A Real-coded Genetic Algorithm for Multiobjective Time-Cost Optimization

Jorge Magalhães-Mendes

***Abstract*—** This paper presents a new multiobjective optimization technique based on genetic algorithm for the time-cost construction problem. The chromosome representation of the problem is based on random keys. The schedules are constructed using a priority rule in which the priorities are defined by the genetic algorithm. Schedules are constructed using a procedure that generates parameterized active schedules. In construction projects, time and cost are the most important factors to be considered. In this paper, a new hybrid genetic algorithm is developed for the optimization of the two objectives time and cost. The results indicate that this approach could assist decision-makers to obtain good solutions for project duration and total cost.

***Keywords*—**Project management, Construction Management, Genetic Algorithms, Time-cost optimization.

I. Introduction

The Multiobjective Optimization Problem (also called multicriteria optimization, multiperformance or vector optimization problem) can then be defined (in words) as the problem of finding [37]:

"a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the decision maker."

The mathematical definition of a multiobjective problem (MOP) is important in providing a foundation of understanding between the interdisciplinary nature of deriving possible solution techniques (deterministic, stochastic); i.e., search algorithms [38].

More precisely, multiobjective problems (MOPs) are those problems where the goal is to optimize *k* objective functions simultaneously. This may involve the maximization of all *k* functions, the minimization of all *k* functions or a combination of maximization and minimization of these *k* functions [38].

J. Magalhães-Mendes is with the Civil Engineering Department, School of Engineering, Polytechnic of Porto, Porto, Portugal (e-mail: jjm@isep.ipp.pt).

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 [28]:

- 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 [39].

With evolutionary techniques 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 [3].

In principle, there is no clear definition of an ''optimum'' in multiobjective optimization (MOP) as in the case of single-objective issues; and there even does not necessarily have to be an absolutely superior solution corresponding to all objectives due to the incommensurability and conflict among objectives. Since the solutions cannot be simply compared with each other, the ''best'' solution generated from optimization would correspond to human decision-makers subjective selection from a potential solution pool, in terms of their particulars [10].

The classical methods reduce the MOP to a scalar optimization optimization by using multiobjective weighting (MOW) or a utility function (multiobjective utility analysis). Multiobjective weighting allows decisions makers to incorporate the priority of each objective into decision making. Mathematically, the solutions obtained by equally weighting all objectives may provide the least objective conflicts, but in most cases, each objective is first optimized separately and the overall objective value is evaluated depending on the weighting factors. The weakness of MOW is that the overall optimum is usually at the dominating objective only [6].

II. The Time-Cost Optimization Construction Problem

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, see Fig.1. Hence, relationship between project time and cost is trade-off [36].

Several approaches to solve the TCO problem have been proposed in the last years: mathematical, heuristic and search methods.

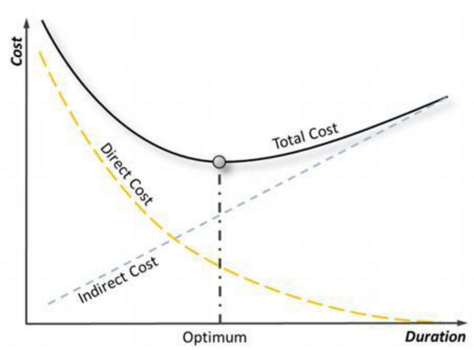

Fig. 1. Project time and cost curve.

A. Mathematical Methods

Several mathematical models such as linear programming (Kelley [12]; Hendrickson and Au [4]; Pagnoni [2]), integer programming, or dynamic programming (Butcher [33]; Robinson [8]; Elmaghraby [27]; De et al. [25]) and LP/IP hybrid (Liu et al. [21]; Burns et al. [29]), Meyer and Shaffer [31] and Patterson and Huber [14] 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]).

B. Heuristic Methods

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 [30]). However, they have lack mathematical consistency and accuracy and are specific to certain instances of the problem (Fondahl [19]; Prager [32]; Siemens [23] and Moselhi [24]) are some of the research studies that have utilized heuristic methods for solving TCO problems.

C. Search Methods

Some researchers have tried to introduce evolutionary algorithms to find global optima such as genetic algorithm (GA) (Feng et al. [6]; Gen and Cheng [22]; Zheng et al. [10]; Zheng and Ng [9]; the particle swarm optimization algorithm (Yang [11]), ant colony optimization (ACO) (Xiong and Kuang [34]; Ng and Zhang [29]; Afshar et al. [1]) and harmony search (HS) (Geem [36]).

In a certain way we can say that the work of Zadeh [20] is the first to advocate the assignment of weights to each objective function and combined them into a single-object function. More recently, Gen and Cheng [22] adopted the adaptive weight approach (AWA) in construction TCO problem (also referred to as GC approach hereafter).

In the GC approach Gen and Cheng [22] proposed the following formulas:

$$Z^{+} = \; Z_c^{\max}, Z_t^{\max} \tag{1}$$

$$Z^{-} = \; Z_c^{\min}, Z_t^{\min} \tag{2}$$

where,

$Z_c^{\max}$ = maximal value for total cost in the current population;

$Z_t^{\max}$ = maximal value for time in the current population;

$Z_c^{\min}$ = minimal value for total cost in the current population;

$Z_t^{\min}$ = minimal value for time in the current population.

$$w_c = 1/(Z_c^{\max} - Z_c^{\min}),\; w_t = 1/(Z_t^{\max} - Z_t^{\min}) \tag{3}$$

$$f(x) = w_c\,(Z_c^{\max} - Z_c) + w_t\,(Z_t^{\max} - Z_t) \tag{4}$$

In 2004, Zheng et al. [10] proposed the modified weight approach (MAWA) to deal with the multi-objective problem.

Under the MAWA, the adaptive weights are formulated through the following four conditions:

1) For $Z_t^{\max}$ is not equal to $Z_t^{\min}$ and $Z_c^{\max}$ is not equal to $Z_c^{\min}$

$$v_c = \frac{Z_c^{\min}}{Z_c^{\max} - Z_c^{\min}} \tag{5}$$

$$v_t = \frac{Z_t^{\min}}{Z_t^{\max} - Z_t^{\min}} \tag{6}$$

$$v = v_c + v_t \tag{7}$$

$$w_c = v_c / v \tag{8}$$

$$w_t = v_t / v \tag{9}$$

$$w_c + w_t = 1 \tag{10}$$

2) For $Z_t^{\max} = Z_t^{\min}$ and $Z_c^{\max} = Z_c^{\min}$

$$w_c = w_t = 0.5 \tag{11}$$

3) For $Z_t^{\max} = Z_t^{\min}$ and $Z_c^{\max} \neq Z_c^{\min}$

$$w_c = 0.1, \quad w_t = 0.9 \tag{12}$$

4) For $Z_t^{\max} \neq Z_t^{\min}$ and $Z_c^{\max} = Z_c^{\min}$

$$w_c = 0.9, \quad w_t = 0.1 \tag{13}$$

Zheng et al. [10] proposed a fitness formula in accordance with the proposed adaptive weight:

$$f(x) = w_t \frac{(Z_t^{\max} - Z_t) + \gamma}{(Z_t^{\max} - Z_t^{\min}) + \gamma} + w_c \frac{(Z_c^{\max} - Z_c) + \gamma}{(Z_c^{\max} - Z_c^{\min}) + \gamma} \tag{14}$$

where,

γ is a small positive random number between 0 and 1.

$Z_c^{\max}$ = maximal value for total cost in the current population;

$Z_t^{\max}$ = maximal value for time in the current population;

$Z_c^{\min}$ = 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 x[th] solution in current population;

Z_t = represents the time of the x[th] solution in current population.

This study uses the fitness formula proposed by Gen and Cheng [22] 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 x[th] solution in current chromosome;

Z_t = represents the time of the x[th] solution in current chromosome.

III. The GA-based approach

The approach presented in this paper is based on a genetic algorithm to perform its optimization process. Fig. 2 shows the architecture of 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.

For each chromosome the following four phases are applied:

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.

After a schedule is obtained, the quality is processed feedback to the genetic algorithm. Fig. 2 illustrates the sequence of phases applied to each chromosome. Details about each of these phases will be presented in the next sections.

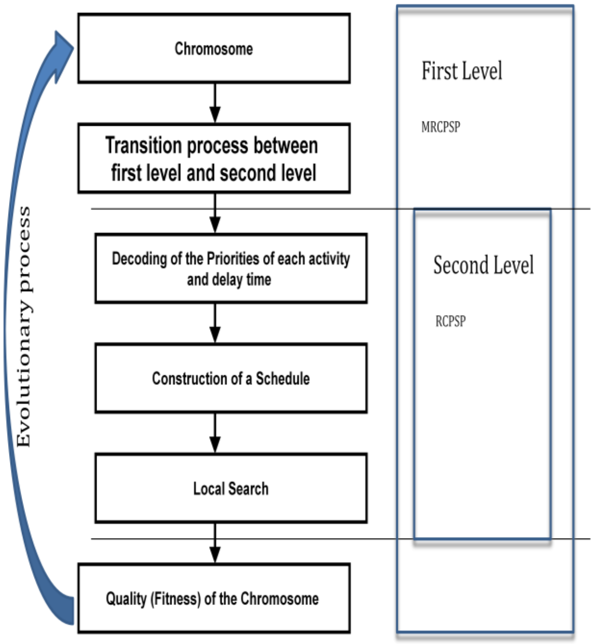

Fig. 2. Architecture of the approach.

A. GA Transition Process

The Genetic Algorithms (GAs) are search algorithms which are based on the mechanics of natural selection and genetics to search through decision space for optimal solutions. One fundamental advantaged of GAs from traditional methods is described by Goldberg [7]: in many optimization methods, we move gingerly from a single solution in the decision space to the next using some transition rule to determine the next solution.

First of all, an initial population of potential solutions (individual) is generated randomly. A selection procedure based on a fitness function enables to choose the individual candidate for reproduction. The reproduction consists in recombining two individuals by the crossover operator, possibly followed by a mutation of the offspring. Therefore, from the initial population a new generation is obtained. From this new generation, a second new generation is produced by the same process and so on. The stop criterion is normally based on the number of generations.

The GA based-approach uses a random key alphabet U (0, 1) and an evolutionary strategy identical to the one proposed by Goldberg [7].

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 $mn+n$ genes where n is the number of activities and m is the number of execution modes, see Fig. 3.

The called first level as the capacity to solving the multi-mode resource constrained project scheduling problem (MRCPSP) [16, 18].

In this case of study we do not consider the requirements to the type and number of resources needed for construction mode for each activity as well as the maximum number of available resources.

The transition process between first level and second level consists in choosing the option or construction mode m_j for each activity j. Using this process we obtain the solution chromosome (second level) composed by 2n genes, see Fig.4.

The called second level as the capacity to solving the resource constrained project scheduling problem (RCPSP) [16, 18].

In this case of study we do not consider the requirements to the type and number of resources needed for each activity as well as the maximum number of available resources.

Activity 1	Mode 1	Gene $_{11}$
	Mode 2	Gene $_{12}$
	…	…
	Mode m	Gene $_{1m}$
	Delay 1	Gene $_{1m+1}$
Activity 2	Mode 1	Gene $_{21}$
	Mode 2	Gene $_{22}$
	…	…
	Mode m	Gene $_{2m}$
	Delay 2	Gene $_{2m+1}$
…		…
Activity n	Mode 1	Gene $_{n1}$
	Mode 2	Gene $_{n2}$
	…	…
	Mode m	Gene $_{nm}$
	Delay n	Gene $_{nm+1}$

Fig. 3. Chromosome structure.

After, we evaluate the quality (fitness) of the solution chromosome.

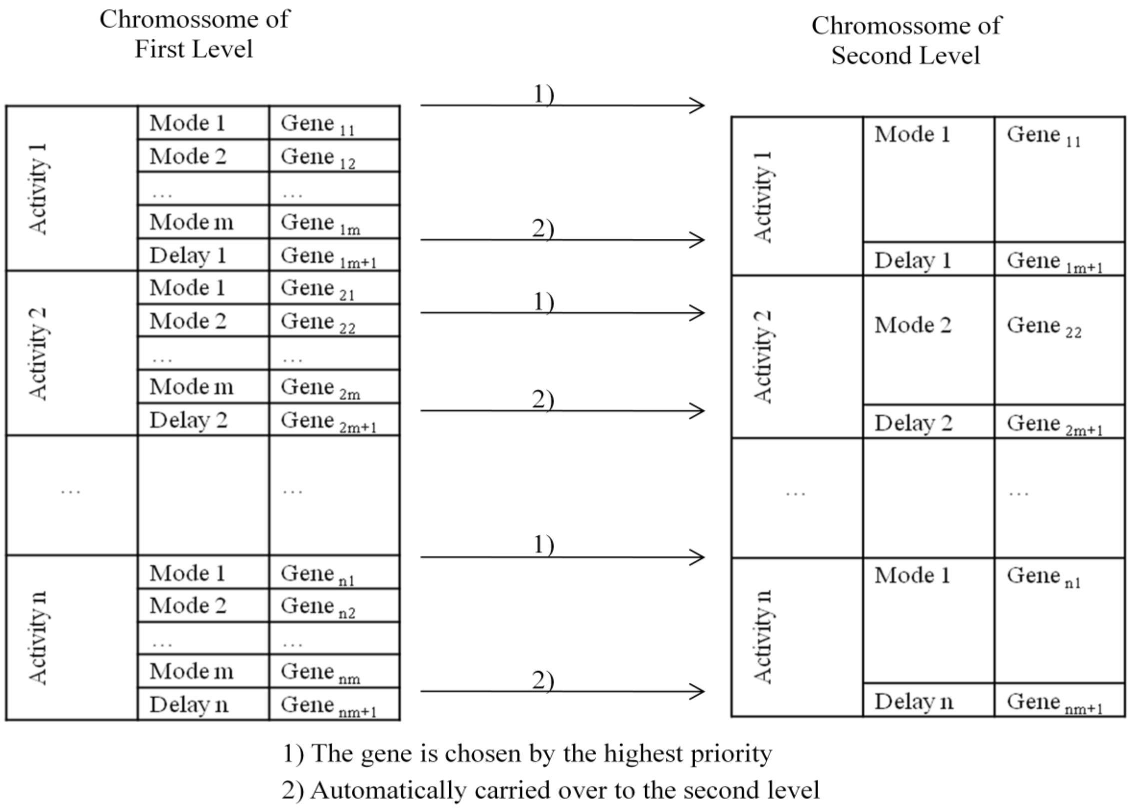

Fig. 4. Transition process between first and second level.

B. GA Decoding

A real-coded GA is adopted in this paper. Compared with the binary-code GA, the real-coded GA has several distinct advantages, which can be summarized as follows (Y.-Z. Luo et al. [35]):

- It is more convenient for the real-coded GA to denote large scale numbers and search in large scope, and thus the computation complexity is amended and the computation efficiency is improved;
- The solution precision of the real-coded GA is much higher than that of the binary-coded GA;
- As the design variables are coded by floating numbers in classical optimization algorithms, the real-coded GA is more convenient for combination with classical optimization algorithms.

The priority decoding expression uses the following expression:

$$PRIORITY_j = \frac{LLP_j}{LCP} \times \left[\frac{1+gene_{mj}}{2}\right] \quad j = 1,\dots,n \qquad (15)$$

where,

[1] LLP_j is the longest length path from the beginning of the activity j to the end of the project;

[2] LCP is the length along the critical path of the project [15];

[3] m_j is the gene of the selected mode for activity j.

The gene jm+1 is used to determine the delay time when scheduling the activities. The delay time used by each activity is given by the following expression:

$$Delay\,time = gene_{jm+1} \times 1.5 \times MaxDur \qquad (16)$$

where *MaxDur* is the maximum duration of all activities. The factor 1.5 is obtained after some experimental tuning.

A maximum delay time equal to zero is equivalent to restricting the solution space to non-delay schedules and a maximum delay time equal to infinity is equivalent to allowing active schedules. To reduce the solution space is used the value given by formula (16), see Gonçalves et al. [13].

C. Construction of a Schedule

Schedule generation schemes (SGS) are the core of most heuristic solution procedures for project scheduling. SGS start from scratch and build a feasible schedule by stepwise extension of a partial schedule.

There are two different classics methods SGS available.

They can be distinguished into activity and time incrementation. The so called serial SGS performs activity-incrementation and the so called parallel SGS performs time-incrementation.

A third method for schedule generating can be applied: the parameterized active schedules. 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. If the predefined period is set to zero, then we obtain a non-delay schedule. This type of SGS is used on this work.

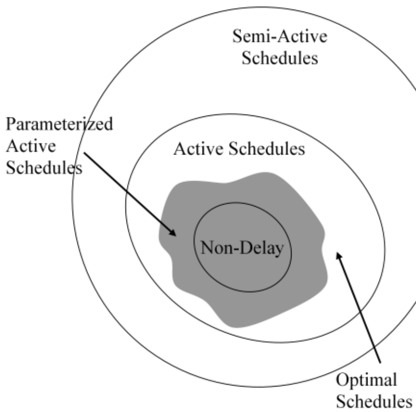

Fig. 5. Types of schedules (adapted from Mendes [18]).

Fig. 5 presents the relationship diagram of various schedules with regard to optimal schedules.

D. Local Search

Local search algorithms move from solution to solution in the space of candidate solutions (the search space) until a solution optimal or a stopping criterion is found. In this paper it is applied backward and forward improvement based on Klein [27].

Initially it is constructed a schedule by planning in a forward direction starting from the project's beginning. After it is applied backward and forward improvement trying to get a better solution. The backward planning consists in reversing the project network and applying the scheduling generator scheme. A detailed example is described by Mendes [15].

E. Evolutionary Strategy

There are many variations of genetic algorithms obtained by altering the reproduction, crossover, and mutation operators. Reproduction is a process in which individual (chromosome) is copied according to their fitness values (makespan). Reproduction is accomplished by first copying some of the best individuals from one generation to the next, in what is called an elitist strategy.

In this paper 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,

$$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \quad , \qquad i = 1, \ldots, n \tag{17}$$

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 [7].

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 Fig. 6.

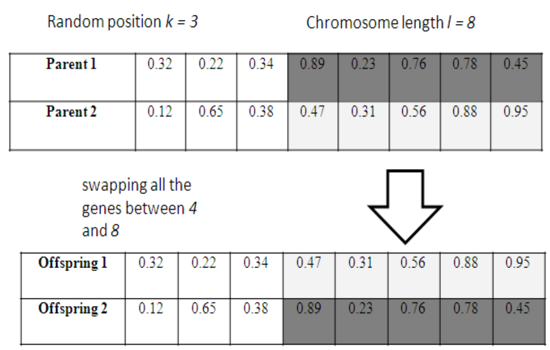

Fig. 6. Crossover example (adapted from Mendes [40]).

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%.

F. GA Configuration

Though there is no straightforward way to configure the parameters of a genetic algorithm, we obtained good results with values: population size of 5 × number of activities in the problem; mutation probability of 0.05; top (best) 1% from the previous population chromosomes are copied to the next generation; stopping criterion of 50 generations.

IV. Case Study

In order to compare the proposed RKV-TCO (Random Key Variant for Time-Cost Optimization) approach, a case study of seven activities proposed initially by Liu et al. [21] was used.

A project of seven activities proposed initially by Liu et al. [21] and fitted by Zheng et al. [10] is presented in Table 1 with available activity options and corresponding durations and costs. Indirect cost rate was $1500/day.

Table 1 Time and cost for each option/mode of activity.

Activity description	Activity number	Precedent activity	Option/ Mode	Duration (days)	Direct cost ($)
Site preparation	1	-	1	14	23,000
			2	20	18,000
			3	24	12,000
Forms and rebar	2	1	1	15	3,000
			2	18	2,400
			3	20	1,800
			4	23	1,500
			5	25	1,000
Excavation	3	1	1	15	4,500
			2	22	4,000
			3	33	3,200
Precast concrete girder	4	1	1	12	45,000
			2	16	35,000
			3	20	30,000
Pour foundation and piers	5	2, 3	1	22	20,000
			2	24	17,500
			3	28	15,000
			4	30	10,000
Deliver PC girders	6	4	1	14	40,000
			2	18	32,000
			3	24	18,000
Erect girders	7	5, 6	1	9	30,000
			2	15	24,000
			3	18	22,000

The robustness of the new proposed model RKV-TCO in the deterministic situation was compared with two other previous models:

1) Gen and Cheng [22] using GC approach;
2) Zheng et al. [10] using MAWA with a GA-based approach.

The results of RKV-TCO approach are presented in Table 2. The Table 2 shows the values of time and cost for the first six generations with Gen and Cheng [22] and Zheng et al. [10] approaches. The algorithm RKV-TCO obtains in the third generation a better solution than the works mentioned above. So, the RKV-TCO ends with project time = 63 days and cost =$225,500 in Table 2.

Additionally we can also state that the RKV-TCO approach produces high-quality solutions quickly once needed only 3 seconds to complete 50 generations.

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.

V. Conclusions and Further Research

A GA based-approach to solving the time-cost optimization problem 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 chromosome representation of the problem is based on random keys. The schedules are constructed using a priority rule in which the priorities are defined by the genetic algorithm. The present approach provides an attractive alternative for the solution of the construction multi-objective optimization problems.

Further research can be extended to the following directions: extended to more construction project problems to reinforce the results obtained namely expanding the optimization model to consider resource allocation and resource leveling constraints and expanding the number of modes of construction for each activity to turn a more complicated optimization problem.

Table 2 Summary of the results.

Approaches	Generation number	Criteria		Calculation Time
		Time	Cost ($)	
Gen and Cheng [22]	0	83	243,500	Not reported
	1	80	242,400	
	2	80	261,900	
	3	79	256,400	
	4	79	256,400	
	5	79	256,400	
Zheng et al. [10]	0	73	251,500	Not reported
	1	73	251,500	
	2	73	251,500	
	3	66	236,500	
	4	66	236,500	
	5	66	236,500	
This paper	0	73	233,000	3 (two) seconds for 50 generations
	1	68	225,500	
	2	**63**	**225,500**	

ACKNOWLEDGMENT

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

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

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

[4] C. Hendrickson and 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.

[5] C. Patrick, *Construction Project Planning and Scheduling*, Pearson Prentice Hall, Ohio, 2004.

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

[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), 1975, pp.158-166.

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

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

[11] I.T. Yang, “Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis”, *Journal of Construction Engineering and Management, ASCE*, 133(7), 2007, pp.498-505.

[12] J. E. Kelley, “Critical-Path Planning and Scheduling: Mathematical Basis”. *Operations Research*, 9(3), 1961, pp. 296-320.

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

[14] J. H. Patterson and D. Huber, “A horizon-varying, zero-one approach to project scheduling”, *Management Science*, 20(6), 1974, pp. 990-998.

[15] J. Magalhaes-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, Vol. 11, 2008, pp.487-496.

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

[17] J. Magalhaes-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, 2011, pp-134-139.

[18] J. Magalhaes-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, 2012, pp. 8871-8882.

[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. A. Zadeh, “Fuzzy sets”, *Inf. Control*, 8, 1965, pp.338-353.

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

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

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

[24] O. Moselhi, “Schedule compression using the direct stiffness method”, *Canadian Journal of Civil Engineering*, 20(1), 1993, pp.65-72.

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

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

[27] R. Klein, “Bidirectional planning: improving priority rule-based heuristics for scheduling resource constrained projects”, *European Journal of Operational Research*, Vol. 127, 2000, pp.619-638.

[28] S. E. Elmaghraby, “Resource allocation via dynamic programming in activity networks”, *Eur. J. Operational Res.*, Vol. 64, 1993, pp. 199-215.

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

[30] T. Hegazy, "Optimization of construction time-cost trade-off analysis using genetic algorithms", *Canadian Journal of Civil Engineering*, 26(6), 1999, pp. 685-697.
[31] W. L. Mayer and 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.
[32] W. Prager, "A structured method of computing project cost polygons", *Management Science*, 9(3), 394–404, 1963.
[33] W.S. Butcher, "Dynamic programming for project cost-time curve", *Journal of Construction Division, ASCE*, 93(C01), 1967, pp. 59-73.
[34] Y. Xiong and 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), 2008, pp. 153-156.
[35] Y.-Z. Luo, G.-J.Tang, Z.-G. Wang and H.-Y. Li, "Optimization of perturbed and constrained fuel-optimal impulsive rendezvous using a hybrid approach", *Engineering Optimization*, 38(8), 2006, pp.959-973.
[36] Z. W. Geem, "Multiobjective optimization of time-cost trade-off using harmony search". *Journal of Construction Engineering and Management*, ASCE, Vol. 136, No. 6, 2010, pp.711-716.
[37] A. Osyczka. "*Multicriteria optimization for engineering design". In J. S. Gero, editor, Design Optimization, Academic Press, 1985, pp. 193-227.
[38] C. A. Coello Coello, D. A. Van Veldhuizen and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, 2002.
[39] 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.
[40] J. Magalhaes-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, Vol. 4, 2013, pp.164-173.

J. Magalhães-Mendes was born in Mancelos (Amarante, Portugal) on January 17, 1963.

He has the following academic degrees:

PhD in Mechanical Engineering and Industrial Management by University of Oporto; M.Sc. in Civil Engineering by University of Aveiro; M.Sc. in Systems and Automation by University of Coimbra; Degree in Civil Engineering by Polytechnic of Oporto and Degree in Applied Mathematics by University of Oporto.

He has been Coordinator Professor of the School of Engineering of Polytechnic of Oporto since January of 2010, where he teaches the courses of organization and management of works and construction management. He has published papers in the European Journal of Operational Research, Computers & Operations Research, Journal of Heuristics, WSEAS/NAUN Journals, book chapters and several national and international conferences. He has about 400 ISI citations. His research interest includes construction management, project management, genetic algorithms, and operational research and supply chain management.