MEMETIC ALGORITHM TO MINIMIZE COSTS OF ACTIVITIES IN RCPSP

ALGORITMO MEMÉTICO PARA LA MINIMIZACIÓN DE COSTO DE ACTIVIDADES EN RCPSP

JAIRO R. MONTOYA-TORRES
Ph.D., Profesor Asociado, Universidad de La Sabana, jairo.montoya@unisabana.edu.co

ELYN L. SOLANO-CHARRIS
M.Sc. Profesor Asistente, Universidad de La Sabana

CONSUELO DURÁN-CANTOR
Ingeniero Industrial, Facultad de Ingeniería, Universidad de La Sabana

Received for review January 27 th, 2012, accepted May 8th, 2012, final version May, 24 th, 2012

ABSTRACT: The resource-constrained project scheduling problem (RCPSP) is a classic and important problem in project management and has gotten much attention from researchers. In this paper, we consider the case in which the cost of activities has to be minimized. This variant of the problem can be found mainly in construction management supply chains. Its characteristics show that this objective function is non-regular and that the problem is NP-hard. This paper presents a memetic algorithm to solve this problem. Computational experiments are carried out using well-known instances from the PSPLIB. Results obtained validate the effectiveness of the proposed algorithm and allow us to understand the algorithm's behavior.

KEYWORDS: Project scheduling, Activity cost, Memetic algorithm.

RESUMEN: La programación de proyectos con recursos restringidos (RCPSP, por su nombre en inglés) es un problema clásico e importante en la gerencia de proyectos, y que ha recibido mucha atención de parte de los investigadores. Este artículo considera el caso en el cual se desea minimizar el costo de las actividades. Esta variante del problema se encuentra principalmente en la gerencia de cadenas de suministro para proyectos de construcción. Sus características muestran que la función es no-regular y que el problema es NP-completo. Este artículo presenta un algoritmo memético pare resolver el problema. Experimentos computacionales son ejecutados empleando las famosas instancias de la librería PSPLIB. Los resultados obtenidos validan la efectividad y eficiencia del algoritmo propuesto.

PALABRAS CLAVE: Programación de proyectos, Costo de actividades, Algoritmo memético.

1. INTRODUCTION

In the late 50s, the development of PERT (project evaluation and review technique) and CPM (critical path method) techniques allowed projects to be portrayed by network diagrams where jobs or activities are represented by arcs, events are represented by nodes, and the inter-relations between the jobs or activities are defined by the network structure [1]. However, project scheduling with these techniques deals only with the time aspect without consideration of resource restrictions and/or cash flows. Yet, in many real-life situations, delays in the execution time of certain activities occur when resources required by these activities are not available in sufficient quantities during the time interval when they are scheduled to be executed. This particular problem is known in the literature as the resource-constrained project scheduling problem (RCPSP) [2].

The RCPSP is a classic and important problem in project management that has attracted the attention of many researchers. The performance measures of RCPSPs include time-based objectives (i.e.,

minimizing makespan, tardiness, or completion time) and cost-based objectives (maximizing net present value or minimizing total cost of activities including the costs of resource consumption, overhead, and tardiness penalties) [3,4]. According to the complexity theory, the RCPSP is known to be NP-hard [5], which means that it is not possible to find optimal solutions for large-sized instances in a reasonable computational time. The approaches to solve the RCPSPs are classified in the literature into two categories: exact and approximate. Exact (optimal) approaches are integer programming models, implicit enumeration with branch and bound, and dynamic programming. These approaches are able to solve small-size instances. For medium- to large-sized instances, approximate procedures are preferred—such as heuristic and meta-heuristic algorithms. State-of-the-art surveys can be found in [6–11].

In the literature we can find that there has been abundant research on RCPSPs, but very little attention has been given to the case of minimizing the cost of activities in projects. This variant of the problem can be found mainly in construction management supply chains [12]. Only the works of Liu and Wang [12] and Liu and Zheng [4] have considered this problem by proposing genetic algorithms for the single- and multiple-project case, respectively. As the reader can observe in such references, we do regret that the work of Liu and Zheng [4]did not propose an extended experimental study, while the work of Liu and Wang [12]carried out a computational experiment for only one project. This deficiency makes it very difficult to compare newly proposed algorithms against theirs.

Since genetic algorithms have demonstrated themselves to be good resolution procedures for various types of RCPSPs [2], it is worth noting that the combination of both population-based and trajectory-based meta-heuristics might allow for a better resolution procedure for RCPSPs. In addition, since memetic algorithms have shown themselves to be efficient and effective when solving different types of planning, scheduling, and timetabling problems [13], this paper proposes the use of a memetic algorithm to solve the RCPSP with minimization of the activities' cost.

This paper is organized as follows. Section 2 describes formally and in detail the problem under study. Section 3 is devoted to the presentation of the proposed memetic algorithm. The results of computational experiments are presented in Section 4. The paper ends in Section 5 with some concluding remarks.

2. PROBLEM DESCRIPTION

The RCPSP with activity costs can be described as follows [12]: A single project consists of a set of $j = 1,...,J$ activities that have to be executed without preemption. Each activity has a duration of d_j units of time. Because of technological requirements, there are precedence relations between some of the activities enforcing that an activity $j = 2,...,J$ may not be started before all its immediate predecessors $i \in P_j$ (with P_j being the set of immediate predecessors of activity j) have been finished. In the description of the activity-on-node (AON) method, the structure of the project is a graph where the nodes representing the activities and the arcs represent the precedence relations between activities. The network is acyclic and numerically labeled, where an activity always has a higher label than its predecessors. Without loss of generality, we can assume that 1 is the only start activity and J is the only finish activity. There are K-types of renewable resources, and activity j needs r_{jk} units of resource $k \hat{I} K$ during every period of its duration. The static cost of activity j is noted as v_j. Let Fj be the finish time of activity j, and A_t the set of activities being in progress in period t. Let us set h as the cost per period of holding completed activities stated as a percent of fixed activity cost. The capacity of resource k is noted at R_k and the unit price is C_k. The due date of the project is noted as H. With a given H, we can get the earliest and latest finish times (respectively, e_j and l_j) of activity j by using the critical path method (CPM). The time parameters in the problem are assumed to be all integer and non-negative. V_t is the holding cost for completed activities of the project through period t.

The objective function, as shown by Eq. (1), seeks to minimize the total cost of project activities by including two items. The first item is the sum of the holding cost of the completed activities from the start time of the project to its finish time (i.e., the dynamic cost related with the activities in the project. The second term is the amount of the activities' static cost:

$$\min \sum_{t=1}^{F_J} V_t + \sum_{j=1}^{J} v_j \quad (1)$$

During the resolution process, several constraints have to be satisfied. For instance, the precedence relations, the termination of the project before its due date, and the use of renewal resources according to their per-period availabilities should be satisfied.

With a given resource supplier already known, the capacities and unit prices of its resources are decided beforehand [12]. Hence, v_j in Eq. (1) can be seen as a constant. Therefore, the later item in objective function (1) is fixed no matter how the project is scheduled. As a result, the focus will be placed on the dynamic cost of activities. This function is non-regular [12] according to the definition proposed in [14]. In addition, as pointed out by Liu and Wang [12], the feasible solutions of the problem are the same as that of the traditional RCPSP with makespan minimization, so that the RCPSP with the objective of minimizing the activities' cost is also NP-hard. In order to solve this problem, we propose a heuristic procedure that combines strategies from both population-based and trajectory-based meta-heuristics. The proposed memetic algorithm allows for one to obtain efficient solutions for RCPSPs with a minimization of the cost of the activities.

3. THE MEMETIC ALGORITHM

Memetic algorithms (MA) are population-based meta-heuristics, which means that the algorithm maintains a *population* of candidates for the problem at hand (i.e., a pool comprising several tentative solutions) [13]. In the terminology of evolutionary computation, each of these candidate solutions is called *individual*; however, the term *agent* is more appropriate for the nature of MA since the former term suggests a passive entity simply subject to evolutionary rules, while the latter term implies the existence of an active behavior reflected in several components of the algorithm such as (but not exclusively as) local search adds-on. One of the most distinctive components of MA is the use of local improvers, which are used as a mechanism for improving solutions on a local (and even autonomous) basis.

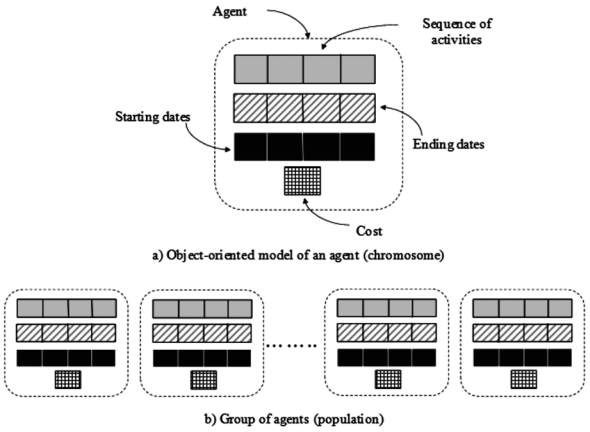
The proposed memetic algorithm will be named MAPS/Cost in which "MAPS" stands for memetic algorithm for project scheduling and "Cost" means that the objective function is the minimization of cost. This section presents the main issues regarding the design of the proposed MA model:

- **Generation of initial population.** Schedule generation schemes (SGS) are an integral part of heuristic procedures for resource-constrained project scheduling problems [15]. Such schemes, when building a solution, work sequentially from the beginning by generating a partial solution. At the end, when the set of candidate activities to be scheduled is empty, the SGS has built a feasible sequence. There are two types of schemes to build a solution: the serial one focused in the schedule generation by increasing the sequenced activities, and the parallel scheme, focused on schedule generation by increasing time. The serial scheme (SSGS) is used in the proposed algorithm. It is composed of n phases, with n being the number of activities to be scheduled. Step by step, each activity is selected and scheduled by taking into account precedence constraints and resource availability. This scheme works from two sets of disjunctive activities: the set of already-scheduled activities and the set of activities that can be scheduled [8]. This last set is characterized by the fact that in each phase it can incorporate activities that were not in it in the precedent phase.

- **Solution representation.** Since the algorithm can be coded using object-oriented programming, the following classes are needed: the class activity containing the job identification (ID), its duration (processing time), the sets of successors and predecessors, the processing starting time, and the level of resource utilization. In order to compute the solution, some temporary variables are needed: activity cost, temporal start, and ending time (which are used to compute, respectively, the [temporal] start and ending times of each activity during the execution of the algorithm). The agent is composed of two attributes: an array containing the sequence of activities and the cost corresponding to the value of the objective function. Finally, the following parameters are needed to be defined by the user: the size of the initial population, crossover

and mutation probabilities, terminating condition, number of generations, type of crossover (one point or two points), and type of cost (cumulative or per activity). Figure 1 presents the structure of the agent. The representation of the chromosome for the RCPSP studied here is based on the concept of *activity list.* This allows coding the sequence through a feasible list of precedence relationships between activities. An activity list has the form . The SSGS can be optimally used to decode an activity list and to obtain the sequence [2].

Figure 1. Structure of an agent, adapted from [2]



- **New generation.** Figure 2 presents the main routing (cycle) of the proposed memetic algorithm. This routine of the algorithm generates sons until obtaining two times the agents in the population. Two parents are randomly selected from the current population. If the probability of performing crossover is achieved, then crossover operators are applied. Otherwise, parents are duplicated and become sons. Mutation operator is applied using a swap (interchange), as explained next. Sons that do not comply with due dates are considered as unfeasible and are therefore eliminated from the population. Finally, 50% of the population having the worst value of the objective function is eliminated.

- **Crossover operator.** Crossover operators applied into the proposed algorithm can be of three types: one-point crossover, two-point crossover, and hybrid. The one-point operator randomly selects a point (the same) for both parents. Activities before that point are selected from parent 1, and those after the point are selected from parent 2. The two-point operator works in a similar manner. Two-points are randomly selected for both parents. Activities before the first point are selected from parent 1; those between the two points are selected from parent 2; and finally, the activities after the second point are selected from parent 1. The hybrid crossover assigns a 0.5 probability of performing one-point crossover and a 0.5 probability of performing the two-point crossover. For the purpose of this study, we tested the performance of the MA by comparing its behavior using the three operators.

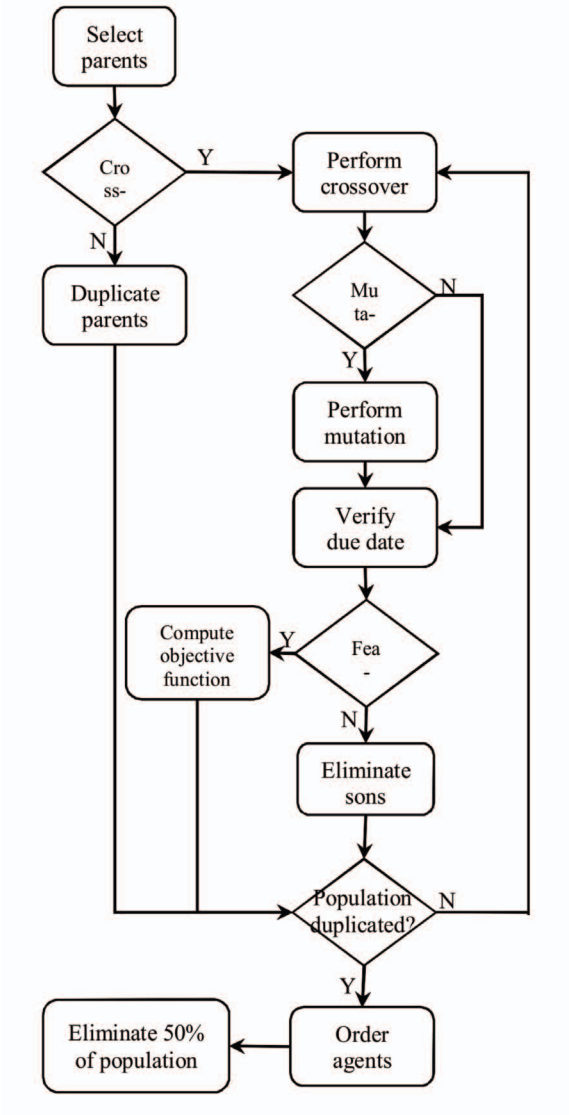


Figure 2. Main routine of the proposed memetic algorithm

- **Mutation operator.** This operator seeks to introduce genetic diversity into the current population. The strategy selected for our algorithm is that of interchanging the genes of a whole chromosome. Genes are chosen randomly

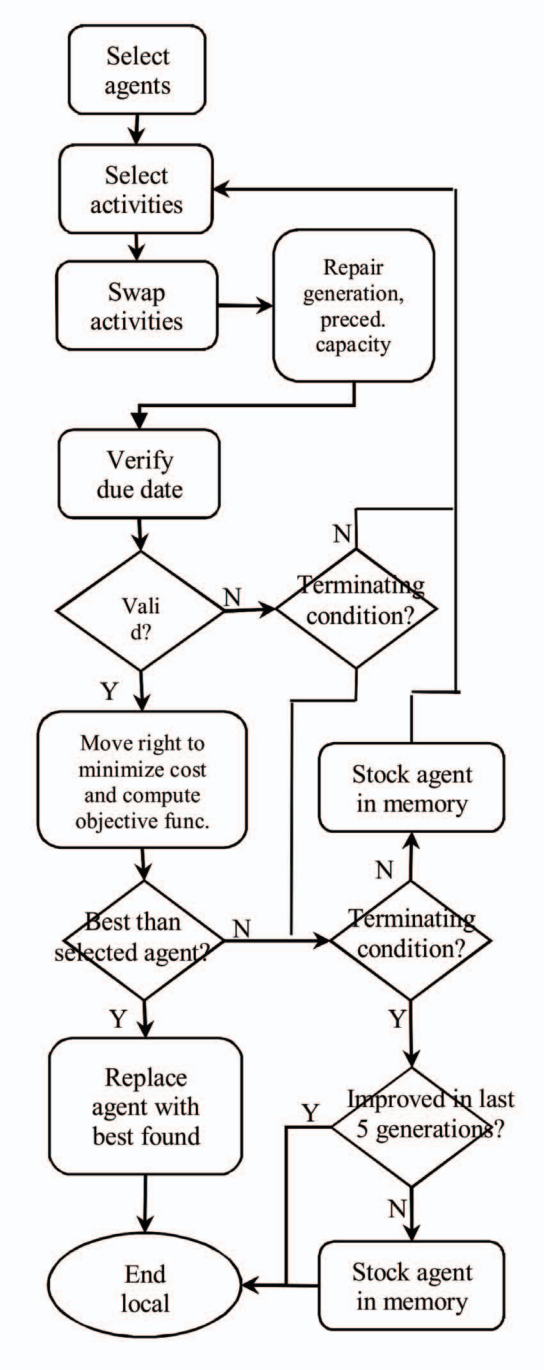


Figure 3. Main routine of the local search procedure

- **Local search.** Figure 3 shows a flow diagram illustrating how local search improvement is performed. First, an agent is selected at random. Then two activities are selected randomly and are interchanged. The next process (generation repair, precedence, and capacity) is in charge of repairing the newly-created agents from the selected parents by respecting precedence and capacity constraints. The tasks corresponding to "Compute Start Time and End Time", "Verify Due date", and "Move Right to Compute" that compute the objective function were explained before. If any improvement is found, the original agent is replaced by the best one found; otherwise those agents are temporarily stored in the memory. Finally, the agent is replaced by a randomly-selected newly-created agent.

4. EXPERIMENTS

The algorithm was programmed using JAVA. For the experiments, all instances from the Project Scheduling Problem Library (PSPLIB) available on Internet at the web page http://129.187.106.231/psplib/ were taken. This is a general-instance library widely used for testing algorithms for RCPSPs. In our experiment, the library was updated in order to include the costs of activities, according to the structure proposed in [12] in which the static cost of each activity and the due dates of each project are given. We considered the whole sets of instances in the library. The sets J30, J60, and J90 correspond to 480 instances with 30, 60, and 90 activities, respectively. The set J120 contains 600 project instances with 120 activities per project. This gives us a total of 2,040 problem instances. In order to run the algorithm, the conditions presented in Table 1 have been defined. The number of generations was set to be 100 for all experiments. In addition to the combination of factors presented in Table 1, experiments with 50 and 200 generations (iterations) were also carried out.

The analysis will be carried out only on the behavior of the proposed memetic algorithm. Tables 2 and 3 present the summary of results for experiments on data sets J30, J60, J90, and J120. During execution, the proposed MA showed convergence. We can observe that a good improvement is obtained between the first and last solution. For some instances, the algorithm found a good solution from the very first iteration.

Table 1. Factors and levels for the experiments

Factors	Levels
Data sets from PSPLIB	J30, J60, J90; J120
Population size	100, 150
Crossover probability	0.9, 0.75, 0.7, 0.5
Mutation probability	0.05, 0.2, 0.1, 0.3
Crossover strategy	one-point, two-points, mixed
Terminating criterion	10, 20
Number of generations	100

As explained previously, the problem under study here has been considered very little in the literature. Only the works of Liu and Wang [12] and Liu and Zheng [4] have considered it by proposing genetic algorithms for the single- and multiple-project case, respectively. The work of Liu and Zheng [4] does not propose an extended experimental study, and hence it is not possible to compare our proposed memetic algorithm against theirs. The work of Liu and Wang [12] carries out a computational experiment for only one project. Hence, it is not possible for us to compare our procedure against those of the literature by means of a rigorous experimental study. Since we are presenting results for an extended computational experiment using standard instances available on the Internet in the PSPLIB, we do hope that the results presented here will become a reference for further researchers in order to improve our results.

Table 2. Summary of results of cumulative cost (average values) with population size 100 for data sets J90 and J120 (the objective value at the first iteration is in parentheses)

Crossover	Stop criterion	Pmut	Results for Data Set J30				Results for Data Set J60			
			Pcross				Pcross			
			0.9	0.75	0.7	0.5	0.9	0.75	0.7	0.5
One-point	10	0.05	1810.64 (1815.60)				2257.56 (2721.22)			
		0.2		1810.64 (1810.64)				2557.56 (2743.31)		
		0.1			1810.64 (1810.64)				2564.24 (2748.54)	
		0.3				1810.64 (1810.64)				2557.56 (2814.24)
	20	0.05	1810.64 (1810.64)				2557.56 (2756.39)			
		0.2		1810.64 (1810.64)				2557.56 (2723.26)		
		0.1			1810.64 (1810.64)				2557.56 (2707.85)	
		0.3				1810.64 (1810.64)				2550.58 (2745.06)
Two-point	10	0.05	1810.64 (1810.64)				2557.56 (2734.01			
		0.2		1810.64 (1826.24)				2557.56 (2737.5)		
		0.1			1810.64 (1915.60)				2560.17 (2747.96)	
		0.3				1810.64 (1815.60)				2557.56 (2803.19)
	20	0.05	1810.64 (1917.73)				2557.56 (2768.89)			
		0.2		1810.64 (1810.64)				2557.56 (2597.67)		
		0.1			1810.64 (1814.18)				2564.24 (2687.79)	
		0.3				1810.64 (1810.64)				2557.56 (2746.51)

Crossover	Stop criterion	Pmut	Results for Data Set J30				Results for Data Set J60			
			Pcross				Pcross			
			0.9	0.75	0.7	0.5	0.9	0.75	0.7	0.5
Mixed	10	0.05	1810.64 (1814.18)				2557.56 (2798.26)			
		0.2		1810.64 (1810.64)				2557.56 (2614.24)		
		0.1			1810.64 (1810.64)				2557.56 (2757.27)	
		0.3				1810.64 (1810.64)				2505.81 (2720.64)
	20	0.05	1810.64 (1810.64)				2557.56 (2691.86)			
		0.2		1810.64 (1810.64)				2557.56 (2744.48)		
		0.1			1810.64 (1810.64)				2557.56 (2749.41)	
		0.3				1810.64 (1810.64)				2564.24 (2717.15)

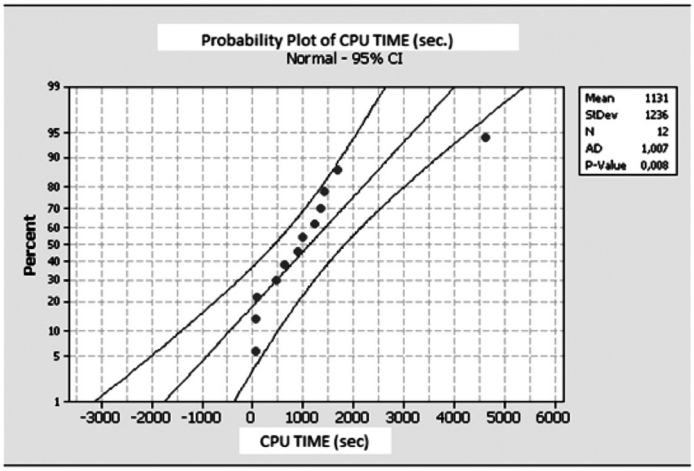
Concerning the execution time, Fig. 4 presents a plot diagram of the average CPU time for each instance of the experiment. From the figure, we can observe that resolution time is highly variable, depending on the instance size. The average time is 1.131 s, with a 95% confidence interval of (431.7, 1830.3) s, and a coefficient of variability of 109.2%. These values clearly confirm the high variability of the computational time.

Table 3. Summary of results of cumulative cost (average values) with population size 100 for data sets J90 and J120 (the objective value at the first iteration is in parentheses)

Crossover	Stop criterion	Pmut	Results for Data Set J90				Results for Data Set J120			
			Pcross				Pcross			
			0.9	0.75	0.7	0.5	0.9	0.75	0.7	0.5
One-point	10	0.05	3213.95				4058.69			
			(3310.08)				(4546.87)			
		0.2		3213.95				4058.27		
				(3304.84)				(4676.08)		
		0.1			3213.95				4072.88	
					(3372.29)				(4741.31)	
		0.3				3213.95				4084.84
						(3375.97)				(4521.97)
	20	0.05	3213.95				4032.68			
			(3288.95)				(4524.89)			
		0.2		3213.95				4088.46		
				(3265.50)				(4841.17)		
		0.1			3213.95				4031.57	
					(3275.39)				(4452.85)	
		0.3				3213.95				4059.80
						(3333.91)				(4600.97)

Crossover	Stop criterion	Pmut	Results for Data Set J90				Results for Data Set J120			
			Pcross				Pcross			
			0.9	0.75	0.7	0.5	0.9	0.75	0.7	0.5
Two-point	10	0.05	3213.95				4083.45			
			(3463.95				(4559.39			
		0.2		3213.95				4041.59		
				(3279.07)				(4493.18)		
		0.1			3213.95				4151.04	
					(3296.12				(4587.06)	
		0.3				3213.95				4132.54
						(3393.41)				(4431.57)
	20	0.05	3213.95				4052.16			
			(3293.22)				(4337.55)			
		0.2		3213.95				4053.96		
				(3239.15)				(4308.76)		
		0.1			3213.95				4070.37	
					(3369.96)				(4426,70)	
		0.3				3213.95				4100.42
						(3285.46)				(4706.54)
Mixed	10	0.05	3213.95				4071.49			
			(3308.14)				(4611.54)			
		0.2		3213.95				4052.16		
				(3253.29)				(4424.89)		
		0.1			3213.95				4043.67	
					(3280.43)				(4517.94)	
		0.3				3113.95				4044.92
						(3178.29)				(4788.32)
	20	0.05	3213.95				4030.74			
			(3267.05)				(4474.55)			
		0.2		3213.95				4092.90		
				(3290.12)				(4325.31)		
		0.1			3213.95				4090.82	
					(3350.97)				(4613.21)	
		0.3				3213.95				4071.63
						(3371.70)				(4458.27)

Figure 4. Plot diagram of the execution time



5. CONCLUDING REMARKS

This paper considered the problem of minimizing activities' cost in a resource-constrained project scheduling problem. An example of such a problem in a real-life situation is the construction supply chain. Since this objective function is non-regular and the problem is NP-hard, a memetic algorithm was proposed in this paper. The idea of analyzing the effectiveness of this procedure is based on the success it has had when applying the combination of strategies from both population-based and trajectory-based meta-heuristics to solve other scheduling- and combinatorial-optimization problems [13]. Computational experiments were carried out using the well-known instances from the PSPLIB, in order to analyze the impact of the algorithm's parameters in the value of the objective function. Further work is to compute the optimal values and then to compare the relative performance of the proposed MA with other procedures from the literature. Other approaches can consider the use of sophisticated heuristics such as those based on discrete-event simulation (see, for example, [16]) or even the use of evolutionary multi-objective procedures, that have shown themselves to be efficient with solving multi-objective optimization problems [17], in order to consider, for example, cost minimization and on-time project completion. According to the review of Yang et al. [18], such works could consider regular- and non-regular objective functions.

ACKNOWLEDGMENTS

This works was supported by the *Universidad de La Sabana* under grants CEA-24-2008 and CEA-53-2010 and was developed under the framework of CYTED-HAROSA@IB Thematic Research Network.

REFERENCES

[1] Hutchings, J., Project Scheduling Handbook. Marcel Dekker, 2003.

[2] Montoya-Torres, J.R., Gutierrez-Franco, E., Pirachicán-Mayorga, C., Project scheduling with limited resources using a genetic algorithm. International Journal of Project Management, 28 (6), pp. 619-628, 2010.

[3] Kolisch, R., Padman, R., An Integrated Survey of Deterministic Project Scheduling. Omega, 3, pp. 249-272, 2001.

[4] Liu, Z., Zheng, Y., Resource-Constrained Multiple Projects Scheduling with the Objective of Minimizing Activities Cost. Proceedings of the 2008 Control and Decision Conference, pp. 1027-1032, 2008.

[5] Blazewicz, J., Lenstra, J.K., Rinnooy Kan, A., Scheduling subject to resource constraints: classification and complexity. Discrete Applied Mathematics, 5, pp. 11-24, 1983.

[6] Lee, J.K., Kim, Y.D., Search Heuristics for Resource-Constrained Project Scheduling. Journal of the Operational Research Society, 3, pp. 678-689, 1996

[7] Brucker, P., Drexl, A., Mohring, R., Neumann, K., Pesch, E., Resource-constrained project scheduling: notation, classification, models, and methods. European Journal of Operational Research, 112, pp. 3-41, 1999

[8] Kolisch, R., Hartmann, S., Heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis. Handbook on recent advances in project scheduling, J. Weglar (ed.), pp. 147-78. Kluwer Academic Publishers, Dordrecht, 1999.

[9] Hartmann, S., Kolisch, R., Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. European Journal of Operational Research, 127, pp. 394-407, 2000.

[10] Hartmann, S., Briskorn, D., A Survey of Variants and Extensions of the Resource-Constrained Project Scheduling Problem. European Journal of Operational Research, 207 (1), pp.1-14, 2010.

[11] Weglarz, J., Józefowska, J, Mika, M., Waligóra, G., Project scheduling with finite or infinite number of activity processing modes - A survey. European Journal of Operational Research, 208 (3), pp. 177-205, 2011.

[12] Liu, Z., Wang, H., GA-Based Resource-Constrained Project Scheduling with the Objective of Minimizing Activities' Cost. Lecture notes in Computers Science, D.S. Huang and X.-P. Zhang and G.-B. Huang (eds.), Volume 3644, pp. 937–946. Springer-Verlag, Berlin Heidelberg, 2005.

[13] Cotta, C., Fernández, A.J. Memetic Algorithms in Planning, Scheduling, and Timetabling. Studies in Computational Intelligence, 49, pp. 1-30, 2007.

[14] Sprecher, A., Kolisch, R., Drexl, A., Semi-active, Active, and Non-delay Schedules for the Resource-Constrained Project Scheduling Problem. European Journal of Operational Research, 80 (1), pp. 94-102, 1995.

[15] Ballestin, F., Valls, V. and Quintanilla, S., Pre-emption in resource-constrained project scheduling. European Journal of Operational Research, 189, pp. 1136-1152, 2008

[16] Montoya-Torres, J.R., Soto-Ferrari, M. and González-Solano, F., Production scheduling with sequence-dependent setups and job release dates. DYNA, 77 (163), pp. 260-269. 2010

[17] Frutos, M., Tohmé, F., Evolutionary multi-objective scheduling procedures in non-standardized production processes. DYNA, 79 (172), pp. 101-107, 2012.

[18] Yang, B., Geunes, J., O'brien, W.J., Resource-Constrained Project Scheduling: Past Work and New Directions. Research Report, Department of Industrial and Systems Engineering, University of Florida, Gainsville, USA.