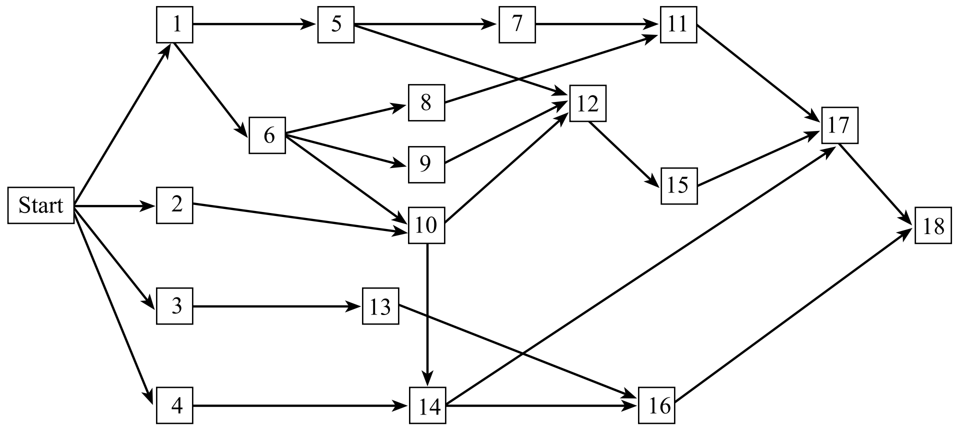

Figure 6 Activity-on-node network for example 2

Table 1 Various options for example 1

Activity	Option	Duration (days)	Cost ($)
1	Method 1	14	23 000
	Method 2	20	18 000
	Method 3	24	12 000
2	Method 1	15	3000
	Method 2	18	2400
	Method 3	20	1800
	Method 4	23	1500
	Method 5	25	1000
3	Method 1	15	4500
	Method 2	22	4000
	Method 3	33	3200
4	Method 1	12	45 000
	Method 2	16	35 000
	Method 3	20	30 000
5	Method 1	22	20 000
	Method 2	24	17 500
	Method 3	28	15 000
	Method 4	30	10 000
6	Method 1	14	40 000
	Method 2	18	32 000
	Method 3	14	18 000
7	Method 1	9	30 000
	Method 2	15	24 000
	Method 3	18	22 000

two-objective space) between solution i in the obtained Pareto front and the nearest solution in the true Pareto front. The generational distance (GD) measures the convergence degree. A smaller value indicates a higher convergence degree, while a zero value means complete convergence.

The second metric considers the range variance (RV) of neighbouring solutions (Zitzler *et al.*, 2000) and is defined as:

$$RV = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d} - d_i)^2} \quad (4)$$

where $d_i = \min_j (|f_1(X^i) - f_1(X^j)| + |f_2(X^i) - f_2(X^j)|)$; $i, j = 1, \ldots, n$, $\bar{d}$ is the mean of all d_i, $f_1(X^i)$ and $f_2(X^i)$ are respectively the fitness values of the first objective (i.e. project duration) and is the second objective (i.e. total cost) for the solution i. The range variance (RV) measures diversity of the non-dominated solutions. A smaller value indicates a higher diversity degree and a value of zero indicates that all the non-dominated solutions are uniformly distributed.

The third metric is the computation time required to get the final results by the CSMOPSO. A shorter computation time indicates a faster convergence process. It should be noted that comparison of computation time must be under the condition that these multi-objective optimization systems be executed within the same hardware environment.

The average values of the generational distance (GD), range variance (RV) and computation time (CT) based on 50 runs for each case and the corresponding standard deviations (Std.dev) are listed in Table 3. It is shown that the proposed CSMOPSO has the smallest values of GD among the three methods and obviously much smaller GD than those of the RSMOPSO, meaning that the CSMOPSO has the highest convergence degree among the three methods. Table 3 shows that the CSMOPSO results in smaller values of RV than those of the RSMOPSO and almost the same as those of the MOGA, meaning that the CSMOPSO outperforms the RSMOPSO with regard to diversity. Table 3 also shows that the CSMOPSO spends less computation time (CT) than the RSMOPSO and MOGA, meaning that the CSMOPSO has the fastest speed of convergence among the three methods.

Table 2 Various options for example 2

Activity	Option	Duration (days)	Cost ($)
1	Method 1	14	2400
	Method 2	15	2150
	Method 3	16	1900
	Method 4	21	1500
	Method 5	24	1200
2	Method 1	15	3000
	Method 2	18	2400
	Method 3	20	1800
	Method 4	23	1500
	Method 5	25	1000
3	Method 1	15	4500
	Method 2	22	4000
	Method 3	33	3200
4	Method 1	12	45 000
	Method 2	16	35 000
	Method 3	20	30 000
5	Method 1	22	20 000
	Method 2	24	17 500
	Method 3	28	15 000
	Method 4	30	10 000
6	Method 1	14	40 000
	Method 2	18	32 000
	Method 3	24	18 000
7	Method 1	9	30 000
	Method 2	15	24 000
	Method 3	18	22 000
8	Method 1	14	220
	Method 2	15	215
	Method 3	16	200
	Method 4	21	208
	Method 5	24	120
9	Method 1	15	300
	Method 2	18	240
	Method 3	20	180
	Method 4	23	150
	Method 5	25	100
10	Method 1	15	450
	Method 2	22	400
	Method 3	33	320
11	Method 1	12	450
	Method 2	16	350
	Method 3	20	300
12	Method 1	22	2000
	Method 2	24	1750
	Method 3	28	1500
	Method 4	30	1000
13	Method 1	14	4000
	Method 2	18	3200
	Method 3	24	1800
14	Method 1	9	3000
	Method 2	15	2400
	Method 3	18	2200
15	Method 1	12	4500
	Method 2	16	3500
16	Method 1	20	3000
	Method 2	22	2000
	Method 3	24	1750
	Method 4	28	1500
	Method 5	30	1000
17	Method 1	14	4000
	Method 2	18	3200
	Method 3	24	1800
18	Method 1	9	3000
	Method 2	15	2400
	Method 3	18	2200

In summary, the above analyses demonstrate that the CSMOPSO performs the best among the three methods, particularly much better than the RSMOPSO. These demonstrated performances actually reflect the respective advantages of the information-based scheme and the roulette-wheel selection superior to the random scheme, which are combined by the CSMOPSO to determine the *gbest* during the searching process.

Use of obtained solutions

The final results the CSMOPSO provides for the two examples are two series of non-dominated solutions (i.e. the Pareto fronts) shown in Figure 7 and Figure 8, each solution corresponding to a selection of construction methods for each activity in the two projects. The series of non-dominated solutions reflect a discrete and balanced relationship between time (project duration) and cost (total cost) to complete the projects. According to the series of non-dominated solutions, practitioners (e.g. contractors or engineers) are able to select a set of construction methods for each activity in the construction projects based on the actual situations such as duration requirement and investment limitation.

For instance, if the practitioners accept the result with project duration 78 and total cost 107 500, one of the series of non-dominated solutions (Figure 7) for example 1, then the selection of construction methods for the seven activities can be determined as shown in Table 4. If the result with project duration 110 and total cost 106 270 for example 2 is desirable among the series of non-dominated solutions (Figure 8), then the practitioners can select the construction methods for the 18 activities as shown in Table 5.

It should be noted that the cost within a TCT solution may be just the direct cost rather than the total cost for completing the construction project if the

Table 3 Metric value for different cases

Methods			CSMOPSO	RSMOPSO	MOGA
Example 1	Generational distance (GD)	Average	2.856	5.483	2.939
		Std.dev	0.974	1.724	0.992
	Range variance (RV)	Average	6.538	10.472	6.423
		Std.dev	1.352	1.767	1.377
	Computation time (minutes)	Average	1.061	2.428	1.436
		Std.dev	0.0503	0.0416	0.0408
Example 2	Generational distance (GD)	Average	4.725	9.792	5.684
		Std.dev	1.556	3.663	1.567
	Range variance (RV)	Average	9.581	14.247	10.156
		Std.dev	1.947	2.769	2.552
	Computation time (minutes)	Average	3.418	5.564	4.427
		Std.dev	0.0856	0.0776	0.0861

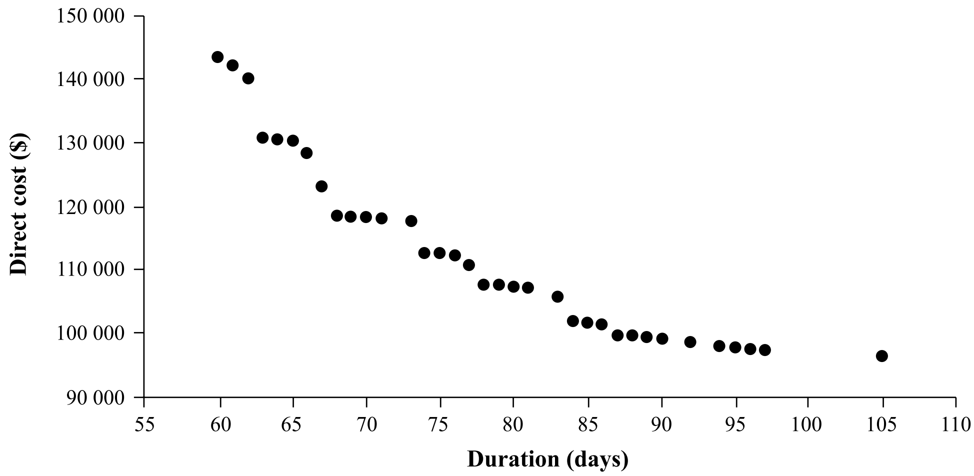

Figure 7 Pareto front (a set of non-dominated solutions) for example 1

Table 4 One selection of construction methods for example 1

Activity index	Construction method index	Time (days)	Cost ($)
1	3	24	12 000
2	1	15	3000
3	1	15	4500
4	3	20	30 000
5	4	30	10 000
6	3	24	18 000
7	1	9	30 000

Project duration: 78 days; Total cost: $107 500

inputted cost for each activity contains just the direct cost. Then the total cost can be obtained by summing up the direct cost and the indirect that is usually assumed to be proportional to the project duration. The availability of a set of non-dominated solutions is enough and convenient to consider various indirect cost rates without experimenting with multi-objective optimization.

Further study

To utilize the proposed CSMOPSO for solving the construction TCT problems requires the data of time and cost associated with various construction methods for various construction activities. Therefore it is necessary to provide contractors or engineers with a convenient means to obtain these data. The study on such issues is being carried out and is simply introduced herein.

Collection of data

The time and cost associated with various construction methods for an activity are collected by means of practical investigation, experts' investigation and referring

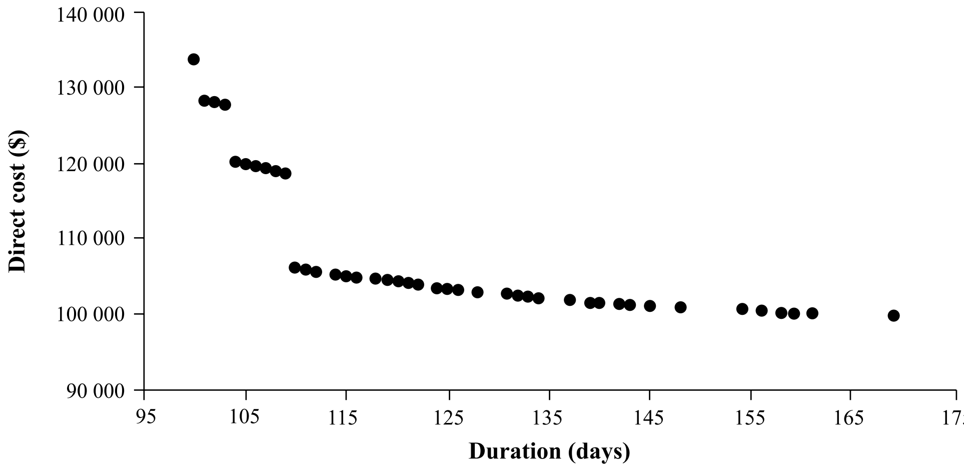

Figure 8 Pareto front (a set of non-dominated solutions) for example 2

Table 5 One selection of construction methods for example 2

Activity index	Construction method index	Time (days)	Cost ($)
1	1	14	2400
2	5	25	1000
3	3	33	3200
4	3	20	30 000
5	4	30	10 000
6	3	24	18 000
7	3	18	22 000
8	5	24	120
9	1	15	300
10	1	15	450
11	3	20	300
12	1	22	2000
13	3	24	1800
14	3	18	2200
15	1	12	4500
16	5	30	1000
17	1	14	4000
18	1	9	3000

Project duration: 110 days; Total cost: $106 270

to historical records. The activities should include not only those in building construction but also those for fundamental projects (road, bridge and tunnel construction). Various activities may consider different construction methods according to their characteristics. Some may consider only different assignments of equipment and manpower; some may consider overtime and different construction technologies. For example, activity excavation may consider various construction methods consisting of different equipment (e.g. excavator); activity girder erection may consider construction methods with combinations of different cranes and crew sizes; activity concreting may consider construction methods with various assignments of equipment (e.g. concrete pump) and technologies (e.g. cast on site and installing prefabricated). It is necessary to consider as many options of construction methods as possible. In addition, the volume or amount of work that each activity involves also determines the actually required time and cost. Therefore, information relating to not only the time and cost associated with various construction methods but also the volume or amount of work the investigated activity actually involves should be collected.

Platform to handle collected data

Once the relevant data are collected, a computer platform is required to help users handle the collected data and provide the parameters suitable for the current project under study. This platform should have the capability to input and store the collected data through practical cases, experts and historical records. In addition, this platform should be able to trace the collected data and find out the suitable parameters according to the actual projects. Moreover, some calculations are to be operated to get the actual parameters based on the time–cost relationship for a unit-volume work of activity and the actual volume the current activity involves. Finally, this platform should be user-friendly and extendable for further development and utilization.

Conclusions

A multi-objective particle swarm optimization (MOPSO) methodology based on a combined scheme for determining the global best (*gbest*) has been proposed for solving the construction time–cost tradeoff problem. In consideration of Pareto-oriented optimization and the need of diversity and convergence, a scheme that combines the sparse-degree and roulette-wheel selection is developed to determine the *gbest* from a set of non-dominated solutions (i.e. a Pareto front). The framework of the combined scheme-based multi-objective particle swarm optimization (CSMOPSO) has been presented, including mapping between the candidate TCT solutions and the multidimension particles as well as the flowchart of the CSMOPSO. The proposed method has been investigated through computational analyses based on two illustrative examples and comparisons with another multi-objective PSO (RSMOPSO) and a multi-objective GA (MOGA). The computational analyses demonstrate that the CSMOPSO outperforms the RSMOPSO and MOGA with regard to convergence degree, diversity and speed of convergence.

The research provides an alternative PSO-based methodology to solve the construction TCT problems, that is, determining an optimal set of construction methods for all activities which corresponds to the minimum project duration and cost. The proposed CSMOPSO is expected to assist researchers or practitioners in effectively analysing and planning construction projects. Further studies can be directed towards development of a platform to handle the collected data, considering the uncertain activity duration and cost as well as adding more objectives such as construction quality to the optimization process.

References

Alvarez-Benitez, J.E., Everson, R.M. and Fieldsend, J.E. (2005) A MOPSO algorithm based exclusively on Pareto dominance concepts, in *Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005,* Guanajuato, Mexico, March, pp. 459–73.

Baumgartner, U., Magele, C. and Renhart, W. (2004) Pareto optimality and particle swarm optimization. *IEEE Transactions on Magnetics,* **40**(2), 1172–5.

Burns, S., Liu, L. and Feng, C. (1996) The LP/IP hybrid method for construction time-cost trade-off analysis. *Construction Management and Economics,* **14**, 265–76.

Cagnina, L., Esquivel, S. and Coello Coello, C.A. (2005) A particle swarm optimizer for multi-objective optimization. *Journal of Computer Science & Technology,* **4**(5), 204–10.

Clerc, M. and Kennedy, J. (2002) The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation,* **6**(1), 58–73.

Coello, C.A.C. (1999) A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems,* **1**(3), 269–308.

Coello, C.A.C. and Lechuga, M.S. (2002) MOPSO: a proposal for multiple objective particle swarm optimization, in *Proceedings of the IEEE World Congress on Evolutionary Computation (CEC 2002),* Hawaii, 12–17 May, pp. 1051–6.

Coello, C.A.C., Pulido, G.T. and Lechuga, M.S. (2004) Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation,* **8**(3), 256–79.

De, P., Dunne, E.J., Gosh, J.B. and Wells, C.E. (1995) The discrete time-cost tradeoff problem revisited. *European Journal of Operational Research,* **81**(2), 225–38.

Eberhart, R.C. and Shi, Y. (1998) Comparison between genetic algorithms and particle swarm optimization, in *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming,* San Diego, CA, pp. 611–6.

Eberhart, R.C. and Shi, Y. (2001) Tracking and optimizing dynamic systems with particle swarms, in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2001),* Seoul, Korea, pp. 94–7.

Elmaghraby, S.E. (1993) Resource allocation via dynamic programming in activity networks. *European Journal of Operational Research,* **64**(22), 199–215.

Elmaghraby, S.E. (1995) Resource allocation via dynamic programming in activity networks. *European Journal of Operational Research,* **64**, 199–215.

Feng, C.-W., Liu, L. and Burns, S.A. (1997) Using genetic algorithms to solve construction time-cost trade-off problems. *ASCE Journal of Computing in Civil Engineering,* **11**(3), 184–9.

Gen, M. and Cheng, R. (2000) *Genetic Algorithms and Engineering Optimization,* Wiley-Interscience, New York.

Hegazy, T. (1999) Optimization of construction time-cost trade-off analysis using genetic algorithms. *Canadian Journal of Civil Engineering,* **26**, 685–97.

Hendrickson, C. and Au, T. (1989) *Project Management for Construction,* Prentice Hall, Inc., Englewood Cliffs, NJ.

Kelly J. (1961) Critical path planning and scheduling: mathematical basis. *Operations Research,* **9**(3), 296–320.

Kennedy, J. and Eberhart, R.C. (1995) Particle swarm optimization, in *Proceedings of the Fourth IEEE Conference on Neural Networks,* Perth, Australia, pp. 1942–8.

Li, H. and Love, P. (1997) Using improved genetic algorithms to facilitate time-cost optimization. *ASCE Journal of Construction Engineering and Management,* **123**(3), 233–7.

Li, H., Cao, J. and Love, P. (1999) Using machine learning and GA to solve time-cost trade-off problems. *ASCE Journal of Construction Engineering and Management,* **125**(5), 347–53.

Meyer, W.L. and Shaffer, L.R. (1963) Extensions of the critical path method through the application of integer programming. Civil Engineering Construction Research Series, No. 2, University of Illinois, Urbana-Champaign, IL.

Moselhi, O. (1993) Schedule compression using the direct stiffness method. *Canadian Journal of Civil Engineering,* **20**, 65–72.

Pagnoni, A. (1990) *Project Engineering: Computer-Oriented Planning and Operational Decision Making,* Springer-Verlag, Berlin/Heidelberg/New York.

Parsopoulos, K.E. and Vrahatis, M.N. (2002) Particle swarm optimization method in multiobjective problems, in

Proceedings of the 2002 ACM Symposium on Applied Computing, ACM Press, Madrid, Spain, 11–14 March, pp. 603–7.

Patterson, J.H. and Huber, D. (1974) A horizon-varying, zero-one approach to project scheduling. *Management Science,* **24**(6), 990–8.

Prager, W. (1963) A structural method of computing project cost polygons. *Management Science,* **9**(3), 394–404.

Raquel, C.R. and Naval, P.C. (2005) An effective use of crowding distance in multiobjective particle swarm optimization, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005),* ACM Press, Washington, DC, 25–26 June, pp. 257–64.

Ray, T. and Liew, K.M. (2002) A swarm metaphor for multiobjective design optimization. *Engineering Optimization,* **34**(2), 141–53.

Robinson, D.R. (1975) A dynamic programming solution to cost-time tradeoff for CPM. *Management Science,* **22**(2), 158–66.

Robinson, J., Sinton, S. and Rahmat-Samii, Y. (2002) Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna, in *IEEE Antennas and Propagation Society International Symposium and URSI National Radio Science Meeting,* San Antonio, TX, June, pp. 168–75.

Salman, A., Ahmad, I. and Al-Madani, S. (2002) Particle swarm optimization for task assignment problem. *Microprocessors and Microsystems,* **26**(8), 363–71.

Siemens, N. (1971) A simple CPM time-cost tradeoff algorithm. *Management Science,* **18**(3B), 354–63.

Sierra, M.R. and Coello, C.A.C. (2005) Improving PSO-based multi-objective optimization using crowding, mutation and ϵ-dominance, in *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization,* EMO 25, Guanajuato, Mexico, 9–11 March, pp. 505–19.

Trelea, I.C. (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters,* **85**(6), 317–25.

Zhang, H. and Li, H. (2005) Particle swarm optimization-based schemes for resource-constrained project scheduling. *Automation in Construction,* **14**(3), 393–404.

Zhang, H., Li, H. and Tam, C.M. (2005) Particle swarm optimization for preemptive scheduling under break and resource-constraints. *ASCE Journal of Construction Engineering and Management,* **132**(3), 259–67.

Zheng, D.X.M., Ng, S.T. and Kumaraswamy, M. (2004) Applying a genetic algorithm-based multiobjective approach for time-cost optimization. *ASCE Journal of Construction Engineering and Management,* **130**(2), 168–76.

Zheng, D.X.M., Ng, S.T. and Kumaraswamy, M. (2005) Applying Pareto ranking and niche formation to genetic algorithm-based multiobjective time-cost optimization. *Journal of Construction Engineering and Management,* **131**(11), 81–91.

Zitzler, E., Deb, K. and Thiele, L. (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation,* **8**(2), 173–95.