GRASP APPROACH TO RCPSP WITH MIN-MAX ROBUSTNESS OBJECTIVE

Hayet Mogaadi and Besma Fayech Chaar

National Engineering School of Tunis, El Manar University, Tunisia
mogaadi_h@yahoo.fr
besma.fayechchaar@insat.rnu.tn

ABSTRACT

This paper deals with the Resource-Constrained Project scheduling Problem (RCPSP) under activity duration uncertainty. Based on scenarios, the object is to minimize the worst-case performance among a set of initial scenarios which is referred to as the min-max robustness objective. Due to the complexity of the tackled problem, we propose the application of the GRASP method which is qualified as a simple and effective multi-start metaheuristic. The proposed approach incorporates an adaptive greedy function based on priority rules to construct new solutions, and a local search with a forward-backward heuristic in the improvement phase. Two different benchmark data sets are investigated, the Patterson set and the PSPLIB J30 set. Comparative results show that the proposed enhanced GRASP outperforms the basic procedure in robustness optimization.

KEYWORDS

RCPSP, uncertainty, Robustness, scenario, GRASP, intensification

1. INTRODUCTION

The Resource-Constrained Project Scheduling Problem (RCPSP) is a well-known project scheduling problem that consists in scheduling a set of activities over resources with limited capacities subject to precedence and resource constraints while optimizing several objectives. The most common objective is to minimize the project duration (so called makespan and denoted by Cmax). As a generalization of a majority of the classical scheduling problems, the RCPSP was classified as NP-hard [1]. This problem was widely studied in the literature; heuristics and metaheuristics were successfully applied in this context such as Genetic Algorithms, Sampling methods, based local search methods, Simulated Annealing, etc. Efficient surveys are given in the following references [2, 3].

Nevertheless, the project scheduling process is really subject to unexpected events that may be related to activities or resources leading, in the most of cases, to schedule disruptions. In the last few decades, the researchers ‘efforts were focused in managing uncertainty in project scheduling to avoid the schedule disruption and the performance degradation when perturbations occur.

David C. Wyld et al. (Eds) : CCSEA, CLOUD, DKMP, SEA, SIPRO - 2016
pp. 137–146, 2016. © CS & IT-CSCP 2016 DOI : 10.5121/csit.2016.60212

One of the basic approaches to deal with uncertainty [4] is the robust approach having the object to find a schedule that remains with a highest quality across a set of scenarios. A scenario represents a problem realization which is founded by matching fixed values to uncertain problem parameters. Inspired from the decision analysis, the scenario-based approach is a simple and effective way to model uncertainty. With the absolute robustness objective, referred to as the min-max objective, the aim is to minimize the maximum performance degradation among all scenarios. However, the regret robustness objective is to minimize the maximum deviation of solutions from optimality across all scenarios.

In the literature [5], Kouvelis and Yu have investigated the cited robustness objectives for different combinatorial optimization problems. Although robust scheduling problems are more blinded to reality, solution methods for robust RCPSP are not exhaustive. In [6], Al Fawzen and Haouari have proposed a bi-objective model for RCPSP with the minimization of the makespan and the maximization of the robustness. The problem was solved by a tabu search heuristic. Chtourou and al. [7] have studied various robustness measures based on priority rules when activity durations vary. The work of Artigues and Leus [8] deals with RCPSP under activity uncertainty. Based on PLNE, the authors proposed a scenario-based bi-level problem formulation that minimizes the absolute and relative regret robustness. In this model, a solution depends on priority rule, also called scheduling policy. The authors have applied, in first, exact method which has taken excessive computational time considering medium sized instances. So they were directed towards heuristic procedures. In addition, the Genetic algorithm was simply adapted to robust optimization problems, such as the one machine problem [9] and the robust RCPSP [10].

Heuristics and metaheuristics are also approved as efficient methods for stochastic RCPSP (SRCSP) where the uncertainty is modeled by probabilistic distributions, and the robustness is evaluated in terms of expected makespan. We cite the work of [11] in which metaheuristics were well investigated to SRCPSP. Recently, the work of [12] gives promising results for RCPSP under uncertainty.

In this context, we are encouraged to use the GRASP to the scenario-based robust RCPSP. Our tackled optimization problem aims to maximize the absolute robustness objective. We propose a GRASP algorithm enhanced with a forward-backward heuristic.

The next section focuses on the problem definition in deterministic and non deterministic version. Section 3 describes the main phases of the GRASP method. In section 4, we explicit the application of the latter method to the robust RCPSP. Computational results are given in section 5. Section 6 concludes the paper.

2. ROBUST PROJECT SCHEDUING PROBLEM

2.1. Deterministic RCPSP

A deterministic version of RCPSP consists in performing a set A of n activities on a set K of m resources. Every activity i has a fixed processing time denoted by p_i and requires r_{ik} units of resource type k which is characterized by a limited capacity R_k that must not be exceeded during the execution, and activities must not be interrupted. Two additive dummy activities 0 and n + 1 are used that represent to start and the end of the project, respectively. Dummy activities have null time duration and null resource requirement. The objective of the standard RCPSP is to construct a precedence and resources feasible schedule with the minimum makespan.

Precedence constraints perform that the start time of an activity i is permitted only when all its previous activities are finished. The Resource constraints satisfy that the use of every resource type, at every instant, does not exceed its capacity.

A schedule S referred to the baseline schedule which is given by the list of activity finish times (start times); let F_i (>=0) denotes the finish time of an activity i, then $S=(F_0, F_1, \ldots, F_n, F_{n+1})$ and the total project duration corresponds to the end project finish time F_{n+1}.

Therefore, the conceptual formulation of the RCPSP is given by the following formula:

$$\min F_{n+1} \text{ sc.} \tag{1}$$

$$F_h \le F_j - p_j; j = 1\Lambda\ n+1; h \in P_j \tag{2}$$

$$\sum_{i \in A(t)} ri_k \le R_k; k \in K; t \ge 0 \tag{3}$$

with $A(t)$ denotes the set of activities which are executing at time t, and P_j denotes the set of predecessors of the activity i.

An instance of the RCPSP can be represented by a graph G = (V, E) where the set of nodes V is defined by project activities and E contains arcs according to the precedence relations.

2.2. Min-Max robust RCPSP

The considered variability, for RCPSP under uncertainty, relies on activity durations. We use a scenarios-based approach to model the problem variability. Hence, we construct a set of scenarios, denotes by $\sum$, for optimization, let σ_i be a single scenario that corresponds to a problem realization. Each scenario is found by altering the initial activities durations with respect to a maximal activity delay.

A feasible solution x for the robust scheduling problem is represented by an activity list; let $f(x, \sigma_i)$ denotes the *makespan* of the generated schedule according to x on scenario σ_i. This value defines the local performance of the solution x according to σ_i. However, the global optimization process has to find the robust schedule with the global performance across the optimization set. Usually, the global performance is measured in terms of mean value, maximum deviation, etc.

The object of the present work is to optimize the min-max robustness objective of RCPSP which consists in minimizing the maximum makespan value over all scenarios. The optimization objective is given by the following formula.

$$\min \max_{\sigma_i \in \Sigma} (f(x, \sigma_i)) \tag{4}$$

Resource and precedence constraints for the robust RCPSP are the same in the deterministic case (Equations (2) and (3)).

2.3. Complexity

The robust scenario-based robust RCPSP with the min-max robustness objective is an NP-hard problem as it can be reduced to the standard NP-hard deterministic version for a number of scenarios equals to one [14].

3. GRASP METAHEURISTIC

The GRASP (Greedy Randomized Adaptive Search Procedures) is a multi-start metaheuristic which was developed for combinatorial optimization [15, 16]. It consists of an iterative process, in each of one, two phases are performed: a construction phase and a local search phase. The first one permits the construction of a feasible solution iteratively, one element at once iteration. However, the second phase performs the improvement of the recently constructed solution by a simple local search heuristic. The best across all generated solutions is then retained.

In the construction phase, a Candidate List (CL) is generated that contains the set of the candidate elements (edges) to be selected and added to the current partial solution. The CL is ordered with respect to a greedy function that measures the benefit of selecting each element. Moreover, the effective selection of one edge is done from an additive list: the Restricted Candidate List (RCL) that regroups the best elements from the CL with highest greedy values.

The GRASP procedure combines crucial characteristics of search methods. In the one hand, it is adaptive because the greedy function values are updated continuously depending on the current partial solution and the considered schedule construction strategy. In the other hand, it is a randomized-based method such that a selection of one element in the RC L is done randomly.

4. APPLICATION TO THE ROBUST RCPSP

We propose the application of the GRASP approach to the RCPSP with the optimization of the absolute robustness so called the min-max robustness objective.

The main steps of the proposed approach are depicted in the following figure. As a multi-start heuristic, the algorithm starts with generating gradually a new solution. This step integrates an intensification strategy. Current solution is improved, in the second step, by a Forward-Backward Improvement heuristic (FBI) and a Local Search heuristic (LS).

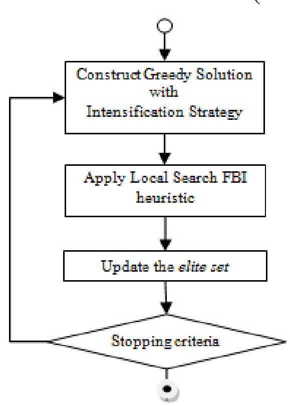

Figure 1. General steps of the enhanced GRASP approach

Throughout this iterative process an elite set (ES) is generated containing the best encountered solutions. The size of the ES is defined by a fixed parameter "*nbElite*". Elite solutions are updated iteratively.

4.1. Solution representation and *robust fitness*

A solution is represented by an activity list that satisfies precedence constraints. To evaluate the global solution performance, a *robust evaluation* is made. We apply decoding procedure to generate the schedule according to the activity list *x* and the scenario σ_i. Then, the *robust fitness* which measures the global solution performance is determined by the maximum makespan over all obtained values for $\sum$. The Serial Schedule Generation Scheme (Serial SGS) is used as a decoding procedure [2] to construct the schedule.

4.2. Construction phase with intensification strategy

At one iteration of the construction phase one activity is selected from an eligible set an added to the current partial solution. We generate the list CL of candidate activities having all their predecessors scheduled. For each activity in the CL, the corresponding greedy function value is equals to the priority rule value. We propose the application of different priority rules: the minimum Latest Finish Time (MLFT), the minimum of the activity free slack (MFLK), the inverse free slack priority rule, and the critical activity based selection. The object is to study the effect of the priority rule on robustness objective.

First activities of the CL are then copied in the *RCL*. The size of the latter list is denoted by *TRCL*. From the constructed RCL, an activity is then chosen randomly and added at the latest position in the partial activity list. A pool of elite solutions *ES* is constructed.

To ensure solutions with a high quality, we incorporate in the construction phase an intensification strategy based on the elite set.

In fact, when the *ES* attempts the fixed parameterized size, then, with probability *pES*, we select randomly one element to be considered at the current iteration of the construction phase. Then, the first activity, in the elite solution, that does not appear in the partial current solution is selected and inserted in.

The above described process is repeated until the construction of the totality of the solution is reached.

4.3. GRASP Improvement Phase

The proposed GRASP improvement phase combines a Local Search procedure (LS) with a FBI heuristic. The proposed Local Search starts from the recently constructed and improved solution *x*. Iteratively, a local move is applied to *x* to generate a neighbourhood set: N(*x*). The proposed move consists of the permutation of one activity of *x* with others nodes. The activity to be permuted is chosen at random. Obtained feasible solutions are saved to be compared with *x*. The best element over all neighbours and the current solution is retained. After a maximum number of iterations, the search method would stop with the best solution over all neighbourhoods.

The Forward-Backward Improvement (FBI) method is one of the basic heuristic for project scheduling. It was successfully hybridized with others methods ensuring efficiency and an acceptable computational time increment [17]. The Forward recursion is given by the serial SGS. However, the backward recursion is the SGS algorithm applied to the precedence-reverse graph starting from the end project activity where priority values are determined according to the lastly generated schedule.

5. EXPERIMENTS

5.1. Data Sets and Scenario Generation

The proposed approach was implemented in Java and ran on a portable personnel computer equipped with an Intel® Core™ i5-2450M CPU@ 2.50 GHz 773MHz, 2.70Go of RAM. Experiments were performed on two benchmark project instances: the Patterson data set [18], and the PSPLIB J30 data set [19]. The first data set contains 110 instances of various projects with 3 resource types and a number of activities that vary between 6 and 51. However, the second data set contains 480 project instances which are generated by the ProGen generator. These instances represent different projects with only 30 activities and 4 resource types.

We generated scenarios with limited size for both the optimization and the evaluation (simulation) set. The optimization set contains *nbScen* scenarios, equals to 10, used to compute the robustness objective. The evaluation set is used for simulation to estimate the expected makespan. The size of the evaluation set is denoted by *l*. A scenario is an initial problem realization where a set of activities are modified by altering their initial durations. In fact, for 10 percent of the total project activities, we add a time increment δ which is taken from a uniform distribution U(1, *maxDelay*). The latter parameter indicates the maximum activity delay which is fixed to 10.

In order to evaluate the performance of the proposed approach, we were interested by the following performance measures:

- The estimate *Expected makespan* which is calculated over the evaluation set $\left(\mathrm{E}(\mathrm{C}_{\max}) = \frac{1}{l}\sum_{i=1}^{l}(f(x,\sigma_{\mathrm{i}}))\right)$;

- The *Standard deviation* of the makespan over the evaluation set;

- The *Relative Optimality gap* that measures the deviation between the estimate expected makespan and the lower bound *LB*, or the optimal makespan if exists, for the corresponding deterministic project $\left(\frac{E(\mathrm{C}_{\max})-LB}{LB}\right)$.

All results are averaged by the number of tested project instances.

5.2. Performance evaluation

5.2.1. Deterministic case

It is inevitable to study the algorithm behaviour on the deterministic case. Hence, the table 1 shows results of the GRASP implementation on the J30 data set. We performed the basic GRASP approach which based on a Local Search (LS) in the improvement phase (column 2). Then, the basic algorithm is improved with the FBI and tested for the same instances set. We vary the number of maximum iteration for both GRASP process (Line 2) and the local search (Line 3). Line 4 reported the average deviation from the well-known optimal solutions in percent, for both two GRASP implementations. The number of the obtained optimums is given in Line 5.

Table 1. Average deviations from optimal solutions J30 data set instances (the deterministic case).

	GRASP-LS	**GRASP-LS+BFI**			
Iterations for GRASP	100	100	300	3000	1000
Iterations for LS	100/2	10	10	10	1000/4
Optimality deviation	0.51	0.57	0.34	0.24	0.20
Optimums	396	390	419	428	434

Results for static RCPSP show the performance of the applied GRASP procedure compared with other methods in the literature [3], especially when combined with the forward-backward heuristic.

5.2.2. Results for Robust case

Under uncertainty, we have performed different runs of the proposed GRASP on Patterson data set. The basic algorithm denoted as (GRASP-LS(10)) is considered as the implementation of the GRASP approach with the LFT priority rule in the construction phase and 10 iterations of local search procedure.

We firstly, vary the maximum number of iterations with the local search incorporated in GRASP. Results are reported in table 2 with 1000 simulations as the size of the evaluation set. The evaluation procedure was ran for each project instance.

Table 2. Robustness evaluation on Patterson data set (1000 simulations).

	GRASP-LS(10)	**GRASP-LS(20)**	**GRASP-LS(100)**
Avg. Optimality gap	0.2366	0.2384	0.2249
Avg. Standard deviation	3.4423	3.4026	3.4121

Referring to the table 2, the local search procedure has an impact on the global performance. In fact, an increment of the number of iterations yields to better results for robustness. However, this parameter must be controlled to ensure the non degradation of the later objective, which is the case with 100 iterations in the local search procedure. Table 3 contains numerical results of the proposed GRASP approach on J30 data set under uncertainty, simulated over 100 replications.

Table 3. Robustness evaluation on J30 data set (100 simulations).

		GRASP-LS(10)	**GRASP-LS(20) + BFI**
Nunber of iterations		500	250
Optimality gap	**avg.**	0.1349	0.1243
	max.	0.2755	0.2603
Standard deviation	**avg.**	4.2515	4.1428
	max.	6.3863	5.8086

In order to study the efficiency of the enhanced GRASP approach for the robust RCPSP, we evaluate the computational time on Patterson instances set for 1000 generated schedules, reported in Table 4. As described in [17], the FBI heuristics needs two passes of the SGS procedure to doubly justified the initial schedule. Thus, the number of generated schedules with the basic GRASP algorithm and the enhanced version with a FBI heuristic is equals to$(\mathrm{nbIterMax} \times 10)$ and$(\mathrm{nbIterMax} \times (10 + 3))$, respectively.

Table 4. Comparison between GRASP and GRASP-FBI on Patterson data set (1000 simulations).

		GRASP-LS(10)	**GRASP-LS(10) + BFI**	**GRASP-LS(10) + BFI**
Nunber of iteration		100	75	333
Optimality gap	**avg.**	0.2362	0.2304	0.2259
	max.	1.3352	1.31197	1.3398
Standard deviation	**avg.**	3.4106	3.4008	3.4089
	max.	5.5832	5.6207	5.5526
Time(s)		2.04	1.775	7.926

In table 4, column 2 and 3 show that for maximum 1000 generated schedules, the enhanced GRASP outperforms the basic GRASP in terms of robustness and computational time.

5.2.3. Priority rule

The idea of the present experiment is to study the effect of priority rules on robustness solution quality. As described in section 4, the construction phase implements a priority rule to order the Candidate List content, from which we select the *TRCL* best elements to the *RCL.*

We investigate in table 5 different priority rules based on critical path: Minimum Latest Finish Time (MLFT), the minimum Slack (MSLK), the inverse MSLK, the critical Activity based rule. The total activity slack is obtained by the difference between its latest and earliest start time. We also propose to study a priority rule which based on graph structure which is the GPRW (Greatest Rank Positional Weigth). We ran the GRASP algorithm with two different RCL size values (*TRCL*).

With limited size of the restricted list, the standard deviation of the estimated makespan is decreased as we reinforce best elements in the *RCL*. The inverse MSLK gives better results than the MSLK; this result can be interpreted as the inverse SLK favour activities having greatest slack values, consequently generated schedule will be more flexible to absorb activity delays.

Table 5. The Standard deviation variation on Patterson data set (1000 simulations).

	GRASP-LS(10)	
Priority rule	**TRCL=5**	**TRCL=3**
MLFT	3.4721	3.4456
MSLK	3.4702	3.4871
inverse MSLK	3.4640	3.4454
Critical Activity	3.4533	3.4741
GPRW	3.4606	3.4499

6. CONCLUSIONS

This paper has presented the application of the GRASP approach to the robust RCPSP. Based on scenarios, the object of the tackled optimization problem is to maximize the min-max robustnesss. The proposed GRASP approach incorporates priority rules in the greedy construction phase and two procedures in the improvement phase such as a local search and the forward-backward heuristic. Experiments have shown the simplicity of the GRASP implementation as a multi-start heuristic compared with other complex metaheuristic as evolutionary-based approach. In addition, the presented meta-heuristic was efficient to deal with uncertainty in acceptable computational time. Further works must be concentrated on the study of the GRASP construction phase to explore diverse solution on the search space, and the application of the algorithm to more large-sized project instances in robust scheduling.

REFERENCES

[1] Blazewicz, J., Lenstra, J., Rinnooy Kan, A., (1983), “Scheduling subject to resource-constraints: Classification and complexity”, Discrete Applied Mathematics, Vol. 5, pp. 11–24.

[2] Kolisch, R., Hartmann, S. ,(1999), “Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis”, International Series in Operations Research and Management Science, Weglarz,J. (ed.), Vol. 14, pp. 147-178.

[3] Kolisch, R., Hartmann, S., (2006) ,“Experimental investigation of heuristics for resource-constrained project scheduling: An update”, European Journal of Operational Research, Vol. 174, pp. 23-37

[4] Davenport, A.J., Beck, J.C., (2000), “A survey of techniques for scheduling with uncertainty”, Available from http://tidel. mie. utoronto.ca/publications.php.

[5] Kouvelis, P. , Yu., G., (1997), “Robust Discrete Optimization and Its Applications”. Klower Academic Publisher.

[6] Al-Fawzan, M., Haouari, (2004), “M.: A bi-objective model for robust resource-constrained project scheduling”, International Journal of production economics, Vol. 18, pp. 1-13.

[7] Chtourou, H., Haouari, M., (2008), “A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling”, Comput. Ind. Eng., Vol. 55, pp. 183-194.

[8] Artigues, C., Leus, R., Talla Nobibon, F., (2013), “Robust optimization for resource-constrained project scheduling with uncertain activity durations”, Flexible Systems and Management Journal, Vol. 25(1-2), pp. 175-205.

[9] Sevaux, M. , Sorensen, K., (2004), “A genetic algorithm for robust schedules in a just-in-time environment”, 4OR ,Quaterly journal of Operations Research Societies, Vol. 2(2), pp.129-147.

[10] Mogaadi, H., Fayech, B., (2015), “Scenario-Based Evolutionary Approach for Robust RCPSP”, International Afro-European Conference for Industrial Advancement AECIA.

[11] Ballestin, F., R. Leus, (2009), “Resource-constrained project scheduling for timely project completion with stochastic activity durations", Production and Operations Management, Vol. 18, pp. 459-474.

[12] Creemers, S, (2015), “Minimizing the expected makespan of a project with stochastic activity durations under resource constraints”, Journal of Scheduling, Vol. 18 (3), pp. 263–273.

[13] S. Horroelen, Leux, R.., (2005), “Project scheduling under uncertainty: Survey and research potentials”, European Journal Of Operational Research, Vol. 165(2), 289–306.

[14] Aissi, H., Bazgan, C., Vanderpooten, D. (2009), “Min–max and min–max regret versions of combinatorial optimization problems: A survey”, European journal of operational research, Vol. 197(2), pp. 427-438.

[15] Feo, T.A., Resende, M.G.C., Smith, S., (1994), “A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set”, Operations Research, Vol. 42, pp. 860-878.

[16] Festa, P., Resende, M. G. (2009), “Effective application of GRASP”, Wiley encyclopedia of operations research and management science.

[17] Valls, V., Ballestin, F., Quintanilla, S., (2005), “Justification and rcpsp: A technique that pays”, European Journal of Operational Research, Vol. 165, pp. 375-386.

[18] Patterson, J. H.,(1984) ,“A comparison of exact approaches for solving the multiple constrained resource, Project Scheduling Problem”, Management Science, Vol. 30, p854-867.

[19] Kolisch, R., A. Sprecher, (1996), “PSPLIB - A project scheduling problem library”, European Journal of Operational Research, Vol. 96, pp. 205-216.

AUTHORS

Hayet Mogaadi received a diploma of Engineer in Computer Science from the National School of Computer Sciences (Tunisia) in 2003, and a Master degree in Automatic and Signal Processing from the National Engineering School of Tunis (Tunisia) in 2005. She is a Ph.D. student in Electrical engineering at the National Engineering School of Tunis. Her interest’s area is project scheduling.

Besma Fayech Chaar received the diploma of Engineer in Industrial Engineering from the National Engineering School of Tunis (Tunisia) in 1999, the D.E.A degree and the Ph.D degree in Automatics and Industrial Computing from the University of Lille (France), in 2000, 2003, respectively. Currently, she is a teacher assistant at the University of Tunis (Tunisia). Her research interests include artificial intelligence, decision-making systems, scheduling, and transportation systems.