A Two-level Genetic Algorithm for the Multi-Mode Resource-Constrained Project Scheduling Problem

J. Magalhães-Mendes

***Abstract*—** This paper presents a genetic algorithm for the multi-mode resource-constrained project scheduling problem (MRCPSP), in which multiple execution modes are available for each of the activities of the project. The objective function is the minimization of the construction project completion time. To solve the problem, is applied a two-level genetic algorithm, which makes use of two separate levels and extend the parameterized schedule generation scheme by introducing an improvement procedure. It is evaluated the quality of the schedule and present detailed comparative computational results for the MRCPSP, which reveal that this approach is a competitive algorithm.

***Keywords*—** Construction Management, Project Scheduling, Multi-mode RCPSP, Resource Constraints, Genetic Algorithms.

I. INTRODUCTION AND BACKGROUND

A construction project is a group of discernible tasks or activities that are conducted in a coordinated effort to accomplish one or more objectives. Construction projects require varying levels of cost, time and other resources (i.e., labor, equipment, material, suppliers), Patrick [57].

Construction projects are found throughout business and areas such as manufacturing facilities, infrastructure development and improvement, and residential and commercial building (as shown in Figure 1).

The project schedule is the core of the project plan. It is used by the project manager to commit people to the project and show the organization how the work will be performed.

No matter the size or scope of your project, the schedule is a key part of project management. The schedule tells you when each activity should be done, what has already been completed, and the sequence in which things need to be finished.

Schedules also help you do the following:

1. They provide a basis for you to monitor and control project activities;
2. They help you determine how best to allocate resources so you can achieve the project goal;
3. They help you assess how time delays will impact the project;
4. You can figure out where excess resources are available to allocate to other projects;
5. They provide a basis to help you track project progress.

A project can be depicted by a graph where the activities are numerically numbered. Associated with each activity is a set of possible durations with specific resource requirements. If resources are available in limited quantities each time period, the resources are considered renewable (e.g., machines or manpower).

As the number of project activities increases and thus the complexity of their sequential ordering, the need for organized planning and scheduling increases too. This need further increases when a large number of project activities are considered relative to the uniqueness of each construction project in terms of the dynamic plant and nonstandardized nature of the work. So, finding feasible schedules which efficiently use scarce resources is a challenging task within project management. In this context, the well-known Resource Constrained Project Scheduling Problem (*RCPSP*) has been studied during the last decades, see e.g. [4, 5, 6, 7, 8, 32, 33, 59].

In the classical *RCPSP*, the activities of a project have to be scheduled so that the makespan of the project is minimized. So, the technological precedence constraints have to be observed as well as limitations of the renewable resources required to accomplish the activities. Once started, an activity may not be interrupted.

This problem has been extended to a more realistic model, the multi-mode resource constrained project scheduling problem (*MRCPSP*), where each activity can be performed in one out of several modes. Each mode of an activity represents an alternative way of combining different levels of resource requirements with a related duration. Each renewable resource has a limited availability such as manpower and machines for the entire project.

The objective of the *MRCPSP* problem is minimizing the *makespan*. While the exact methods are available for providing optimal solution for small problems, its computation time is not feasible for large-scale problems. Hence, in

J. Magalhães-Mendes is with the School of Engineering of Polytechnic of Oporto, Rua Antonio Bernardino de Almeida, 431, 4200-072 Porto Portugal (corresponding author to provide phone: 00351228340500; fax: 00351228321159; e-mail: jjm@ isep.ipp.pt).

practice heuristic and metaheuristics methods to generate near-optimal solutions for large problems are of special interest.

Figure 1: Commercial building project.

Several approaches to solve the MRCPSP have been proposed in the last years:

Exact procedures: Talbot [12] was the first to present an enumeration scheme for solving the problem. Patterson et al. [13] proposed an enumerative type of branch and bound algorithm based on the generation of a precedence tree that guides the search for solutions. Speranza and Vercellis [38] proposed a depth-first branch-and-bound algorithm, but Hartmann and Sprecher [39] have shown that this algorithm may be unable to find the optimal solution for instances with two or more renewable resources. Sprecher et al. [40], Hartmann and Drexl [41] and Sprecher and Drexl [42] presented a branch-and-bound algorithm. Demeulemeester et al. [43] presented a depth-first branch-and-bound procedure for the discrete time/resource trade-off problem.

Heuristics procedures: Talbot [12] and Sprecher and Drexl [42] proposed to use a time limit on their exact branch-and-bound procedure. Boctor [21] tested 21 heuristic scheduling rules and suggested a combination of 5 heuristics which have a high probability of giving the best solution. Drexl and Grunewald [19] proposed a biased random sampling approach, while Ozdamar and Ulusoy [44] proposed a local constraint based analysis approach. Boctor [23] presented a heuristic algorithm based on the Critical Path Method computation, Kolisch and Drexl [25] suggested a local search method with a single-neighborhood search, Lova et al. [53] proposed heuristics based on priority rules and Knotts et al. [45] evaluated different agent-based algorithms for solving the MRCPSP.

Metaheuristics procedures: evolutionary algorithms have been presented by Vaca [22], Mori and Tseng [24], Ozdamar [46], Hartmann [28], Alcaraz et al. [47], Lorenzoni et al. [35], Lova et al. [30], Damak et al. [29], Mendes [54] and Pan and Yeh [58]. Slowinski et al. [48], Jozefowska et al. [49] and Bouleimen and Lecocq [50] used the simulated annealing approach. Zhang et al. [51] proposed the particle swarm optimization methodology for solving the MRCPSP.

This paper introduces a new genetic algorithm approach for solving the MRCPSP based on the work proposed by Mendes [54].

Extending this approach, we develop a phenotype which consists of a random key vector with genes for all modes for all activities and for delay time. The basic idea is to choice one mode for each activity and converts the phenotype for multi-mode into a phenotype for a single-mode. Using the best mode for each activity the solution is generated using a parameterized scheduling scheme for a single-mode RCPSP. A local search procedure is applied trying to improve the initial solution. For the evolutionary process the genetic algorithm uses the phenotype for multi-mode.

II. PROBLEM FORMULATION

The MRCPSP problem can be stated as follows. A project consists of $n+2$ activities where each activity has to be processed in order to complete the project. Let $J = \{0, 1, \dots, n, n+1\}$ denote the set of activities to be scheduled and $K = \{1, \dots, k\}$ the set of renewable resources. Each resource type k has a limited capacity of R_k at any point in time.

The activities 0 and $n+1$ are dummy, have no duration and represent the initial and final activities. The activities are interrelated by two kinds of constraints:

1. The precedence constraints, which force each activity j to be scheduled after all predecessor activities, P_j, are completed.
2. Performing the activities requires resources with limited capacities.

Each activity can be performed in one of several different modes. A mode represents a combination of different resources and/or levels of resources with an associated duration. Once an activity is started in one mode, it may not be changed. One activity j can be executed in m modes given by the set $\boldsymbol{M}_j = \{1, \dots, m_j\}$. The duration of activity j being performed in mode m_j is given by d_{jm}. The activity j executed in mode m_j uses r_{jmk} units of renewable resource k, where $r_{jmk} <= R_k$ for each renewable resource k.

While being processed, activity j requires r_{jmk} units of resource type $k \in K$ during every time instant of its non-preemptable duration d_{jm}. The parameters d_{jm}, r_{jmk} and R_k are assumed to be non-negative and deterministic.

The problem depends on finding a schedule of the activities, taking into account the resources and the precedence constraints, which minimize the *makespan* ($C_{\max}$).

Let F_j represent the finish time of activity j. A schedule can be represented by a vector of finish times ($F_1, \dots, F_m, \dots, F_{n+1}$). The *makespan* of the solution is given by the maximum of all predecessors activities of activity $n+1$, i.e. $F_{n+1} = Max_{l \in P_{n+1}} \{F_l\}$.

A mathematical programming formulation of this problem has been given by Talbot [12].

III. TYPES OF SCHEDULES

Classifying schedules is the basic work to be done before attacking scheduling problems. Schedules can be classified into one of the following three types of schedules:

i) *Semi-active schedules*. These are feasible schedules obtained by sequencing activities as early as possible. In a semi-active schedule the start time of a particular activity is constrained by the processing of a different activity on the same resource or by the processing of the directly preceding activity on a different resource.

ii) *Active schedules*. These are feasible schedules in which no activity could be started earlier without delaying some other activity or breaking a precedence constraint. Active schedules are also semi-active schedules. An optimal schedule is always active.

iii) *Non-delay schedules*. These are feasible schedules in which no resource is kept idle at a time when it could begin processing some activity. Non-delay schedules are active and hence are also semi-active.

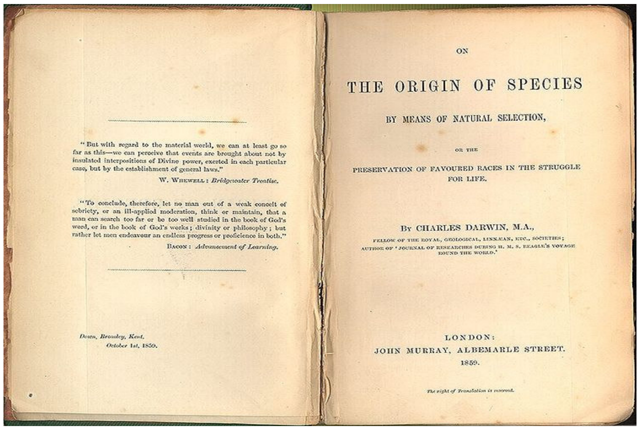
In this work are generated active schedules. The constructive heuristic used to construct active schedules is based on a parameterized active generation scheme.

IV. GENETIC ALGORITHM

The approach presented in this paper is based on a genetic algorithm to perform its optimization process.

Genetic algorithms (GAs) are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search [1].

The GAs follows the principles of *The Origin of Species* proposed by Charles Darwin [60], see Figure 2.



"But with regard to the material world, we can at least go so far as this—we can perceive that events are brought about not by insulated interpositions of Divine power, exerted in each particular case, but by the establishment of general laws."

W. Whewell: *Bridgewater Treatise*.

"To conclude, therefore, let no man out of a weak conceit of sobriety, or an ill-applied moderation, think or maintain, that a man can search too far or be too well studied in the book of God's word, or in the book of God's works; divinity or philosophy; but rather let men endeavour an endless progress or proficience in both."

Bacon: *Advancement of Learning*.

Down, Bromley, Kent,
October 1st, 1859.

ON
THE ORIGIN OF SPECIES
BY MEANS OF NATURAL SELECTION,
OR THE
PRESERVATION OF FAVOURED RACES IN THE STRUGGLE FOR LIFE.

By CHARLES DARWIN, M.A.,
FELLOW OF THE ROYAL, GEOLOGICAL, LINNÆAN, ETC., SOCIETIES;
AUTHOR OF 'JOURNAL OF RESEARCHES DURING H. M. S. BEAGLE'S VOYAGE ROUND THE WORLD.'

LONDON:
JOHN MURRAY, ALBEMARLE STREET.
1859.

The right of Translation is reserved.

Figure 2: The Origin of Species.

One fundamental advantaged of GAs from traditional methods is described by Goldberg [1]: 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. This solution-to-solution method is dangerous because it is a perfect prescription for locating false peaks in multimodal search spaces. By contrast, GAs work from a rich database of solutions simultaneously (a population of chromosomes), climbing many peaks in parallel; thus the probability of finding a false peak is reduced over methods that go solution to solution.

First of all, an initial population of potential solutions (individuals) is generated randomly. A selection procedure based on a fitness function enables to choose the individuals 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 general schema of GAs may be illustrated as follows, see Figure 3.

```
procedure GENETIC-ALGORITHM

Generate initial population $P_0$;
  Evaluate population $P_0$;
  Initialize generation counter g ← 0;
  While stopping criteria not satisfied repeat
    Select some elements from $Pg$ to copy into $Pg+1$;
      Crossover some elements of $Pg$ and put into $Pg+1$;
      Mutate some elements of $Pg$ and put into $Pg+1$;
      Evaluate some elements of $Pg$ and put into $Pg+1$;
      Increment generation counter: g ← g+1;
   End while

End GENETIC-ALGORITHM;
```

Figure 3: Pseudo-code of a genetic algorithm.

A. Decoding

The genetic algorithm uses a random key alphabet U (0, 1) and an evolutionary strategy identical to the one proposed by Goldberg [1].

A 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* is made of *mn*+1 genes where n is the number of activities (first level):

$$\textbf{Chromosome} = (\underbrace{\textbf{gene}_{11}, \textbf{gene}_{21}, \dots, \textbf{gene}_{m1}}_{\text{Genes for each mode of activity } 1}, \dots, \underbrace{\textbf{gene}_{1n}, \textbf{gene}_{2n}, \dots, \textbf{gene}_{mn}}_{\text{Genes for each mode of activity } n}, \underbrace{\textbf{gene}_{mn+1}}_{\text{Gene for delay time}})$$

For each activity *j* we have a set of execution modes:

$$\textbf{Activity}_{\,1} = (\textbf{gene}_{11}, \textbf{gene}_{21}, \ldots, \textbf{gene}_{m1})$$
$$\textbf{Activity}_{\,2} = (\textbf{gene}_{12}, \textbf{gene}_{22}, \ldots, \textbf{gene}_{m2})$$
$$\ldots\ldots\ldots\ldots$$
$$\textbf{Activity}_{\,n} = (\textbf{gene}_{1n}, \textbf{gene}_{2n}, \ldots, \textbf{gene}_{mn})$$

After the choice of the execution mode m_j for each activity *j*, the solution chromosome is composed by $n+1$ genes (second level).

The priority decoding expression uses the following expression:

$$\boldsymbol{PRIORITY}_j = \frac{\boldsymbol{LLP}_j}{\boldsymbol{LCP}} \times \left[\frac{1+\boldsymbol{gene}_{m_j}}{2}\right] \qquad \boldsymbol{j}=1,\ldots,\boldsymbol{n} \qquad (1)$$

where,

$\boldsymbol{LLP_j}$ is the longest length path from the beginning of the activity *j* to the end of the project
$\boldsymbol{LCP}$ is length along the critical path of the project, see Mendes [5].
m_j is the gene of the selected mode for activity *j*.

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

$$\boldsymbol{Delay\,time} = \boldsymbol{gene}_{mn+1} \times 1.5 \times \boldsymbol{MaxDur} \qquad (2)$$

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 expression (2), see Mendes [55].

B. *A Two-Level Genetic Algorithm*

The genetic algorithm presented in this paper is based on a two-level mechanism.

The first level is composed by the chromosome with all genes to solve the initial problem – MRCPSP.

For each *initial chromosome*, we must select only one gene by activity. With a gene by activity we must solve the *solution chromosome*, that is, the *RCPSP*, see Figure 4.

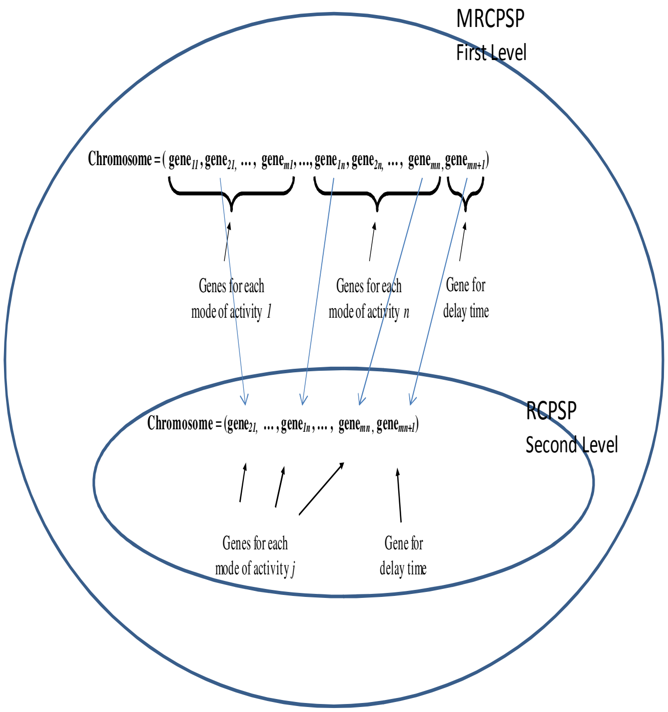


Figure 4: A Two-Level Genetic Algorithm.

C. *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,

$$\boldsymbol{p}_i = \frac{\boldsymbol{f}_i}{\sum_{i=1}^{N} \boldsymbol{f}_i} \quad , \qquad \boldsymbol{i}=1,\ldots,\boldsymbol{n} \qquad (3)$$

An example is presented in Table 1.

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 [1], see Figure 5. After selection the mating population consists of the chromosomes (individuals): 1, 2, 3, 4, 5 and 6.

Table 1: Selection probability and fitness value.

Number of chromosome	Fitness value	Selection probability
1	14	0,20
2	12	0,17
3	10	0,14
4	9	0,13
5	8	0,11
6	7	0,10
7	4	0,06
8	3	0,04
9	2	0,03
10	1	0,01

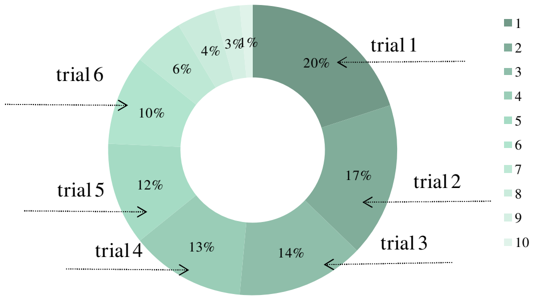
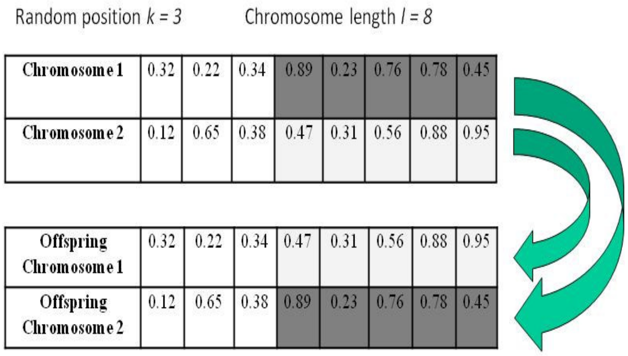


Figure 5: Roulette-wheel selection.

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* [1], see Figure 6.

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 0.001. With 200 genes positions we should expect 200 x 0.001 = 0.2 genes to undergo mutation for this probability value.



Chromosome 1	0.32	0.22	0.34	0.89	0.23	0.76	0.78	0.45
Chromosome 2	0.12	0.65	0.38	0.47	0.31	0.56	0.88	0.95
Offspring Chromosome 1	0.32	0.22	0.34	0.47	0.31	0.56	0.88	0.95
Offspring Chromosome 2	0.12	0.65	0.38	0.89	0.23	0.76	0.78	0.45

Figure 6: Crossover operator example.

V. SCHEDULE GENERATIONS SCHEMES

Schedule generation schemes (SGS) are the core of most heuristic solution procedures for the *RCPSP*. SGS start from scratch and build a feasible schedule by stepwise extension of a partial schedule. A partial schedule is a schedule where only a subset of the *n*+2 activities have been scheduled. 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 [37].

The constructive heuristic used to construct active schedules is based on a parameterized active schedules generation scheme [55].

VI. 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 was applied backward and forward improvement based on Klein [34].

Initially 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. An example is described by Mendes [5].

VII. COMPUTATIONAL EXPERIMENTS

This section presents results of the computational experiments done with the algorithm proposed in this paper. This algorithm is called RKV-AS-MM (*Random Key Variant Active Schedules for Multi-Mode*). This computational experience has been performed on a computer with an Intel Core 2 Duo CPU T7250 @2.00 GHz. The algorithm proposed in this work has been coded in Visual Basic 6.0 under Microsoft Windows NT.

A. *Genetic algorithm 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 0.001; **top** (best) 1% from the previous population chromosomes are copied to the next generation; **stopping criterion** of 50 generations.

B. *Experimental results*

In what follows, it is compared the RKV-AS-MM with others approaches reported in the literature. Table 2 lists the eleven chosen approaches and the problem (project) instances are those used by [18] and [22]. The RKV-AS-MM and each of the eleven chosen approaches are applied individually to each chosen problem (single-mode), [18]. The final project durations generated by the RKV-AS-MM are compared with the corresponding results generated by each one of the eleven approaches, see Table 3. The results of the RKV-AS-MM are the same (optimal values) as the procedures proposed by Vaca [22] and RKV-MM [54]).

Table 2: The approaches for testing single-mode instances.

Rule	**Heuristic**
TR	Technological ranking
EST	Ascendant Earliest Start Times
EFT	Ascendant Earliest Finish Times
LST	Ascendant Latest Start Times
LFT	Ascendant Latest Finish Times
Total Float	Ascendant total Float
ACTIM	Descendent Activity-Time
ACTRES	Descendent Activity-Resources
Li & Willis [18]	Backward Scheduling
Vaca [22]	GA
RKV-MM [54]	GA - Random Key Variant for Multi-Mode

Table 3: Results for single-mode instances.

Project network [18]	**Prob No.**	**Optimal duration**	**TR**	**EST**	**EFT**	**LST**	**LFT**	**Total Float**	**ACTIM**	**ACTRES**	**Li & Willis [18]**	**Vaca [22]**	**RKV-MM [54]**	**RKV-MM-AS**
3	17	23	24	24	25	25	24	24	24	23	24	23	23	**23**
5	18	17	21	24	19	23	21	26	23	18	22	17	17	**17**
6	19	39	47	44	45	39	40	39	39	43	39	39	39	**39**
7	20	13	15	15	16	14	13	14	14	16	13	13	13	**13**
8	21	88	112	112	108	92	100	100	92	92	92	88	88	**88**
9	22	45	50	46	50	46	46	53	46	46	46	45	45	**45**
18	23	35	41	40	47	36	39	47	36	39	38	35	35	**35**
19	24	35	42	39	36	39	36	36	39	35	36	35	35	**35**
20	25	64	70	70	74	73	68	74	73	73	67	64	64	**64**

In what follows, it is compared the RKV-AS-MM with the approaches proposed by Vaca [22] and Mendes [54] for the problem instances (multi-mode). To illustrate its effectiveness, we consider a total of 26 instances (with a number of activities between 7 and 80) proposed by Vaca [22] which are adapted to the field of construction. The problem instances require between two and ten renewable resource types. All resources are given as renewable resources and the availabilities associated with all resources are assumed to be constant over time. Instance details are described by Vaca [22].

Table 4, Column 5, summarizes the obtained results by RKV-AS-MM. The results are the best when compared with the values presented by Vaca [22] and Mendes [54]. The maximal computational time dispended is 60 seconds for each instance.

Table 5 shows the *number of instances solved* (NIS) and the *average relative deviation* (ARD) with respect to the *best known solution* (BKS).

Table 4: Experimental results for multi-mode instances.

Probl. No.	**Duration without limited resources**	**BKS**	**Best duration for multi-mode [22]**	**RKV-MM [54]**	**RKV-AS-MM**
1	71	96	107	107	96
2	74	77	77	77	78
3	238	306	309	309	306
4	23	28	28	28	28
5	125	132	132	132	132
6	46	75	75	75	75
7	26	42	42	42	42
8	75	126	141	141	126
9	37	66	66	66	66
10	85	103	103	103	103
11	93	140	140	140	140
12	9	11	11	11	11
13	8	9	9	9	9
14	48	88	88	88	88
15	18	19	19	19	19
16	11	12	12	12	12
17	16	22	23	23	22
18	16	17	17	17	17
19	36	37	37	37	37
20	8	13	14	13	13
21	80	80	80	80	80
22	28	38	38	38	38
23	33	34	40	34	34
24	33	35	37	35	35
25	31	56	61	58	56
26	60	84	86	85	84

For comparison between the methods, we have used one measure, namely the *average relative deviation* (ARD):

$$\boldsymbol{RE} = \sum_{i=1}^{NIS} \frac{C_{\max_i} - \boldsymbol{BKS}_i}{\boldsymbol{BKS}_i} \tag{4}$$

$$\boldsymbol{ARD} = \frac{\boldsymbol{RE}}{\boldsymbol{NIS}} \tag{5}$$

Table 5: Average relative deviation.

Algorithm	**NIS**	**ARD**
Oscar Lopez Vaca [22]	26	0.027
J. Magalhaes-Mendes [54]	26	0.013
This paper	26	0.000

Overall, we solved 26 instances with RKV-AS-MM and obtained an ARD of 0.000%. The RKV-AS-MM obtained the best-known solution for 25 instances, i.e. in 96% of problem instances. RKV-AS-MM presented an improvement with respect to almost all others algorithms.

VIII. CONCLUSIONS

The resolution of the MRCPSP by a two-level genetic algorithm is proposed in this paper. The chromosome representation of the problem is based on random keys. Reproduction, crossover and mutation are applied to successive chromosome populations to create new chromosome populations. These operators are simplicity themselves, involving random number generation, chromosome copying and partial chromosome exchanging.

The schedules are constructed using priorities defined by the genetic algorithm with a constructive heuristic. The constructive heuristic for constructing feasible schedules is extended by the flexible use of different planning directions including the backward and forward improvement (FBI) planning. For some instance, a combination of a heuristic constructive and the genetic algorithm may yield a good result, but in another instance the FBI can improve the initial schedule.

The approach was tested on a set of 26 standard instances with multi-modes in the field of construction, taken from the literature and compared with the others approach. The algorithm produced good results when compared with other approaches therefore validating the effectiveness of the proposed algorithm.

Further research can be extended to problems where the resource availability varies within time and in contexts where several projects must be concurrently executed.

REFERENCES

[1] D.E. Goldberg, "Genetic Algorithms in Search Optimization & Machine Learning", Addison-Wesley, 1989.

[2] J.H. Holland, "Adaptation in Natural and Artificial Systems", Ann Arbor: The University of Michigan Press, 1975.

[3] J. Blazewicz, J.K. Lenstra and A. H. G. Rinnooy Kan, "Scheduling subject to resource constraints: Classification and complexity", *Discrete Applied Mathematics*, 5, 11–24, 1983.

[4] J.J.M. Mendes and J.F. Gonçalves, "A Memetic Algorithm-Based Heuristics for the Resource Constrained Project Scheduling Problem", in Proceedings of *II International Conference on Computational Methods for Coupled Problems in Science and Engineering*, Ibiza, Spain, 644-648, 2007.

[5] J. Magalhães-Mendes, "Project scheduling under multiple resources constraints using a genetic algorithm", *WSEAS Transactions on Business and Economics*, 11, 5, 487-496, 2008.

[6] C.H. Yeh and H. Pan, "System Development for Fuzzy Project Scheduling", *WSEAS Transactions on Business and Economics*, World Scientific and Engineering Academy and Society, USA, Vol. 1(4), pp. 311-317, 2005.

[7] J.J.M. Mendes, J.F. Gonçalves and M.G.C. Resende, "A random key based genetic algorithm for the resource constrained project scheduling problem", *Computers & Operations Research,* 36, 92-109, 2009.

[8] R. Kolisch and S. Hartmann, "Experimental investigation of heuristics for resource-constrained project scheduling: an update", *European Journal of Operational Research*, 174 (1), 23-37, 2006.

[9] J.E. Day and M.P. Hottenstein, "Review of Sequencing Research", *Naval Research Logistics Quarterly*, 17, 11-40, 1970.

[10] W.E. Davis, "Project Scheduling Under Resource Constraints - Historical Review and Categorization of Procedures", *AIIE Transactions*, 5, 4, 147-163, 1973.

[11] R. Slowinski, "Multiobjective Network Scheduling With Efficient Use of Renewable and Nonrenewable Resources", *European Journal of Operational Research*, 7, 265-273, 1981.

[12] F.B. Talbot, "Resource-Constrained Project Scheduling With Time-Resource Tradeoffs: the Nonpreemptive Case", *Management Science*, 28, 10, 1197-1210, 1982.

[13] J.H. Patterson, F.B. Talbot, R. Slowinski and J. Weglarz, "Computational Experience With A Backtracking Algorithm for Solving a General Class of Precedence and Resource-Constrained Scheduling Problems", *European Journal of Operational Research*, 49, 68-79, 1990.

[14] F. F. Boctor, "Some Efficient Multi-Heuristic Procedures for Resource-Constrained Project Scheduling", *European Journal of Operational Research*, 49, 3-13, 1990.

[15] R.A. Jones, "Resource Scheduling: A Monte Carlo Approach to CPM", *Proceedings of CIB W-65*, 422-427, 1992.

[16] H.R. Storer, S.W. Wu, and R. Vaccari, "New Search Spaces For Sequencing Problems With Application to Job Shop Scheduling", *Management Science*, 38, 10, 1495-1509, 1992.

[17] B.L. Maccarthy and J. Liu, #Addressing the Gap In Scheduling Research: A Review of Optimization And Heuristic Methods In Production Scheduling", *International Journal of Production Research*, 31, 1, 59-79, 1993.

[18] K.Y. Li and R.J. Wills, "An Iterative Scheduling Technique for Resource-Constrained Project Scheduling", *European Journal of Operational Research*, 56, 370-379, 1992.

[19] A. Drexel and J. Gruenewald, "Nonpreemptive Multi-Mode Resource-Constrained Project Scheduling", *IIE Transaction*, 25, 5, 74-81, 1993.

[20] O. Moselhi and P. Lorterapong, "Near Optimal Solution for Resource-Constrained Scheduling Problems", *Construction Management and Economics*, 11, 293-303, 1993.

[21] F. F. Boctor, "Heuristics for scheduling projects with resource restrictions and several resource-duration modes", *International Journal Production Research*, 31, 11, 2547-2558, 1993.

[22] O. C. L. Vaca, "Um Algoritmo Evolutivo para a Programação de Projectos Multi-modos com Nivelamento de Recursos Limitados", Ph.D. Thesis, Universidade Federal de Santa Catarina, Florianópolis, Brasil, 1995. (*In portuguese*)

[23] F. F. Boctor, "A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes", *European Journal of Operational Research*, 90, 349-361, 1996.

[24] M. Mori and C. Tseng, "A genetic algorithm for multi-mode resource constrained project scheduling problem", *European Journal of Operational Research*, 100, 134-141, 1997.

[25] R. Kolisch and A. Drexl, "Local search for nonpreemptive multi-mode resource constrained project scheduling", *IIE Transactions*, 29, 987-99, 1997.

[26] B. De Reyck and W. Herroelen, "The multi-mode resource-constrained Project scheduling problem with generalized precedence relations", *European Journal of Operational Research*, 119, 538-556, 1999.

[27] R. Heilmann, "Resource-constrained project scheduling: a heuristic for the multi-mode case", *OR Spektrum*, 23, 335-357, 2001.

[28] S. Hartmann, “Project Scheduling with Multiple Modes: A Genetic Algorithm”, *Annals of Operations Research*, 102, 111-135, 2001.

[29] N. Damak, B. Jarboui, P. Siarry and T. Loukil, “Differential evolution for solving multi-mode resource-constrained project scheduling problems”, *Computers & Operations Research*, 36, 2653-2659, 2009.

[30] A. Lova, P. Tormos, M. Cervantes and F. Barber, “An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes”, *International Journal of Production Economics*, 117, 302-316, 2009.

[31] A.B. Badiru, “Toward The Standardization of Performance Measures For Project Scheduling Heuristics”, *IEEE Transaction On Engineering Management*, 35, 2, 80-89, 1988.

[32] J. Magalhães-Mendes, “Project Scheduling” in Progress in Management Engineering, Editors: Lucas P. Gragg and Jan M. Cassell, Nova Publishers, New York, USA, pp. 117-134, 2009.

[33] J. Magalhães-Mendes, "An Evolutionary Algorithm for the Resource Constrained Project Scheduling Problem", in *Proceedings of the Sixth International Conference on Engineering Computational Technology*, M. Papadrakakis and B.H.V. Topping, (Editors), Civil-Comp Press, Stirlingshire, United Kingdom, paper 67, 2008.

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

[35] LL. Lorenzoni, H. Ahonen and AG. Alvarenga, “A multi-mode resource-constrained scheduling problem in the context of port operations”, Computers and Industrial Engineering, 50: 55-56, 2006.

[36] E. Demeulemeester, B. De Reyck, and W. Herroelen, “The discrete time/resource trade-off problem in project networks: a branch-and-bound approach”, *IIE Transactions*, 32, 1059-1069, 2000.

[37] R. Kolisch and S. Hartmann, “Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis”, J. Weglarz (editor), Kluwer, Amsterdam, the Netherlands, 147–178, 1999.

[38] M. Speranza and C.Vercellis, “Hierarchical models for multi-project planning and scheduling”, *European Journal of Operational Research* 64, 312-325, 1993.

[39] S. Hartmann and A Sprecher, “A note on hierarchical models for multi-project planning and scheduling”, *European Journal of Operational Research*, 94, 377-383, 1996.

[40] A. Sprecher, S. Hartmann and A. Drexl, “An exact algorithm for the project scheduling with multiple modes”, *OR Spektrum* 19, 195-203, 1997.

[41] S. Hartmann and A. Drexl, “Project scheduling with multiple modes: a comparison of exact algorithms”, *Networks 32*, 283-297, 1998.

[42] A. Sprecher and A. Drexl, “Solving multi-mode resource-constrained project scheduling problems by a simple, general and powerful sequencing algorithm”, *European Journal of Operational Research* 107, 431-450, 1998.

[43] E. Demeulemeester, B. De Reyck and W. Herroelen, “The discrete time/resource trade-off problem in project networks: a branch-and-bound approach”, *IIE Transactions* 32, 1059-1069, 2000.

[44] L. Ozdamar and G. Ulusoy, “A local constraint based analysis approach to project scheduling under general resource constraints”, *European Journal of Operational Research*, 79, 287-298, 1994.

[45] G. Knotts, M. Dror and B. Hartman, “Agent-based project scheduling”, *IIE Transactions* 32 (5), 387-401, 2000.

[46] L. Ozdamar, “A genetic algorithm approach to a general category project scheduling problem”, *IEEE Transactions on Systems, Man, and Cybernetics*, Part C: Applications and Reviews, 29, 44-59, 1999.

[47] J. Alcaraz, C. Maroto and R. Ruiz, “Solving the multi-mode resource constrained project scheduling problem with genetic algorithms”, *Journal of the Operational Research Society*, 54, 614-626, 2003.

[48] R. Slowinski, B. Soniewicki and J. Weglarz, “DSS for multiobjective project scheduling”, *European Journal of Operational Research*, 79, 220-229, 1994.

[49] J. Jozefowska, M. Mika, R. Rozycki, G. Waligora and J.Weglarz, “Simulated annealing for multi-mode resource-constrained project scheduling”, *Annals of Operations Research*, 102, 137-155, 2001.

[50] K. Bouleimen and H. Lecocq, “A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version”, *European Journal of Operational Research* 149, 268-281, 2003.

[51] H. Zhang, C. Tam and H. Li, “Multi-mode project scheduling based on particle swarm optimization”, Computer-Aided Civil and Infrastructure Engineering 21, 93-103, 2006.

[52] F. Nafici, “Design, implementation and application of a network software for scheduling multiple resource-constrained projects”, Unpublished Ph.D. Thesis, University of Leeds, 1976.

[53] A. Lova, P. Tormos and F. Barber, Multi-Mode Resource Constrained Project Scheduling: Scheduling Schemes, Priority Rules and Mode Selection Rules, *Revista Iberoamericana de Intelegencia Artificial*, no. 30, pp. 69-86, 2006.

[54] J. Magalhães-Mendes, "Project Scheduling with Multiple Modes: A Genetic Algorithm based Approach", in Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, B.H.V. Topping, Y. Tsompanakis, (Editors), Civil-Comp Press, Stirlingshire, United Kingdom, paper 12, 2009. doi:10.4203/ccp.92.12.

[55] J.J.M. Mendes, “Sistema de Apoio à Decisão para Planeamento de Sistemas de Produção do Tipo Projecto”, Ph.D. Thesis, Departamento de Engenharia Mecânica e Gestão Industrial, Faculdade de Engenharia da Universidade do Porto, Portugal, 2003. (In portuguese).

[56] D.E. Goldberg, “Genetic Algorithms in Search Optimization & Machine Learning”, Addison-Wesley, 1989.

[57] C. Patrick, “Construction Project Planning and Scheduling”, Pearson Prentice Hall, Ohio, 2004.

[58] H. Pan and C.H. Yeh, “GA for Fuzzy Multi-Mode Resource-Constrained Project Scheduling”, *WSEAS Transactions on Systems*, Issue 4, Volume 2, pp.893-990, 2003.

[59] M. Seda, “Solving Resource-Constrained Project Scheduling Problem as a Sequence of Multi-Knapsack Problems”, *WSEAS Transactions on Information Science & Applications*, Issue 10, Vol. 3, pp.1785-1791, 2006.

[60] C. Darwin, “The Origin of Species by Means of Natural Selection”, London, John Murray, Albemarle Street, 1859.

J. Magalhães-Mendes was born in Mancelos (Amarante, Portugal) on January 17, 1963. He received the PhD in Mechanical Engineering and Industrial Management at the University of Oporto, in 2004 and the licentiate degree in Applied Mathematics from the same university. He has also a licentiate degree in Civil Engineering from the Polytechnic of Oporto and a M.S. degree in Systems and Automation by University of Coimbra.

He has been Associate 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, WSEAS Journals and several international conferences. His research interest includes construction management, project management, genetic algorithms, and operational research and supply chain management.

The main papers:

1. J.J. Magalhães Mendes, J.F. Gonçalves and M.G.C. Resende, "A random key based genetic algorithm for the resource constrained project scheduling problem", Computers & Operations Research, Vol. 36, pp. 92-109, 2009.
2. J.F. Gonçalves, J.J. Magalhães Mendes and M.G.C. Resende. “A genetic algorithm for the Resource Constrained Multi-Project Scheduling Problem”. European Journal of Operational Research, Vol. 189, pp. 1171-1190, 2008.
3. J. F. Gonçalves, J.J. Magalhães Mendes and M.G.C. Resende. “A hybrid genetic algorithm for the job shop scheduling problem”. European Journal of Operational Research, Vol. 167, pp. 77-95, 2005.

Dr. Magalhães-Mendes is a member of Portuguese Engineering Association (OE) and Portuguese Association of Operational Research (APDIO).