ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

MULTIOBJECTIVE OPTIMIZATION USING GENETIC ALGORITHMS IN TIME-COST CONSTRUCTION PROJECT SCHEDULING PROBLEM

J. Magalhães-Mendes[1]

[1] Departamento de Engenharia Civil
CIDEM
ISEP - Instituto Superior de Engenharia – Instituto Politécnico do Porto
Rua Dr. António Bernardino de Almeida, 431, 4249-015 Porto
PORTUGAL
e-mail: jjm@isep.ipp.pt

Keywords: Multiobjective optimization, genetic algorithms, project management, construction management.

Abstract. *Multiobjective formulations are realistic models for many complex engineering optimization problems [1]. In a construction project, there are two main factors, such as project duration and project cost. The activity duration is a function of resources (i.e. crew size, equipments and materials) availability. On the other hand, resources demand direct costs. Therefore, the relationship between project time and direct cost of each activity is a monotonously decreasing curve. It means if activity duration is compressed then that leads to an increase in resources and so that direct costs. But, project indirect costs increase with the project duration. In general, for a project, the total cost is the sum of direct and indirect costs and exists an optimum duration for the least cost. Hence, relationship between project time and cost is trade-off [2].*

There are two general approaches to multiple-objective optimization. One is to combine the individual objective functions into a single composite function. Determination of a single objective is possible with methods such as utility theory, weighted sum method, etc., but the problem lies in the correct selection of the weights or utility functions to characterize the decision-makers preferences [1].

The main difficulty with the single composite function is selecting a weight vector for each run. To overcome this drawback a GA based-approach to solving the time-cost optimization problem has been proposed. The idea is transforming weights in the objective function by genes obtained from the genetic algorithm for each run.

The present approach provides a powerful alternative for the solution of the time-cost construction project scheduling problems when compared against others approaches including evolutionary algorithms.

1 INTRODUCTION

The time and cost are usually two objectives which are often tradeoff in project practices. When the construction time is shortened, the project cost should be added. The main objective of construction project management is to execute the project within the anticipated time while satisfying the minimum cost.

Several approaches to solve the time-cost optimization (TCO) problem have been proposed in the last years: mathematical, heuristic and search methods.

Several mathematical models such as linear programming (Hendrickson and Au [5]; Pagnoni [3]), integer programming, or dynamic programming (Robinson [8]; P. De et al. [23]) and LP/IP hybrid (Liu et al. [20]; Burns et al. [25]) and Meyer and Shaffer [27] use mixed integer programming. However, for large number of activity in network and complex problem, integer programming needs a lot of computation effort (Feng et al. [6]).

Heuristic algorithms are not considered to be in the category of optimization methods. They are algorithms developed to find an acceptable near optimum solution. Heuristic methods are usually algorithms easy to understand which can be applied to larger problems and typically provide acceptable solutions (Hegazy [26]). However, they have lack mathematical consistency and accuracy and are specific to certain instances of the problem (Fondahl [19]; Siemens [22]) are some of the research studies that have utilized heuristic methods for solving TCO problems.

Some researchers have tried to introduce evolutionary algorithms to find global optima such as genetic algorithm (GA) (Feng et al. [6]; Gen and Cheng [21]; Zheng et al. [10]; Zheng and Ng [9]; Mendes [16, 18] and Parveen and Saha [32]) the particle swarm optimization algorithm (Yang [10]), ant colony optimization (ACO) (Xiong and Kuang [28]; Ng and Zhang [24]; Afshar et al. [2]) and harmony search (HS) (Geem [29]). In this paper, the optimal time and cost generated by the GA techniques are compared with those produced by other techniques through some problems obtained from literature.

This paper is organized as follows. Section 2 describes the multiobjective optimization problem. Section 3 presents the approach. The case study and results are presented in Section 4. Finally, conclusions and future work are outlined in Section 5.

2 MULTIOBJECTIVE OPTIMIZATION

Multiobjective optimization deals with solving optimization problems which involve multiple objectives. We can say that there are two types of methods for solving problems with multi-objective optimization: the classical methods and methods based on evolutionary algorithms.

The disadvantages of the classical methods are shown in [31]:

- Only one non-dominated solution is obtained by each execution of the algorithm. It means that in order to get a set of solutions, it should be run many times.
- Some of them require some kind of information of the problem treated.
- Some of them are sensitive to the shape of the Pareto frontier, so in non-convex ones, they cannot find solutions.
- The dispersion of the founded Pareto solutions depends on the efficiency of the monocriteria optimizator.
- In problems that contain stochasticities, classical methods are not appropriate.
- Problems with discrete domain cannot be solved by classical methods, neither in the multiobjective case. Consequently, the problem treated in the present article, discrete, could not be solved by this kind of methods.

All this disadvantages are overcome with evolutionary multiobjective methods such as genetic algorithms (MOGA) [30].

With evolutionary algorithms being used for single-objective optimization for over two decades, the incorporation of more than one objective in the fitness function has finally gained popularity in the research [33].

The approach presented in this paper is based on a random key based genetic algorithm to perform its optimization process, so this approach aims to stipulate multiple search directions at each generation without using any additional parameters.

2.1 Linear Combination of Weights

The classical approach to solve a multi-objective optimization problem is to assign a weight w_i to each normalized objective function $z_i^{'}(x)$ so that the problem is converted to a single objective problem with a scalar objective function as follows [35]:

$$\min z = w_1 * z_1^{'}(x) + w_2 * z_2^{'}(x) + ... + w_n * z_n^{'}(x) \tag{1}$$

where $z_i^{'}(x)$ is the normalized objective function of $z_i\ (x)$ and $\Sigma\, w_i\ = 1$. The main difficulty with this approach is selecting a weight vector for each run.

2.2 Linear Combination of Genes

Linear Combination of Weights also called the weighted sum method (WSM) is the simplest approach and probably the most widely used classical method. This method scalarizes the set of objectives into a single objective by multiplying each objective with a user supplied weight.

Based on the WSM idea, this paper proposes a linear combination of genes where $z_i^{'}(x)$ is the normalized objective function of $z_i\ (x)$, with the following formulation:

$$\min z = gene_1 * z_1^{'}(x) + gene_2 * z_2^{'}(x) + ... + gene_n * z_n^{'}(x) \tag{2}$$

where $z_i^{'}(x)$ is the normalized objective function of $z_i\ (x)$ and $0 < \sum gene_i < n$, each $gene_i$ is randomly generated for individual solution $\boldsymbol{x}$ during the selection phase at each generation.

3 THE GA-BASED APPROACH

The approach combines a genetic algorithm, a schedule generation scheme and a local search procedure. The genetic algorithm is responsible for evolving the chromosomes which represent the priorities of the activities, delay times and objective function.

For each chromosome the following four phases are applied [16]:

1) *Transition parameters* - this phase is responsible for the process transition between first level and second level;
2) *Schedule parameters* - this phase is responsible for transforming the chromosome supplied by the genetic algorithm into the priorities of the activities and delay time;
3) *Schedule generation* - this phase makes use of the priorities and the delay time and constructs schedules;
4) *Schedule improvement* - this phase makes use of a local search procedure to improve the solution obtained in the schedule generation phase.

This study considers both project cost and time. For effective time-cost optimization the approach proposes an objective function with the following formulation:

$$Z_1^{'} = Z_{time}^{'} = \frac{(Z_t^{\max} - Z_t)}{(Z_t^{\max} - Z_t^{\min})} \tag{3}$$

$$Z_2^{'} = Z_{\cos t}^{'} = \frac{(Z_c^{\max} - Z_c)}{(Z_c^{\max} - Z_c^{\min})} \tag{4}$$

and finally:

$$\min f(x) = Gene_t \frac{(Z_t^{\max} - Z_t)}{(Z_t^{\max} - Z_t^{\min})} + Gene_c \frac{(Z_c^{\max} - Z_c)}{(Z_c^{\max} - Z_c^{\min})} \tag{5}$$

where,

$Z_c^{\max}$ = maximal value for total cost in the current chromosome;
$Z_t^{\max}$ = maximal value for time in the current chromosome;
$Z_c^{\min}$ = minimal value for total cost in the initial population;
$Z_t^{\min}$ = minimal value for time in the initial population;
Z_c = represents the total cost of the xth solution in current chromosome;
Z_t = represents the time of the xth solution in current chromosome.

3.1 GA-Decoding

Each chromosome represents a solution to the problem and it is encoded as a vector of random keys (random numbers). Each solution encoded as initial chromosome (first level) is made of 2+*mn*+*n* genes where *n* is the number of activities and *m* is the number of execution modes, see Figure 1 [16, 18].

Genes for Activity 1	Activity 1	Mode 1	Gene $_{11}$
		Mode 2	Gene $_{12}$
		…	…
		Mode m	Gene $_{1m}$
		Delay 1	Gene $_{1m+1}$
Genes for Activity 2	Activity 2	Mode 1	Gene $_{21}$
		Mode 2	Gene $_{22}$
		…	…
		Mode m	Gene $_{2m}$
		Delay 2	Gene $_{2m+1}$
… .	…		…
Genes for Activity n	Activity n	Mode 1	Gene $_{n1}$
		Mode 2	Gene $_{n2}$
		…	…
		Mode m	Gene $_{nm}$
		Delay n	Gene $_{nm+1}$
Genes for Objective Function	OF	Time	Gene $_t$
		Cost	Gene $_c$

Figure 1: Chromosome structure.

To decode each chromosome a schedule generation scheme (SGS) based on the idea of parameterized active schedules is applied [11, 14, 16]. This type of schedule consists of schedules in which no resource is kept idle for more than a predefined period if it could start processing some activity [16] and employs operators described in [15, 16].

3.2 Evolutionary Strategy

The GA based-approach uses an evolutionary strategy identical to the one proposed by Goldberg [7]. To breed good solutions, the population is operated by a genetic algorithm. There are many variations of genetic algorithms obtained by altering the reproduction, crossover, and mutation operators.

In this approach reproduction is accomplished by first copying some of the best individuals from one generation to the next, in what is called an elitist strategy.

The fitness proportionate selection, also known as roulette-wheel selection, is the genetic operator for selecting potentially useful solutions for reproduction. The characteristic of the roulette wheel selection is stochastic sampling.

The fitness value is used to associate a probability of selection with each individual chromosome. If f_i is the fitness of individual i in the population, its probability of being selected is,

$$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \quad , \qquad i = 1, \ldots, n \tag{6}$$

A roulette wheel model is established to represent the survival probabilities for all the individuals in the population. Then the roulette wheel is rotated for several times.

After selecting, crossover may proceed in two steps. First, members of the newly selected (reproduced) chromosomes in the mating pool are mated at random. Second, each pair of chromosomes undergoes crossover as follows: an integer position k along the chromosome is selected uniformly at random between 1 and the chromosome length l. Two new chromosomes are created swapping all the genes between k+1 and l, see Mendes [17].

The mutation operator preserves diversification in the search. This operator is applied to each offspring in the population with a predetermined probability. We assume that the probability of the mutation in this paper is 5% [17].

4 CASE STUDY

The GA based-approach is to minimize the project overall time and cost and RKTCO is applied in the case study of a project of eighteen activities originally introduced by Feng et al. [6], see Figure 2. The activity relationship for the model project consists of 18 activities and three modes of construction for each activity and their associated time and cost are presented in [6, 16]. Indirect cost rate was $1000/day.

The Table 1 shows the results for several mathematical and evolutionary-based methods. The algorithm RKTCO obtains better solution than the other GA-based approaches. Furthermore, the algorithm RKTCO reaches the optimal solution quickly, i.e., in five seconds.

The results of the RKTCO illustrates that evolutionary methods based on genetic algorithms can obtain the better solutions and in very reasonable computational time.

Based on Feng et al. [6] five new problems are generated: three experiments using the parallel expansion of the 18-activity problem and two using the serial expansion. The parallel and serial problems consisting of 36, 54 and 180 activities are described in Golzarpoor [4].

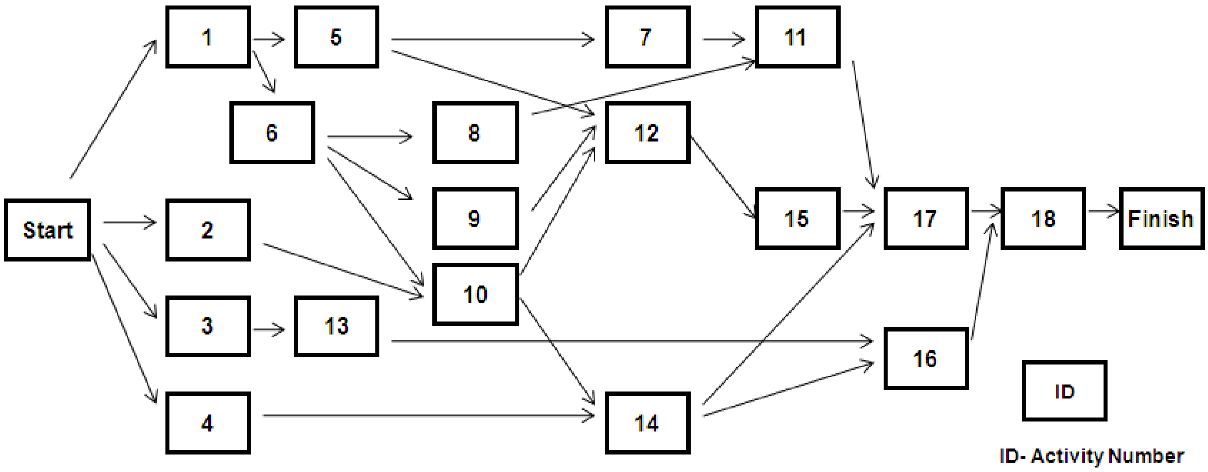

Figure 2: Project network of original problem with eighteen activities [6].

Approaches	Description	Deviation**	Criteria		Calculation Time
			Time	Cost ($)	
Optimal Solution	-	0%	110	216,270	-
Excel Solver*	Easy-to-use mathematical optimization tool	18%	110	254,620	2 min.
Risk Solver Platform Standard SLGRG Nonlinear*	Risk analysis, simulation, and optimization tools	0%	110	216,270	1.5 min.
Risk Solver Platform Standard Large-scale GRG Solver*	Risk analysis, simulation, and optimization tools	0%	110	216,270	1.5 min.
TCT Optimization Using Evolver (includes an evolutionary engine)*	GA-based optimization tool effective in optimizing complex and large-scale models	10%	110	238,070	30 min.
Risk Solver Platform Standard Evolutionary Solver*	Risk analysis, simulation, and optimization tools	27%	110	275,320	18 min.
Optimization Results using CPLEX CP Optimizer*	A solver for optimization based on constraint programming technique	0%	110	216,270	9 min.
Constraint Programming method using IBM ILOG Optimization Studio*	Constraint Programming method	0%	110	216,270	9 min.
This paper (RKTCO)	GA-based	0%	110	216,270	5 sec. for 50 generations

* Reported by Golzarpoor [4] **Percentage of deviation of the result from optimal solution

Table 1: Results for Project of 18 activities.

Table 2 shows the results for Project of 36 activities (parallel expansion). The Table 2 shows that the results of the RKTCO are the best when compared with the other methods reported by Golzarpoor [4].

Approaches	Description	Deviation**	Criteria		Calculation Time
			Time	Cost ($)	
Optimal Solution	-	0%	110	322,540	-
Excel Solver*	Easy-to-use mathematical optimization tool	27%	110	409,840	5 min
Risk Solver Platform Standard SLGRG Nonlinear*	Risk analysis, simulation, and optimization tools	20%	110	385,890	14 min
Risk Solver Platform Standard Large-scale GRG Solver*	Risk analysis, simulation, and optimization tools	18%	110	379,990	18 min
TCT Optimization Using Evolver (includes an evolutionary engine)*	GA-based optimization tool effective in optimizing complex and large-scale models	16%	109	373,790	30 min
Risk Solver Platform Standard Evolutionary Solver*	Risk analysis, simulation, and optimization tools	36%	110	438,440	15 min
Optimization Results using CPLEX CP Optimizer*	A solver for optimization based on constraint programming technique	0%	110	322,540	10 min
Constraint Programming method using IBM ILOG Optimization Studio*	Constraint Programming method	0%	110	322,540	10 min
This paper (RKTCO)	GA-based	0%	110	322,540	12 sec. 50generations

* Reported by Golzarpoor [3] **Percentage of deviation of the result from optimal solution

Table 2: Results for Project of 36 activities (parallel expansion).

Table 3 shows the results for Project of 36 activities (serial expansion). The RKTCO presents the best results when compared with the other methods reported by Golzarpoor [4].

Approaches	Description	Deviation**	Criteria		Calculation Time
			Time	Cost ($)	
Optimal Solution	-	0%	220	432,540	-
Excel Solver*	Easy-to-use mathematical optimization tool	0%	220	433,094	30 min
Risk Solver Platform Standard SLGRG Nonlinear*	Risk analysis, simulation, and optimization tools	11%	220	478,549	30 min
Risk Solver Platform Standard Large-scale GRG Solver*	Risk analysis, simulation, and optimization tools	19%	220	513,369	30 min
TCT Optimization Using Evolver (includes an evolutionary engine)*	GA-based optimization tool effective in optimizing complex and large-scale models	12%	220	484,640	30 min
Risk Solver Platform Standard Evolutionary Solver*	Risk analysis, simulation, and optimization tools	31%	219	566,640	1 min
Optimization Results using CPLEX CP Optimizer*	A solver for optimization based on constraint programming technique	0%	220	432,540	10 min
Constraint Programming method using IBM ILOG Optimization Studio*	Constraint Programming method	0%	220	432,540	10 min
This paper (RKTCO)	GA-based	0%	220	432,540	12 sec. 50generations

* Reported by Golzarpoor [4] **Percentage of deviation of the result from optimal solution

Table 3: Results for Project of 36 activities (serial expansion).

Table 4 shows the results for Project of 54 activities (parallel expansion) and the results of the RKTCO are the best when compared with the other methods reported by Golzarpoor [4].

Approaches	**Description**	**Deviation****	**Criteria**		**Calculation Time**
			Time	**Cost ($)**	
Optimal Solution	-	0%	110	428,810	-
Excel Solver*	Easy-to-use mathematical optimization tool	42%	110	607,435	30 min
Risk Solver Platform Standard SLGRG Nonlinear*	Risk analysis, simulation, and optimization tools	6%	110	454,198	26 min
Risk Solver Platform Standard Large-scale GRG Solver*	Risk analysis, simulation, and optimization tools	6%	110	454,198	26 min
TCT Optimization Using Evolver (includes evolutionary engine)*	GA-based optimization tool effective in optimizing complex and large-scale models	17%	110	500,610	30 min
Risk Solver Platform Standard Evolutionary Solver*	Risk analysis, simulation, and optimization tools	44%	110	618,260	1 min
Optimization Results using CPLEX CP Optimizer*	A solver for optimization based on constraint programming technique	0%	110	428,810	10 min
Constraint Programming method using IBM ILOG Optimization Studio*	Constraint Programming method	0%	110	428,810	10 min
This paper (RKTCO)	GA-based	0%	110	428,810	14 sec. 50generations

* Reported by Golzarpoor [4] **Percentage of deviation of the result from optimal solution

Table 4: Results for Project of 54 activities (parallel expansion).

Table 5 shows the results for Project of 54 activities (serial expansion). The RKTCO presents the best results when compared with the other methods reported by Golzarpoor [4].

Approaches	**Description**	**Deviation****	**Criteria**		**Calculation Time**
			Time	**Cost ($)**	
Optimal Solution	-	0%	330	648,810	-
Excel Solver*	Easy-to-use mathematical optimization tool	19%	330	769,460	13 min
Risk Solver Platform Standard SLGRG Nonlinear*	Risk analysis, simulation, and optimization tools	8%	330	698,851	30 min
Risk Solver Platform Standard Large-scale GRG Solver*	Risk analysis, simulation, and optimization tools	10%	330	714,551	30 min
TCT Optimization Using Evolver (includes an evolutionary engine)*	GA-based optimization tool effective in optimizing complex and large-scale models	8%	330	700,410	30 min
Risk Solver Platform Standard Evolutionary Solver*	Risk analysis, simulation, and optimization tools	57%	330	1.018,260	1 min
Optimization Results using CPLEX CP Optimizer*	A solver for optimization based on constraint programming technique	0%	330	648,810	10 min
Constraint Programming method using IBM ILOG Optimization Studio*	Constraint Programming method	0%	330	648,810	10 min
This paper (RKTCO)	GA-based	0%	330	648,810	15 sec. 50generations

* Reported by Golzarpoor [4] **Percentage of deviation of the result from optimal solution

Table 5: Results for Project of 54 activities (serial expansion).

Table 6 shows the results for Project of 180 activities (parallel expansion). The RKTCO presents the best results when compared with the other methods reported by Golzarpoor [4].

Approaches	Description	Deviation**	Criteria		Calculation Time
			Time	Cost ($)	
Optimal Solution	-	0%	110	1.172,700	-
Excel Solver*	Easy-to-use mathematical optimization tool	N/A	Too many variable cells		
Risk Solver Platform Standard SLGRG Nonlinear*	Risk analysis, simulation, and optimization tools	14%	110	1.336,900	30 min.
Risk Solver Platform Standard Large-scale GRG Solver*	Risk analysis, simulation, and optimization tools	35%	110	1.583,095	30 min.
TCT Optimization Using Evolver (includes an evolutionary engine)*	GA-based optimization tool effective in optimizing complex and large-scale models	54%	109	1.801,700	21 min.
Risk Solver Platform Standard Evolutionary Solver*	Risk analysis, simulation, and optimization tools	54%	110	1.807,000	21 min.
Optimization Results using CPLEX CP Optimizer*	A solver for optimization based on constraint programming technique	0%	110	1.172,700	13 min
Constraint Programming method using IBM ILOG Optimization Studio*	Constraint Programming method	0%	110	1.172,700	13 min
This paper (RKTCO)	GA-based	0%	110	1.172,700	180 sec. for 1000 generations

* Reported by Golzarpoor [4] **Percentage of deviation of the result from optimal solution

Table 6: Results for Project of 180 activities (parallel expansion).

From the above results it is clear that no algorithm dominates RKTCO. The CPLEX CP optimizer and Constraint Programming method using IBM ILOG Optimization Studio seems to have similar performance, but with higher computational time than RKTCO.

The proposed RKTVO has the best performance between all evolutionary–based algorithms (EA).

The time necessary by RKTCO to obtain the optimal solution is highly promising and shows that a good implementation can be critical to the success of the genetic algorithms.

This computational experience has been performed on a computer with an Intel Core 2 Duo CPU T7250 @2.33 GHz and 1,95 GB of RAM. The algorithm proposed in this work has been coded in VBA under Microsoft Windows NT.

5 CONCLUSIONS AND FURTHER RESEARCH

A new GA based-approach to solving the time-cost optimization for construction projects has been proposed. The project activities have various construction modes, which reflect different ways of performing the activity, each mode having a different impact on the duration and cost of the project. The present approach provides an attractive alternative for the solution of the construction multi-objective optimization problems. The developments made in this paper provide guidelines for designing and implementing practical GA applications in the civil engineering domain and gives to construction managers a tool to balance critical construction resources in the competitive construction industry.

Further research can be extended to others multiobjective problems with relationships between cost, time and quality in the construction industry.

ACKNOWLEDGMENTS

This work has been partially supported by the CIDEM (Centre for Research & Development in Mechanical Engineering). CIDEM is an investigation unit of FCT – Portuguese Foundation for the Science and Technology located at the School of Engineering of Polytechnic of Porto.

REFERENCES

[1] A. David, W. Coit, A. E. Smith, Multi-objective optimization using genetic algorithms: A tutorial, *Reliability Engineering & System Safety*, 91(9), 992–1007, 2006.

[2] A.Afshar, A. Ziaraty, A. Kaveh, F. Sharifi, Nondominated Archiving Multicolony Ant Algorithm in Time–Cost Trade-Off Optimization, *J. Constr. Eng. Manage*., 135(7), 668-674, 2009.

[3] A. Pagnoni, *Project engineering: computer-oriented planning and operational decision making*. Springer-Verlag, 1990.

[4] B. Golzarpoor, *Time-Cost Optimization of Large-Scale Construction Projects Using Constraint Programming*, Master Thesis, University of Waterloo, Ontario, Canada, 2012.

[5] C. Hendrickson, T. Au, *Project management for construction: fundamental concepts for owners, engineers, architects, and builders*, Prentice-Hall International Series in Civil Engineering and Engineering Mechanics, 1989.

[6] C.-W. Feng , L. Liu, S.A. Burns, Using genetic algorithms to solve construction time-cost trade-off problems, *J. Comp. Civ.Engrg.,* ASCE, 11(3), 184-189, 1997.

[7] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley: Reading, MA, 1989.

[8] D. R. Robinson, A dynamic programming solution to cost-time tradeoff for CPM, *Management Science*, 22(2), 158-166, 1975.

[9] D. X. M. Zheng, S. T. Ng, Stochastic time–cost optimization model incorporating fuzzy sets theory and nonreplaceable front, *Journal of Construction Engineering and Management*, ASCE, 131(2), 176-186, 2005.

[10] D. X.M. Zheng, S.T. Ng, M.M. Kumaraswamy, Applying a genetic algorithm-based multiobjective approach for time-cost optimization, *Journal of Construction Engineering and Management*, 130(2), 168-176, 2004.

[11] J.F. Gonçalves, J.J. M. Mendes, M.C.G. Resende. A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, 167, 77-95, 2005.

[12] J. Magalhães-Mendes, Project scheduling under multiple resources constraints using a genetic algorithm, *WSEAS Transactions on Business and Economics*, World Scientific and Engineering Academy and Society, USA, 11, 487-496, 2008.

[13] J. Magalhães-Mendes, A two-level genetic algorithm for the multi-mode resource-constrained project scheduling problem, *International Journal of Systems Applications, Engineering & Development*, 5(3), 271-278, 2011.

[14] J. Magalhães-Mendes, Active, Parameterized Active, and Non-Delay Schedules for Project Scheduling with Multi-Modes, *Proceedings of the 16th WSEAS International Conference on Applied Mathematics*, Montreaux, Switzerland, December 29-31, 134-139, 2011.

[15] J. Magalhães-Mendes, A hybrid genetic algorithm for the multi-mode resource-constrained project scheduling, *Proceedings of the 6th European Congress on Computational Methods in Applied Sciences and Engineering* (ECCOMAS 2012), Vienna University of Technology, Austria, e-Book Full Papers, 8871-8882, 2012.

[16] J. Magalhães-Mendes, *Multiobjective Approach for Time-Cost Optimization using a Multi-Mode Hybrid Genetic Algorithm*, In J. Magalhaes-Mendes and D. Greiner (Editors), Evolutionary Algorithms and Metaheuristics in Civil Engineering and Construction Management, Springer International Publishing, 39, 33-49, 2015.

[17] J. Magalhães-Mendes, A Comparative Study of Crossover Operators for Genetic Algorithms to Solve the Job Shop Scheduling Problem, *WSEAS Transactions on Computers*, World Scientific and Engineering Academy and Society, USA, 12(4), 164-173, 2013.

[18] J. Magalhães-Mendes, Time-Cost Optimization of Construction Projects: A Genetic Algorithm-Based Approach, in Y. Tsompanakis, J. Kruis, B.H.V. Topping, (Editors), *Proceedings of the Fourth International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, Civil-Comp Press, Stirlingshire, UK, Paper 11, 2015. doi:10.4203/ccp.109.11

[19] J. W. Fondahl, *A non-computer approach to the critical path method for the construction industry*, Technical Report No. 9, The Construction Institute, Department of Civil Engineering, Stamford University., 1961.

[20] L. Liu, S. Burns, C. Feng, Construction time-cost trade-off analysis using LP/IP hybrid method, *J. Constr. Eng. Manage.,* 121(4), 446-454, 1995.

[21] M. Gen, R. Cheng, *Genetic algorithms & engineering optimization*, Wiley-Interscience, New York, 2000.

[22] N. Siemens, A simple CPM time-cost trade-off algorithm, *Management Science*, 17(6), B354-B363, 1971.

[23] P. De, E.J. Dunne, J.B. Ghosh, C. E. Wells, The discrete time-cost trade-off problem revisited, *European Journal of Operational Research*, 81(2), 225-238, 1995.

[24] S. T. Ng, Y. Zhang, Optimizing construction time and cost using ant colony optimization approach, *Journal of Construction Engineering and Management*, ASCE, 134(9), 721-728, 2008.

[25] S.A. Burns, L. Liu, C. W. Feng, The LP/IP hybrid method for construction time-cost trade-off analysis, *Constr. Manag. Econ.,* 14(3), 265-276, 1996.

[26] T. Hegazy, Optimization of construction time-cost trade-off analysis using genetic algorithms, *Canadian Journal of Civil Engineering*, 26(6), 685-697, 1999.

[27] W. L. Mayer, L.R. Shaffer, Extension of the critical path method through the application of integer programming, *Civ. Engrg. Constr. Res. Ser. 2*, Univ. of Illinois, Urbana, III, 1963.

[28] Y. Xiong, Y. Kuang, Applying an ant colony optimization algorithm-based multiobjective approach for time–cost trade-off, *Journal of Construction Engineering and Management*, ASCE, 134(2), 153-156, 2008.

[29] Z. W. Geem, Multiobjective optimization of time-cost trade-off using harmony search, *Journal of Construction Engineering and Management*, ASCE, 136(6), 711-716, 2010.

[30] D. Greiner, J.M. Emperador, G. Winter, Multiobjective Optimisation of Bar Structures by Pareto GA, *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering* (ECCOMAS 2000), CIMNE, Barcelona, Spain, 2000.

[31] K. Deb, *Evolutionary Algorithms for Multicriterion Optimization in Engineering Design*. Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, J. Wiley & Sons, 1999.

[32] S. Parveen, S. K. Saha, GA Based Multi-Objective Time-Cost Optimization in a Project with Resources Consideration, *International Journal of Modern Engineering Research* (IJMER), 2(6), 4352-4359, 2012.

[33] C. C. Coello, An updated survey of GA-based multiobjective optimization techniques, *ACM Comput. Surv*., 32(2), 109–143, 2000.

[34] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.