

IMPACTS OF DIFFERENT OBJECTIVE FUNCTIONS ON RESOURCE LEVELING IN CONSTRUCTION PROJECTS: A CASE STUDY

Atilla DAMCI, Gul POLAT

Department of Civil Engineering, Istanbul Technical University, Istanbul, Turkey

Received 23 Aug 2011; accepted 13 Apr 2012

Abstract. A review of the recent literature on the models that focus on resource leveling in Critical Path Method networks shows that different objective functions have been used to optimize resource utilization. The main objective of this study is to investigate the impacts of using different objective functions on resource utilization histograms in Critical Path Method networks. For this purpose, nine different resource leveling objective functions were optimized via a genetic algorithm-based model. The model was developed using actual data obtained from a steel framed industrial building construction project. It was found that each of these objective functions generates different resource utilization histograms. In order to determine the improvement levels achieved by resource leveling using nine different objective functions, the improvement percentage in each parameter and the average improvement percentage for each objective function were calculated. Even though the objective function that involves the minimization of the sum of the square of the deviations in daily resource usage provided the best average improvement percentage in the studied case, another objective function(s) may provide better average improvement percentage in different projects. The contractor should consider all objective functions for resource leveling and select the one(s) that provides the best average improvement percentage.

Keywords: scheduling, critical path method, resource levelling, genetic algorithm.

Reference to this paper should be made as follows: Damci, A.; Polat, G. 2014. Impacts of different objective functions on resource leveling in construction projects: a case study, *Journal of Civil Engineering and Management* 20(4): 537–547. <http://dx.doi.org/10.3846/13923730.2013.801909>

Introduction

There are several scheduling techniques, such as Gantt charts, Critical Path Method (CPM), and Program Evaluation and Review Technique (Pert) used in construction projects. Among these techniques, CPM has been widely used by construction practitioners since the 1950s (Leu, Yang 1999a; Leu *et al.* 2000; Galloway 2006; Rogalska, Hejducki 2007; Lu *et al.* 2008; Hariga, El-Sayegh 2010). Despite its popularity, CPM has a major shortcoming as it assumes that resources are unlimited (Senouci, Adeli 2001; Chen, Weng 2009; Hariga, El-Sayegh 2010). However, in real life, resources are limited and several activities may concurrently require these limited resources, which in turn may lead to shortages and idleness. Such interruptions in production may bring about severe delays as well as cost overruns resulting from shortage costs, waiting costs, penalties, etc. Therefore, resources should be utilized efficiently in order to prevent project failures. Several studies focus on resource management in CPM networks (Hegazy 1999; Hegazy, Kassab 2003; Hinze 2004; Hariga, El-Sayegh 2010).

Studies on resource management in CPM networks in construction projects can be categorized according to two approaches. The first approach, which is known as resource allocation or resource-constrained scheduling,

assumes that there are limitations on resources. The main objective of this approach is to minimize project duration given constraints on resources (Senouci, Adeli 2001).

The second approach, which is known as resource leveling (often referred to as resource smoothing), assumes that there are sufficient resources available; however, the project duration is limited (e.g. Hegazy 1999; Doulabi *et al.* 2011). The goal of resource leveling is to minimize fluctuations, peaks and valleys in resource utilization without changing the completion time of a project and the number of resources required (Son, Skibniewski 1999; Leu *et al.* 2000; Hegazy, Kassab 2003; Senouci, Eldin 2004; Doulabi *et al.* 2011; Hariga, El-Sayegh 2010). The main idea behind resource leveling is shifting the start times of the non-critical activities along their available float times in order to level resource utilization (Lu, Li 2003; Lu *et al.* 2008).

Resource leveling can be considered as an optimization problem. Optimization is a mathematical discipline that focuses on finding the minima or maxima of an objective function, subject to constraints. In this regard, resource leveling can be achieved as long as the objective function and constraints are correctly defined. In a typical resource leveling problem, constraints are the float times of the non-critical activities that are derived from CPM calculations and precedence relationships that are consid-

ered during the construction of the CPM network, whereas there is no consensus on which objective function should be selected. A review of the literature on resource leveling in construction projects indicates that nine objective functions have been used, whose details will be presented in the succeeding sections. Although the main idea behind all these objective functions is very similar, namely achieving an even flow of resources while the original project duration remains unchanged, each of them aims to optimize different parameters and therefore generates diverse resource utilization histograms (Popescu 1976; Mattila, Abraham 1998). Therefore, a contractor should be aware of the possible outcomes of using any of these objective functions and carefully select the one that yields the resource utilization histogram that fulfills the special requirements of the project.

The main objective of this study is to investigate the impacts of using different objective functions in leveling resources in CPM networks. For this purpose, a genetic algorithm-based model was developed and tested using actual data obtained from a steel-framed industrial building project. The constraints of the model were the free float times of the non-critical activities derived from the CPM calculations. Nine different objective functions were optimized via the genetic algorithm-based model. It was found that each of these objective functions generates different resource utilization histograms.

1. Research methodology

A review of the literature indicates that there are several studies focusing on resource leveling in CPM networks. However, the impacts of using different objective functions in optimizing resource utilization have not been investigated. The main objective of this study is to investigate the impacts of using different objective functions on resource utilization histograms in CPM networks. For this purpose, the following tasks were performed:

1. Reviewing the literature on resource leveling in CPM networks in order to determine the objective functions that are used in optimizing resource utilization;
2. Obtaining relevant data from a real-life steel framed industrial building construction project in order to demonstrate the impacts of using different objective functions on resource histograms;
3. Determining the total project duration, the critical path of the project, and activity floats of the real-life construction project through basic CPM scheduling calculations using MS Project;
4. Developing a genetic algorithm based model for resource leveling of schedules established through CPM by using the actual data obtained from the real-life steel framed industrial building construction project using Evolver;
5. Generating resource utilization histograms for nine different objective functions by using the genetic algorithm-based model.

2. Literature review

In past years, many studies have been conducted to develop models for resource leveling in CPM networks. The methods used in these studies can be classified into three main groups, namely: (1) exact algorithms; (2) heuristic methods; and (3) metaheuristics (Easa 1989; Harris 1990; Son, Skibniewski 1999; Leu *et al.* 2000; Jaskowski, Sobotka 2004; Senouci, Eldin 2004; Hariga, El-Sayegh 2010). Each group has its own advantages and disadvantages. In exact algorithms, the solution converges to the optimum iteratively or in a finite number of steps (e.g. linear programming, branch and bound algorithms). These algorithms can be useful for finding the optimum solution on small-scale problems; however, they can be inefficient on large-scale problems, because they may consume too much time in order to find the optimal solution. Heuristic methods (e.g. using a rule of thumb) can considerably speed up the process of finding a solution on large-scale problems, but they do not guarantee an optimal solution. Exact algorithms and heuristic methods may fail to solve complex optimization problems. Metaheuristics (e.g. genetic algorithms, tabu search, simulated annealing) can be used in order to solve complex optimization problems (Leu *et al.* 2000; Osman, Laporte 1996; Senouci, Eldin 2004; Taha 1995). Osman and Laporte (1996) defined metaheuristics “as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions”. Resource leveling in real-life CPM networks is large-scale and complex problems since there are a great number of activities and dependencies among these activities. Therefore, it is commonly acknowledged that heuristic methods or metaheuristics may simply and fast solve resource leveling problem in CPM networks, but may not give the best solution in all cases (Leu *et al.* 2000; Senouci, Eldin 2004).

One of the earliest attempts to solve the resource leveling problem was proposed by Burgess and Killebrew (1962), whose objective was to minimize the sum of the squares of the resource usage in order to have a resource histogram in the shape of a rectangle. Wagner *et al.* (1964), Ahuja (1976), Popescu (1976) and Wiest and Levy (1977) discussed alternative objective functions. Harris (1978) proposed the minimum moment algorithm, which minimizes the moment of the resource histogram around the horizontal axis. In spite of the fact that the minimum moment algorithm is one of the most commonly used resource leveling method, it has drawbacks, in that no consideration is given to activity stretching (compressing), activities cannot be interrupted, and resource assignments for each activity are considered constant. It has been modified by a number of researchers such as Dubey (1993), Hiyassat (2000, 2001), Christodoulou *et al.* (2009). Harris (1990) developed another method – Packing Method for Resource Leveling (PACK) – that endeavors to overcome the deficiencies of the minimum moment algorithm. Easa (1989) presented an integer-linear optimization model for resource leveling, which intends to minimize the sum of the absolute deviations

between the resource usage and the average resource usage. Savin *et al.* (1996) recommended a neural network model whereas Hegazy (1999) used genetic algorithms by considering the minimum moment algorithm. Son and Skibniewski (1999) presented a hybrid resource leveling model, combining a local optimizer with simulated annealing so as to minimize the sum of the squares of daily resource usage. Leu *et al.* (2000) also proposed a genetic algorithm based optimization model by minimizing the sum of the absolute deviations between the resource usage and the average resource usage. Hegazy and Ersahin (2001) proposed a spreadsheet model to integrate of the CPM network scheduling with resource leveling and resource allocation, using the minimum moment algorithm. Senouci and Adeli (2001) recommended a neural dynamics model that is based on the minimization of the absolute deviation between daily resource usage. Senouci and Eldin (2004) also presented a genetic algorithm-based model that performs resource leveling and resource constrained scheduling simultaneously with an objective function that attempts to minimize absolute deviations between daily resource usage. El-Rayes and Jun (2009) proposed two objective functions that are essentially based on minimization of the sum of absolute deviations in daily resource usage. Doulabi *et al.* (2011) developed a genetic algorithm-based model that allows activity splitting, using an objective function that minimizes the sum of the absolute deviations of the daily resource usage from the average daily resource usage.

3. Objective functions for resource leveling

The literature review on resource leveling indicated that nine different objective functions had been used by various

researchers to obtain a leveled resource histogram (Table 1). The use of different objective functions for resource leveling may provide an important insight to management teams for efficient use of different resource types. Neither different objective functions nor different resource leveling models may provide uniform resource usage for every construction project (Mattila, Abraham 1998). Therefore, the goal of resource leveling should be either making the resource usage as uniform as possible or making a particular non-uniform resource distribution according to the resource type and the needs of the project and the contractor (Mattila, Abraham 1998; Harris 1978; Senouci, Adeli 2001; Easa 1989). This can be achieved by selecting the most appropriate objective function.

4. Genetic algorithms

Genetic algorithms are search methods for locating the global optimal solution, based upon the idea of simulating the natural selection process in which stronger individuals stay alive in a competing environment (Man *et al.* 1996; Al-Tabtabai, Alex 1999; Hegazy 1999; Leu *et al.* 2000; Senouci, Eldin 2004). Genetic algorithms are different from traditional optimization methods. These differences might be considered as advantages, such as: (1) using a coding set of variables instead of variables themselves; (2) searching for a population of solutions for the problem in preference to improving a single solution (Goldberg 1989; Senouci, Eldin 2004). Moreover, there is a mounting interest for using genetic algorithms as an alternative to existing optimization models, because genetic algorithms are efficient in finding optimal solutions for large and complex problems (Hegazy 1999; Senouci, Eldin 2004).

Table 1. Objective functions for resource leveling

| Objective function No. | Optimization criteria | Source(s) |
|------------------------|--|---|
| 1 | Minimization of the sum of the absolute deviations in resource usage for a determined time interval (day, week, etc.) | Wagner <i>et al.</i> (1964), Ahuja (1976), Easa (1989) |
| 2 | Minimization of the sum of the only increases in resource usage for a determined time interval (day, week, etc.) | Wagner <i>et al.</i> (1964) |
| 3 | Minimization of the sum of the absolute deviations between resource usage for a determined time interval (day, week, etc.) and the average resource usage | Wagner <i>et al.</i> (1964), Easa (1989), Leu <i>et al.</i> (2000), Doulabi <i>et al.</i> (2011) |
| 4 | Minimization of the maximum resource usage for a determined time interval (day, week, etc.) | Wagner <i>et al.</i> (1964) |
| 5 | Minimization of the maximum deviation in resource usage for a determined time interval (day, week, etc.) | Wagner <i>et al.</i> (1964) |
| 6 | Minimization of the maximum absolute deviation between resource usage for a determined time interval (day, week, etc.) and the average resource usage | Popescu (1976) |
| 7 | Minimization of the sum of the square of resource usage for a determined time interval (day, week, etc.) | Burgess and Killebrew (1962), Ahuja (1976), Harris (1978), Harris (1990), Son and Skibniewski (1999), Hiyassat (2000), Hegazy and Ersahin (2001), Hiyassat (2001), Christodoulou <i>et al.</i> (2009) |
| 8 | Minimization of the sum of the square of the deviations in resource usage for a determined time interval (day, week, etc.) | Popescu (1976) |
| 9 | Minimization of the sum of the square of the deviations between resource usage for a determined time interval (day, week, etc.) and the average resource usage | Popescu (1976) |

Genetic algorithms consider that a candidate solution of a problem exists in a population of chromosomes (Hegazy 1999; Leu, Hung 2002). Each chromosome consists of a series of genes that represent the value of a variable for a particular problem. These values might be represented as binary or real numbers according to the nature of the problem or the preference of the user (Hegazy 1999; Leu, Yang 1999b).

The simplest form of genetic algorithms consists of three main stages, namely reproduction, crossover, and mutation (Goldberg 1989; Senouci, Eldin 2004; Sesok, Belevicius 2008). The chromosomes evolve through a reproduction process among the population members. This process is conducted by crossover and mutation. In crossover (also known as marriage), two randomly selected parent chromosomes merge by exchanging their information, in order to produce a pair of offspring that takes part in the population as an alternative solution for the problem (Fig. 1). In other words, a particular parent chromosome is fragmented into parts through a pre-determined crossover rate in order to exchange its parts with the corresponding parts of another parent chromosome (Senouci, Eldin 2004). Crossover rate is the parameter that affects the probability at which the crossover operator is applied. The crossover rate is usually high as it introduces new strings more quickly into the population. On the other hand, while too high crossover rate may cause high performance strings to be eliminated faster than selection can produce improvements, too low crossover rate may cause stagnation due to the lower exploration rate.

In mutation (Fig. 2), contrary to crossover, an offspring is produced by arbitrarily changing the genes of a parent chromosome selected randomly from the population. In other words, a chromosome is modified according to a pre-determined mutation rate in order to produce an offspring that is nonexistent in the population. Mutation rate is the parameter that determines the probability that mutation will occur. The mutation rate is usually set low as a very high mutation rate may result in primitive random search. The mutation process has a natural advantage in that it provides solutions that may never be explored without this process. Besides, it can break any stagnation in the evolutionary process. Whenever an offspring is produced, its fitness is evaluated in accordance with a fitness function and the constraints. A fitness function ($F(x)$) is derived from the objective function ($f(x)$) of a minimization or

maximization problem. The fitness function can be the same as the objective function where a maximization problem is considered, because chromosomes that will survive are determined by maximizing the fitness function. Nevertheless, the objective function of a minimization problem has to be transformed to a fitness function that deals with a maximization problem (Eqn 1). Every single chromosome in a population represents a potential solution for a particular problem (Hegazy 1999; Leu, Hung 2002; Prashant, Ganguli 2011; Senouci, Eldin 2004; Wenyuan 2011).

$$F(x) = 1 / (1 + f(x)). \tag{1}$$

This process (Fig. 3) ends after the generation of a chromosome that represents the optimum or near-optimum solution for a particular problem (Hegazy 1999; Leu *et al.* 2000; Senouci, Eldin 2004).

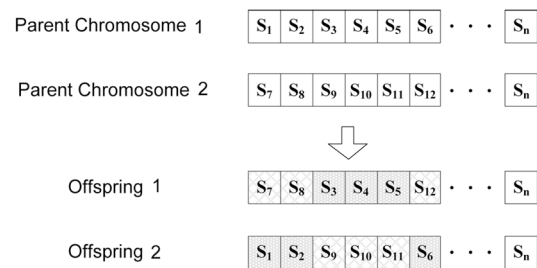


Fig. 1. Crossover process

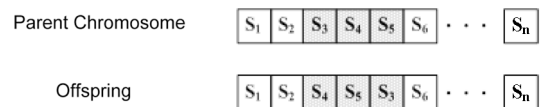


Fig. 2. Mutation process

5. Case study

A real-life steel framed industrial building construction project is used in order to demonstrate the impacts of using different objective functions on resource histograms through a genetic algorithm-based model. The industrial building is 48 m in width, 208 m in length, 8 m in height, and 16 m in column span. The activity network of the construction project is established by MS Project without considering any resource limitations. The network consists of six activities that are repetitive throughout the construction project: (1) fabrication of primary and secondary columns; (2) fabrication of trusses;

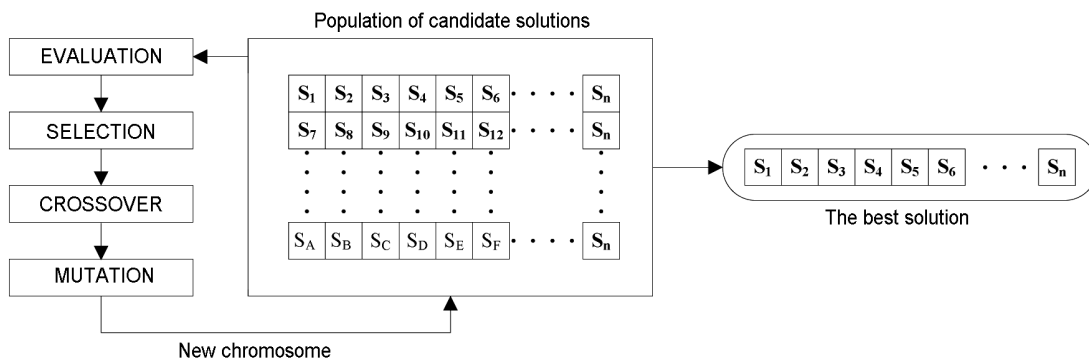


Fig. 3. Example for an operational flow of a genetic algorithm

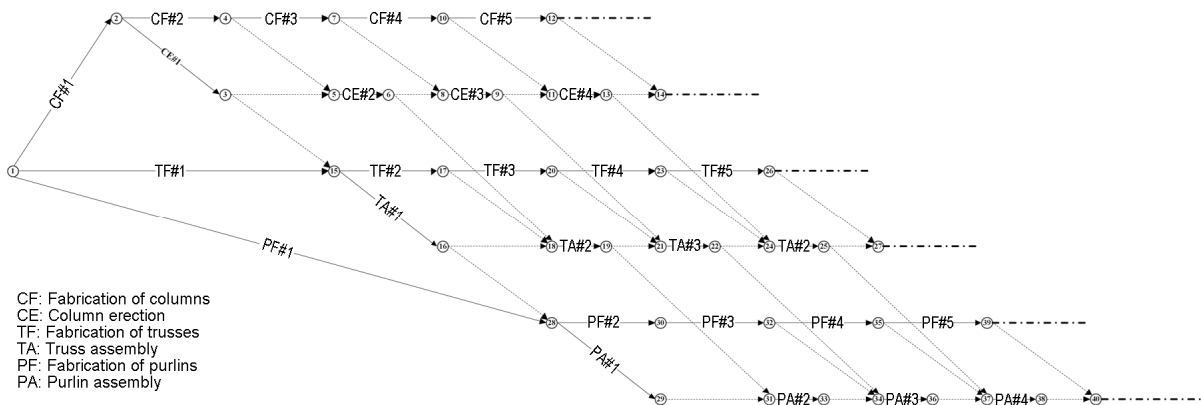


Fig. 4. Logic diagram of the CPM network

(3) fabrication of purlins; (4) column erection; (5) truss assembly; and (6) purlin assembly. The total number of all the repetitive activities is 2,530 which starts with site mobilization and ends with the completion of the steel framed industrial building. The information concerning activity durations, precedence relationships and logical and technological interdependencies between activities, and the resources required to carry out these activities was obtained through interviews with experienced civil engineers who work at the construction company that undertook the construction of this steel framed industrial building. Precedence relationships and logical and technological interdependencies between activities in the CPM network are shown in Figure 4.

MS Project scheduling software was used to determine the total project duration – 44 days – and the free float values of the non-critical activities. Although a great number of different resource types are necessary for completing the activities of the steel framed industrial building, only workers required for each activity in the network were considered in this study. Each activity has a constant resource demand. In this study, the total completion time of the project has not been altered, which means that the critical path determined through basic CPM calculations has not changed. In other words, the resource leveling has been made within the completion time of the construction project (44 days) that was determined in the initial schedule. The logical and technological relationships between the activities were not violated.

First, the resource histogram for the initial CPM network was formed. The number of resource-days obtained from the initial histogram (7,753 workers-days) is divided by the total duration of the project (44 days) in order to calculate the average daily resource usage (Ave): $7,753 \text{ workers-days} / 44 \text{ days} = 176.20 \text{ workers/day}$. This result is rounded up to 177 workers due to the nature of the resource type (i.e. worker) that is used in this study. Also, the sum of the absolute values of the deviations between the resource usage on any day and the average resource usage is 3,561 for the initial resource histogram (Fig. 5).

In this study, Evolver 5.5 is used, which is an add-in program for MS Excel, in order to solve the resource leveling problem through genetic algorithms. Evolver 5.5

was selected as it has features such as fast model setup, handy playback controls, and process speed. Hegazy and Ersahin (2001) also used an earlier version of Evolver (Evolver 4.4) in their study. The application process of Evolver 5.5 consists of defining the MS Excel cells that represent the variables, the objective function, the constraints, and the identification of selection strategy.

| | |
|------------------------------------|------------------------------------|
| Average resource usage (Ave) = 177 | Maximum resource usage (Max) = 348 |
| Total deviation from "Ave" = 3,548 | Minimum resource usage (Min) = 2 |

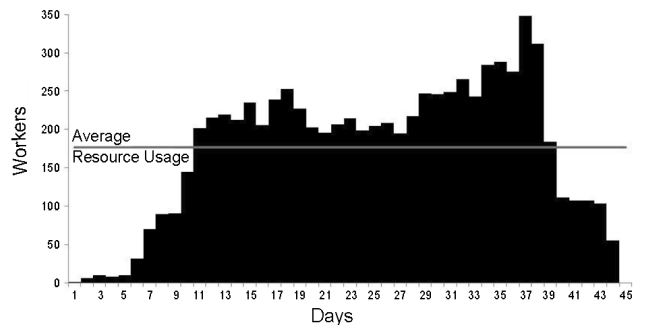


Fig. 5. Resource histogram before resource leveling

The first step is chromosome representation. Each chromosome consists of genes and represents a solution for the resource leveling problem (Liu *et al.* 2005). In this study, a chromosome consists of a number of genes which represents the start times of the activities. In other words, the value of a gene corresponds to a possible start time of a particular activity. For instance, the chromosome [5, 10, ..., n] means that the start time of the first activity is the 5th day, and the start time of the second activity is the 10th day. The total number of genes is equal to the total number of activities. Evolver 5.5 enables the user to determine the range of variables. The variables of this study are the start times of the non-critical activities, which have free float. Free float is the time by which the completion of an activity can be delayed beyond its earliest finish time without affecting the earliest start of any other succeeding activities. The start times of these non-critical activities may assume any value within the limitations of their free floats as expressed in Eqn (2). The precedence relationships and logical and technological interdependencies between

activities were considered during the construction of the network. Therefore, shifting the start times of the non-critical activities within the limitations of their free floats does not violate the precedence relationships in the network.

The second step is the determination of the objective function, which evaluates each chromosome generated. Evolver 5.5 allows the user to select a MS Excel cell in order to define the objective function. In this study, nine different objective functions were used for the solution of the resource leveling problem without changing the total project time. The formulations of the objective functions are presented in Table 2.

After the determination of the objective function, a number of constraints were specified via the command

window of Evolver 5.5. Evolver 5.5 makes it possible for the user to define the constraints in the form of hard or soft constraints. A hard constraint discards the solutions that do not meet the constraint. A soft constraint disfavors the solutions that do not meet the constraint based on a user defined penalty function (Evolver 2010). Hard constraints were used in the resource leveling models because the early and late start times should not be overrun (Eqn (2)). The early and late start times were determined according to the free floats calculated according to basic CPM principles.

$$\begin{aligned} \text{Early start time} &\leq \text{Start time of the activity} \leq \\ \text{Early start time} &+ \text{Free float} \end{aligned} \quad (2)$$

Table 2. Objective functions for resource leveling

| Objective function No. | Optimization criteria | Formulas | Notations |
|------------------------|---|--|--|
| 1 | Minimization of the sum of the absolute deviations in daily resource usage | $Z = \min \sum_{i=1}^T Rdev_i $ | min = minimize; i = day under consideration; T = the duration of the project; $Rdev_i$ = deviation between resources required on day i and $i + 1$. |
| 2 | Minimization of the sum of only the increases in daily resource usage from one day to the next | $Z = \min \sum_{i=1}^T Rinc_i $ | min = minimize; i = day under consideration; T = the duration of the project; $Rinc_i$ = increase in between resources required on day i and $i + 1$. |
| 3 | Minimization of the sum of the absolute deviations between daily resource usage and the average resource usage | $Z = \min \sum_{i=1}^T R_i - A_{rr} $ | min = minimize; i = day under consideration; T = the duration of the project; R_i = resources required on day i ; A_{rr} = average resource usage. |
| 4 | Minimization of the maximum daily resource usage | $Z = \min [\max (R_i)]$ | min = minimize; max = maximum; i = day under consideration; R_i = resources required on day i . |
| 5 | Minimization of the maximum deviation in daily resource usage | $Z = \min [\max Rdev_i]$ | min = minimize; max = maximum; i = day under consideration; $Rdev_i$ = deviation between resources required on day i and $i + 1$. |
| 6 | Minimization of the maximum absolute deviation between daily resource usage and the average resource usage | $Z = \min [\max R_i - A_{rr}]$ | min = minimize; max = maximum; i = day under consideration; R_i = resources required on day i ; A_{rr} = average resource usage. |
| 7 | Minimization of the sum of the square of daily resource usage | $Z = \min M_x = \sum_{i=1}^T (R_i)^2$ | min = minimize; i = day under consideration; T = the duration of the project; R_i = resources required on day i . |
| 8 | Minimization of the sum of the square of the deviations in daily resource usage | $Z = \min \sum_{i=1}^T (Rdev_i)^2$ | min = minimize; i = day under consideration; T = the duration of the project; $Rdev_i$ = deviation between resources required on day i and $i + 1$. |
| 9 | Minimization of the sum of the square of the deviations between daily resource usage and the average resource usage | $Z = \min \sum_{i=1}^T (R_i - A_{rr})^2$ | min = minimize; i = day under consideration; T = the duration of the project; R_i = resources required on day i ; A_{rr} = average resource usage. |

The next step is to identify the selection strategy. Evolver 5.5 provides six solving methods which can be selected by the user according to the characteristics of the problem. In this study, the “recipe” method was selected because in this method, each variable can be adjusted independently from the other variables. Hegazy and Ersahin (2001) also used the same solving method in their study. After the objective function and constraints are set and the selection strategy is selected, the population size and the crossover and mutation rates should be set. Evolver 5.5 has default values for these operators. However, it also allows the user to change these default values. Population size has an important effect on both the solution and the time the process takes. Although the possibility of obtaining an optimum solution may increase when the population size is larger, the run time may increase considerably too (Hegazy 1999; Hegazy, Kassab 2003). The population size was set as 50 which is the default value of Evolver 5.5. When a new offspring is to be created, two parent chromosomes are chosen from the population. In Evolver 5.5, parent chromosomes are chosen with a rank-based mechanism which prevents good chromosomes from entirely dominating the evolution from an early point. This approach provides a smoother selection probability curve than conventional genetic algorithm systems, where a parent’s chance to be chosen for reproduction is proportional to its fitness. Evolver allows the user to select more than one crossover and mutation operators, (e.g. arithmetic crossover, heuristic crossover, Cauchy mutation, boundary mutation) thus it automatically tests valid combinations of the selected operators to identify the best performing ones for your model (Evolver 2010).

The crossover and mutation rates were determined after the genetic algorithm-based model had been tested with various values of the crossover and mutation rates. These tests were performed with crossover rates of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9 as it was the case in the studies of Leu and Yang (1999b), Leu *et al.* (2000) and Leu and Hung (2002), and an auto-mutation rate adjustment. Evolver 5.5 provides an auto-mutation rate adjustment option to increase the mutation rate automatically when the best solution remains the same over an extended number of trials. The model was run for three times with each crossover rate for each of the nine objective functions separately (e.g. three trials with the crossover rate of 0.1 and auto-mutation rate adjustment for nine different objective functions). Afterwards, the arithmetic mean of the three results was calculated in order to decide which one of the crossover rates performs better than others. It was found that the crossover rate of 0.1, 0.7, 0.5, 0.5, 0.1, 0.5, 0.6, 0.1, and 0.3 performed better than others for Objective Functions 1 to 9, respectively.

An Excel sheet that consists of 51 columns and 2,530 rows was created. The rows represent the activity names. The seven columns consist of activity duration, early/late start times, early/late finish times, free floats, and required daily resources. The rest of the columns (44 columns) represent the project days of the schedule. This Excel sheet provides the required information for the resource leveling

process. The required resources for an activity on a specific day may change as the start times of non-critical activities are shifted along their available free float times in order to level resources, however, any non-critical activity with zero free float is started on the scheduled time (i.e. early start time) and completed within the stipulated duration. The model automatically changes the required daily resources as it shifts the non-critical activities along their available free float times. Once all the parameters are set genetic algorithm-based model was run by Evolver 5.5. The process was carried out separately for nine different objective functions. Each of the processes was stopped after 100,000 trials because the model stopped generating better solutions. The average runtime for 100,000 trials on a personal computer (PC) with a 2.53 GHz Core2 Duo CPU and 4 GB RAM was approximately 2 hours. The resource histograms, which show the distributions of resource usage, were generated by MS Excel. The results are presented in Figure 6.

It was observed that each of the objective functions generates different resource utilization histograms after resource leveling. In other words, nine different objective functions bring about nine different solutions since each of the objective functions tries to minimize different parameters (i.e. the sum of the absolute deviations in daily resource usage, the sum of only the increases in daily resource usage from one day to the next, the sum of the absolute deviations between daily resource usage and the average resource usage, the maximum daily resource usage, the maximum deviation in daily resource usage, the maximum absolute deviation between daily resource usage and the average resource usage, the sum of the square of daily resource usage, the sum of the square of the deviations in daily resource usage, the sum of the square of the deviations between daily resource usage and the average resource usage). In order to assess these results, the values of nine different parameters were calculated for each of these leveled resource utilization histograms. The initial values of these parameters calculated using the earliest start times of the activities (i.e. before leveling) and the new values of these parameters, which were calculated using the start times of the activities, determined after resource leveling for each of nine objective functions are presented in Table 3. As seen in Table 3, while the values of some of these parameters improved after resource leveling when compared to their initial values, some of them did not. In order to determine the improvement levels achieved by resource leveling using nine different objective functions, the improvement percentage in each parameter and the average improvement percentage for each objective function were calculated. It should be noted that the importance of each parameter was considered to be equal in this study. The findings are shown in Table 4. As seen in Table 4, Objective Function 8 (i.e. minimization of the sum of the square of the deviations in daily resource usage) provided the best average improvement percentage (35%) when compared to the other objective functions. Even though Objective Function 8 provided the best average improvement percentage in the studied case study, another objective function(s) may provide the highest

Table 3. Results for different objective functions after resource levelling

| Objective function No. | Before leveling | After leveling | | | | | | | | |
|------------------------|-----------------|-------------------------|-------------------------|-------------------------------|---------------|------------------|-------------------------|------------------------|---------------------------|---------------------------------|
| | | $\sum_{i=1}^T Rdev_i $ | $\sum_{i=1}^T Rinc_i $ | $\sum_{i=1}^T R_i - A_{rr} $ | $[\max(R_i)]$ | $[\max(Rdev_i)]$ | $[\max R_i - A_{rr}]$ | $\sum_{i=1}^T (R_i)^2$ | $\sum_{i=1}^T (Rdev_i)^2$ | $\sum_{i=1}^T (R_i - A_{rr})^2$ |
| 1 | 1,018 | 628 | 654 | 756 | 1,006 | 1,010 | 1,048 | 636 | 590 | 636 |
| 2 | 509 | 352 | 350 | 467 | 518 | 511 | 525 | 411 | 350 | 411 |
| 3 | 3,558 | 3,287 | 3,387 | 2,424 | 3,383 | 3,370 | 3,519 | 2,452 | 2,894 | 2,430 |
| 4 | 348 | 322 | 318 | 330 | 314 | 342 | 348 | 314 | 314 | 314 |
| 5 | 128 | 122 | 112 | 122 | 120 | 56 | 128 | 98 | 62 | 102 |
| 6 | 176 | 175 | 175 | 175 | 175 | 175 | 175 | 175 | 175 | 175 |
| 7 | 1,729,973 | 1,682,949 | 1,701,513 | 1,615,665 | 1,701,117 | 1,709,313 | 1,725,845 | 1,598,441 | 1,636,853 | 1,598,661 |
| 8 | 50,958 | 32,294 | 30,358 | 35,126 | 45,402 | 38,590 | 50,818 | 25,162 | 15,286 | 26,502 |
| 9 | 394,907 | 347,884 | 366,448 | 280,600 | 366,052 | 374,428 | 390,780 | 263,376 | 301,788 | 263,176 |

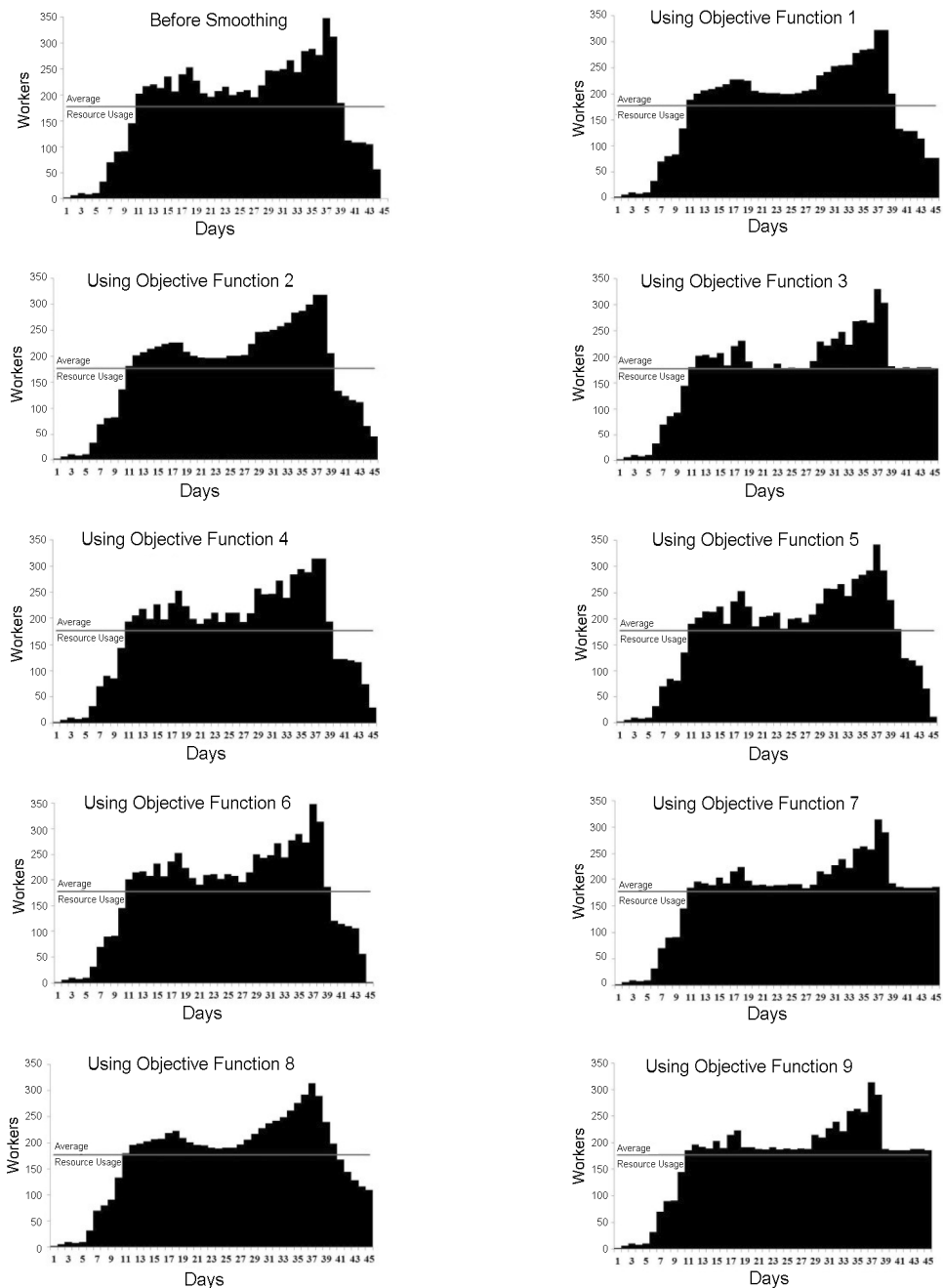


Fig. 6. Resource histogram before and after resource leveling

Table 4. Percentages of improvement (%)

| Objective function No. | The parameters and the improvement percentages in each parameter (%) | | | | | | | | | Average improvement percentage (%) |
|------------------------|--|-------------------------|-------------------------------|---------------|------------------|------------------------|------------------------|---------------------------|---------------------------------|------------------------------------|
| | $\sum_{i=1}^T Rdev_i $ | $\sum_{i=1}^T Rinc_i $ | $\sum_{i=1}^T R_i - A_{rr} $ | $[\max(R_i)]$ | $[\max(Rdev_i)]$ | $[\max R_i - A_{rr}]$ | $\sum_{i=1}^T (R_i)^2$ | $\sum_{i=1}^T (Rdev_i)^2$ | $\sum_{i=1}^T (R_i - A_{rr})^2$ | |
| 1 | 38 | 36 | 26 | 1 | 1 | – | 38 | 42 | 38 | 24 |
| 2 | 31 | 31 | 8 | – | – | – | 19 | 31 | 19 | 16 |
| 3 | 8 | 5 | 32 | 5 | 5 | 1 | 31 | 19 | 32 | 15 |
| 4 | 7 | 9 | 5 | 10 | 2 | – | 10 | 10 | 10 | 7 |
| 5 | 5 | 13 | 5 | 6 | 56 | – | 23 | 52 | 20 | 20 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 3 | 2 | 7 | 2 | 1 | – | 8 | 5 | 8 | 4 |
| 8 | 37 | 40 | 31 | 11 | 24 | – | 51 | 70 | 48 | 35 |
| 9 | 12 | 7 | 29 | 7 | 5 | 1 | 33 | 24 | 33 | 17 |

average improvement percentage in different projects. Therefore, contractors should run the model for nine different objective functions in order to determine the objective function that provides the best average improvement percentage. Also, contractors may assign different weights to these parameters depending on their special needs and determine the leveled resource utilization histogram that best fits their needs.

Conclusion

The objective of this research was to investigate the impacts of using different objective functions on resource utilization histograms in CPM networks. For this purpose, the objective functions that are used in existing resource leveling models were reviewed, and nine different objective functions were identified. Although there are several studies concerning the different resource leveling models for CPM networks, a comparison of those objective functions has never been made. A genetic algorithm-based resource leveling model that considers only a single resource type was developed and run using actual data obtained from a steel-framed industrial building project. The nine objective functions were optimized via this model. It was observed that each of these objective functions generates different resource utilization histograms. In other words, nine different objective functions bring about nine different solutions since each of the objective functions tries to minimize different parameters. In order to determine the improvement levels achieved by resource leveling using nine different objective functions, the improvement percentage in each parameter and the average improvement percentage for each objective function were calculated. The objective function that involves the minimization of the sum of the square of the deviations in daily resource usage provided the best average improvement percentage in the real-life case that was used in this study. However, it should be pointed out that a different objective function(s) may provide the highest average improvement percentage in a different case. This study makes construction professionals aware of the fact that resource leveling may not be optimal without considering the different objective functions. The contractor should run all nine models and should select the objective function(s) that provides the highest average improvement percentage. Moreover, they may assign different

weights to these parameters depending on their special needs and determine the leveled resource utilization histogram that best fits their needs. The implementation of a metaheuristic model for resource leveling that considers different objective functions and multiple resources would be a potential improvement for this study.

References

- Ahuja, H. N. 1976. *Construction performance control by networks*. New York: John Wiley and Sons.
- Al-Tabtabai, H.; Alex, A. P. 1999. Using genetic algorithms to solve optimization problems in construction, *Engineering, Construction and Architectural Management* 6(2): 121–132. <http://dx.doi.org/10.1108/eb021105>
- Burgess, A. R.; Killebrew, J. B. 1962. Variation in activity level on a cyclical arrow diagram, *Journal of Industrial Engineering* 13(2): 76–83.
- Chen, P. H.; Weng, H. 2009. A two-phase GA model for resource-constrained project scheduling, *Automation in Construction* 18(4): 485–498. <http://dx.doi.org/10.1016/j.autcon.2008.11.003>
- Christodoulou, S.; Ellinas, G.; Aslani, P. 2009. Entropy-based scheduling of resource-constrained construction projects, *Automation in Construction* 18(7): 919–928. <http://dx.doi.org/10.1016/j.autcon.2009.04.007>
- Doulabi, S. H. H.; Seifi, A.; Shariat, S. Y. 2011. An efficient hybrid genetic algorithm for resource leveling via activity splitting, *Journal of Construction Engineering and Management* 137(2): 137–146. [http://dx.doi.org/10.1061/\(ASCE\)CO.1943-7862.0000261](http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000261)
- Dubey, A. 1993. *Resource leveling and linear scheduling*: MSc Dissertation, The University of British Columbia.
- Easa, S. M. 1989. Resource leveling in construction by optimization, *Journal of Construction Engineering and Management* 115(2): 302–316. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1989\)115:2\(302\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1989)115:2(302))
- El-Rayes, K.; Jun, D. H. 2009. Optimizing resource leveling in construction projects, *Journal of Construction Engineering and Management* 135(11): 1172–1180. [http://dx.doi.org/10.1061/\(ASCE\)CO.1943-7862.0000097](http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000097)
- Evolver 2010. Guide to using Evolver – the genetic algorithm solver for Microsoft Excel, Palisade Corporation, New York, USA.

- Galloway, P. D. 2006. Survey of the construction industry relative to the use of CPM scheduling for construction projects, *Journal of Construction Engineering and Management* 132(7): 697–711. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2006\)132:7\(697\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2006)132:7(697))
- Goldberg, D. E. 1989. *Genetic algorithms in search, optimization, and machine learning*. Reading, Massachusetts: Addison-Wesley. 412 p.
- Hariga, M.; El-Sayegh, S. M. 2010. Cost optimization model for the multiresource leveling problem with allowed activity splitting, *Journal of Construction Engineering and Management* 137(1): 56–64. [http://dx.doi.org/10.1061/\(ASCE\)CO.1943-7862.0000251](http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000251)
- Harris, R. B. 1978. *Precedence and arrow networking techniques for construction*. New York: John Wiley and Sons.
- Harris, R. B. 1990. Packing method for resource leveling (PACK), *Journal of Construction Engineering and Management* 116(2): 331–350. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1990\)116:2\(331\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1990)116:2(331))
- Hegazy, T. 1999. Optimization of resource allocation and leveling using genetic algorithms, *Journal of Construction Engineering and Management* 125(3): 167–175. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1999\)125:3\(167\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1999)125:3(167))
- Hegazy, T.; Ersahin, T. 2001. Simplified spreadsheet solutions. II: overall schedule optimization, *Journal of Construction Engineering and Management* 127(6): 469–475. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2001\)127:6\(469\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2001)127:6(469))
- Hegazy, T.; Kassab, M. 2003. Resource optimization using combined simulation and genetic algorithms, *Journal of Construction Engineering and Management* 129(6): 698–705. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2003\)129:6\(698\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2003)129:6(698))
- Hinze, J. 2004. *Construction planning and scheduling*. 2nd ed. Upper Saddle River, N.J.: Prentice-Hall.
- Hiyassat, M. A. S. 2000. Modification of minimum moment approach in resource leveling, *Journal of Construction Engineering and Management* 126(4): 278–284. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2000\)126:4\(278\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2000)126:4(278))
- Hiyassat, M. A. S. 2001. Applying modified minimum moment method to multiple resource leveling, *Journal of Construction Engineering and Management* 127(3): 192–198. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2001\)127:3\(192\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2001)127:3(192))
- Jaskowski, P.; Sobotka, A. 2004. Scheduling construction projects with resources accessibility limited and changeable in time, *Journal of Civil Engineering and Management* 10(4): 267–276. <http://dx.doi.org/10.1080/13923730.2004.9636319>
- Leu, S. S.; Hung, T. H. 2002. A genetic algorithm-based optimal resource-constrained scheduling simulation model, *Construction Management and Economics* 20(2): 131–141. <http://dx.doi.org/10.1080/01446190110109148>
- Leu, S. S.; Yang, C. H. 1999a. GA-based multicriteria optimal model for construction scheduling, *Journal of Construction Engineering and Management* 125(6): 420–427. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1999\)125:6\(420\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1999)125:6(420))
- Leu, S. S.; Yang, C. H. 1999b. A genetic-algorithm-based resource-constrained construction scheduling system, *Construction Management and Economics* 17(6): 767–776. <http://dx.doi.org/10.1080/0144619993711105>
- Leu, S. S.; Yang, C. H.; Huang, J. C. 2000. Resource leveling in construction by genetic algorithm-based optimization and its decision support system application, *Automation in Construction* 10(1): 27–41. [http://dx.doi.org/10.1016/S0926-5805\(99\)00011-4](http://dx.doi.org/10.1016/S0926-5805(99)00011-4)
- Liu, Y.; Zhao, S. L.; Du, X. K.; Li, S. Q. 2005. Optimization of resource allocation in construction using genetic algorithms, in *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, 3428–3432.
- Lu, M.; Lam, H. C.; Dai, F. 2008. Resource-constrained critical path analysis based on discrete event simulation and particle swarm optimization, *Automation in Construction* 17(6): 670–681. <http://dx.doi.org/10.1016/j.autcon.2007.11.004>
- Lu, M.; Li, H. 2003. Resource-activity critical-path method for construction planning, *Journal of Construction Engineering and Management* 129(4): 412–420. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2003\)129:4\(412\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2003)129:4(412))
- Man, K. F.; Tang, K. S.; Kwong, S. 1996. Genetic algorithms: concepts and applications, *Transactions on Industrial Electronics* 43(5): 519–533. <http://dx.doi.org/10.1109/41.538609>
- Mattila, K. G.; Abraham, D. M. 1998. Resource leveling of linear schedules: using integer linear programming, *Journal of Construction Engineering and Management* 124(3): 232–244. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1998\)124:3\(232\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1998)124:3(232))
- Osman, I. H.; Laporte, G. 1996. Metaheuristics: a bibliography, *Annals of Operations Research* 63: 513–623. <http://dx.doi.org/10.1007/BF02125421>
- Popescu, C. M. 1976. *How to use C.P.M. in practice*. Part II – Resources. University of Texas at Austin.
- Prashant, M. P.; Ganguli, R. 2011. *Structural health monitoring using genetic fuzzy systems*. New York: Springer. 138 p.
- Rogalska, M.; Hejducki, Z. 2007. Time buffers in construction process scheduling, *Journal of Civil Engineering and Management* 13(2): 143–148. <http://dx.doi.org/10.1080/13923730.2007.9636430>
- Savin, D.; Alkass, S.; Fazio, P. 1996. Construction resource leveling using neural networks, *Canadian Journal of Civil Engineering* 23(4): 917–925. <http://dx.doi.org/10.1139/196-898>
- Senouci, A. B.; Adeli, H. 2001. Resource scheduling using neural dynamics model of Adeli and Park, *Journal of Construction Engineering and Management* 127(1): 28–34. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2001\)127:1\(28\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2001)127:1(28))
- Senouci, A. B.; Eldin, N. N. 2004. Use of genetic algorithms in resource scheduling of construction projects, *Journal of Construction Engineering and Management* 130(6): 869–877. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2004\)130:6\(869\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2004)130:6(869))
- Sesok, D.; Belevicius, R. 2008. global optimization of trusses with a modified genetic algorithm, *Journal of Civil Engineering and Management* 14(3): 147–154. <http://dx.doi.org/10.3846/1392-3730.2008.14.10>

- Son, J.; Skibniewski, M. J. 1999. Multiheuristic approach for resource leveling problem in construction engineering: hybrid approach, *Journal of Construction Engineering and Management* 125(1): 23–31.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1999\)125:1\(23\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1999)125:1(23))
- Taha, H. A. 1995. *Operations research: an introduction*. 5th ed. New Jersey: Prentice Hall International, Inc.
- Wagner, H. M.; Giglio, R. J.; Glaser, R. G. 1964. Preventive maintenance scheduling by mathematical programming, *Management Science* 10(2): 316–334.
<http://dx.doi.org/10.1287/mnsc.10.2.316>
- Wenyuan, L. 2011. *probabilistic transmission system planning*. New Jersey: Wiley-IEEE Press.
- Wiest, J. D.; Levy, F. K. 1977. *A management guide to PERT/CPM: With GERT/PDM/DCPM and other networks*. 2nd ed. New Jersey: Prentice-Hall.

Atilla DAMCI. He is currently a Research Assistant at the Department of Civil Engineering, Istanbul Technical University. He served as a Visiting Research Scholar at Illinois Institute of Technology, Chicago, USA in 2011–2012. His research work over the years has resulted in 18 technical papers in refereed journals and proceedings. His main areas of research interest include scheduling, linear scheduling, resource management, and most aspects of construction management.

Gul POLAT. She is an Associate Professor of Civil Engineering at Istanbul Technical University, where she also serves as the Director of the Construction Management Graduate Program. She conducted her PhD studies at Istanbul Technical University and Illinois Institute of Technology in Chicago. She has been involved with several research projects funded by Istanbul Technical University (ITU), Feyzi Akkaya Fund for Supporting Scientific Activities (FABED), and other construction industry agencies. Her research work over the years has resulted in 74 technical papers in refereed journals and proceedings. She has served over the years on several organizing committees of national and international conferences. Dr. Polat is currently serving on the editorial board of an international journal and routinely reviews papers for several scholarly publications. Her main areas of research interest include decision support techniques, resource management, marketing management, risk management, precast concrete systems, and most aspects of construction management.