A Genetic Algorithm Application for Multi-objective Multi-project Resource Constrained Project Scheduling Problem

Fikri Küçüksayacıgil, Gündüz Ulusoy

Sabanci University, Faculty of Engineering and Natural Sciences, Industrial Engineering Department, 34956, Istanbul, Turkey (kucuksayacigil@sabanciuniv.edu), (gunduz@sabanciuniv.edu)

Abstract
Resource Constrained Project Scheduling Problem (RCPSP) has been studied extensively by researchers by considering limited renewable and non-renewable resources. Several exact and heuristic methods have been proposed. Some important extensions of RCPSP such as multi-mode RCPSP, multi-objective RCPSP and multi-project RCPSP have also been focused. In this study, we consider multi-project and multi-objective resource constrained project scheduling problem. As a solution method, non-dominated sorting genetic algorithm (NSGA-II) is adopted. By experimenting with different crossover and parent selection mechanisms, a detailed fine-tuning process is conducted, in which response surface optimization method is employed. In order to improve the solution quality, backward-forward pass (BFP) procedure is proposed as both post-processing as well as for new population generation. The performance of the algorithm and CPU times are reported. The results show that backward-forward pass procedure is successful to improve the solution quality.

Keywords: RCPSP, Genetic Algorithms, Multi-objective RCPSP, Multi-project RCPSP, backward-forward scheduling

1 Introduction

RCPSP has been extensively studied in the literature. There are several extensions of this problem such as multi-mode RCPSP, multi-objective RCPSP and multi-project RCPSP. As the number of project-based organizations increases, the importance of multi-project RCPSP increases as a management tool. In the last decades, projects become increasingly more preferred way of doing business. Projects have by their nature a number of stakeholders. For example, there can be a client and a contractor who negotiate the due date and the cost of the project. This decision environment can then be represented as a bi-objective RCPSP with the minimization of the makespan (Cmax) and the maximization of the net present value (NPV) being the objectives. There are further relevant objectives such as, for example, minimization of the maximum outflow, i.e., the maximum cumulative cash outflow throughout the project duration.

In the following, first the solution methodology is explained in some detail and then an extensive computational study is presented. The computational results reveal that BFP procedure proposed here as an improvement routine works well.

2 Solution methodology

In this study, genetic algorithm (GA) approach for the study of multi-objective multi-project RCPSP is studied. NSGA-II is preferred as the solution method, since it represents the current state-of-the-art among GA approaches to this problem [2]. The parameters of NSGA-II, which are population size, generation number, crossover rate and mutation rate are determined with a detailed fine-tuning experiment. In addition to classical GA operators some additional operators are included in NSGA-II, such as non-dominated sorting procedure and crowding distance operator. In this study, NSGA-II is further extended through some improvement procedures and divergence applications are proposed. BFP procedure ([5], [6]) is applied as an improvement routine on the solutions.

2.1 Individual representation

An individual is represented by a double list including precedence feasible activity list and mode assignment list. In precedence activity list, activities are placed into the genes so that the predecessors of an activity appear before that activity. By doing so, precedence relations between the activities are satisfied. In the second list, modes assigned to each activity from their mode sets exist.

2.2 Initial population generation

In this study, two different initial population generation techniques are applied. The first one is called the random initial population generation technique, where the precedence feasible activity list is formed by selecting the next activity from the eligible set randomly. In other words, the activities

Published in the Proceedings of the 15th Workshop of the EURO Working Group, Toklu, Cengiz and Bekdas, Gabriel (eds.), Bilecik Seyh Edebali University, Bilecik, Turkey, March 2014, 49-52, [Updated on October 16, 2015]

existing in the eligible set have equal probability of selection. Another type of random sampling applied here is the regret based-biased random sampling, where the selection probabilities are derived from the latest finishing time of the activities and assigned to corresponding activities. It has to be noted that both of them are implemented, but random sampling is generally preferred because of undesired complexity of regret based-biased random sampling. After completing the precedence feasible activity list, modes are assigned to each activity randomly.

In the second method called the feasible initial population generation technique, the precedence feasible activity list is formed with random sampling. However, mode assignment methods vary because of the observation that random mode assignment sometimes causes infeasible individuals with respect to non-renewable resource capacities. Thus, several mode assignment techniques are proposed in order to use them along with random mode assignment.

2.3 Scheduling the activities

In order to schedule the activities, serial schedule generation scheme is used ([3], [4]). During assigning a starting and finishing time for an activity, precedence relations and renewable resource capacities are taken into account. In addition, it is also considered that an activity cannot start earlier than the activities existing in the previous positions of precedence feasible activity list.

2.4 Chromosome evaluation

In NSGA-II, an individual is assigned a rank value instead of fitness value, which is used in classical GAs so as to evaluate the quality of an individual. Rank value is assigned by using the domination principle. An individual dominates another individual if all objective values are better than those of the second individual and at least one objective value is strictly better than that of the second individual.

For maintaining the diversity of the algorithm, crowding distance operator is employed developed by [2]. Crowding distance of an individual represents how far that individual is from the neighbouring individuals on the same front.

2.5 Forming the next generation

Three different crossover techniques proposed in the literature are used in this study: One-point crossover, two-point crossover and multi-component uniform order-based crossover.

As for mutation operator, it tries to change every activity pair *i* and *i+1*, if the precedence relations and mutation rate are satisfied. In addition, mutation operator tries to change the mode of every activity if the mutation rate is satisfied.

Parent selection is applied with two different methods. In roulette wheel selection, selection probabilities of the individuals are determined by using rank values so that the individuals whose rank values are the best are assigned the highest selection probabilities. In binary tournament selection, the winner is determined by using the rank values and crowding distance values.

Population reduction is succeeded with a simple method that the best individuals with respect to rank value are left in the population and the worst individuals are removed from the population. If a tie occurs among the rank values, then the individuals whose crowding distance values are the highest are selected for the next generation.

3 Fine-tuning of the parameters

The parameters of the algorithm, which are population size, generation number, crossover rate and mutation rate, are determined with response surface optimization. It is a statistical method in order to optimize the multiple output variables in the existence of multiple input variables. In our case, input variables are parameters of the algorithm. As for the output variables, performance measures preferred are hypervolume, maximum spread and the number of non-dominated individuals.

For applying response surface optimization, 10-activity, 20-activity and 30-activity instances are selected. An instance is solved five times using a parameter and operator combination (that is, the combinations of crossover and parent selection mechanisms). In each replication, three performance measures are calculated. Average performance measures for a parameter combination are obtained after taking the average values of the performance measures per five replications. Thus, at the end, each instance has several average performance measures each of which pertains to a parameter combination. Using the performance measures, response surface optimization calculates a desirability value, which represents the quality of the parameter combination. For each instance, the parameter combination is selected whose desirability value is the highest. For selecting a unique parameter combination for each 10-activity, 20-activity and 30-activity instance sets, the parameter combination, whose difference of its parameter values from the parameters of other parameter combinations is the least, is selected.

In order to select the best operator combination, each instance is solved with the determined parameter combination. After evaluating the solution qualities of each operator combination, one-point crossover and roulette wheel selection mechanism are determined to be the best combination.

For larger size of projects, the same fine-tuning process is repeated with some differences. In the current experiment, crossover and parent selection mechanism are not experimented and one-point selection and roulette wheel selection mechanism are used. In addition, crossover rate and mutation rate are not experimented, rather the crossover rate and mutation rate that are determined as the best for 30-activity instances in the previous experiment are used. Moreover, population size and generation number are the functions of the number of the activities existing in the project network. At the end of the current experiment, the best population size and generation number coefficients are determined for each objective combination.

4 Extensions of NSGA-II

In this study, several divergence application and local searches are developed. However, we will report here on the results obtained employing backward-forward pass (BFP) procedure.

BFP procedure applies backward pass and forward pass sequentially. Before applying, the individual is inserted into an empty individual list. Backward pass shifts all the activities to the right by considering the precedence relations and renewable resource capacities after sorting the activities in decreasing order of finishing times. In contrast, forward pass shifts all activities to the left by considering the same conditions after sorting the activities in increasing order of starting times. After each backward pass, if the starting time of the first activity in the precedence feasible activity list is larger than zero, then finishing and starting time of all activities are decreased by that difference. At this phase, the resulting individual is inserted into the individual list. After each forward pass, the resulting individual is inserted into the individual list. This procedure constitutes an iteration in BFP procedure. After each iteration, individual list is updated and the individuals dominated are removed from the set. If the non-dominated solutions set cannot renew itself through 10 generations, then BFP procedure is terminated and the individuals existing in the non-dominated solutions set are presented as the improved version of the initial individual.

5 Computational study and results

The test instances generated [1] are used here. Three problem sets denoted by A, B, C are created to represent a variety of different environmental factors. Problem set A is formed to analyse the effect of resource based factors while fixing other factors. Set A includes multi-project cases with the same number of projects and the same number of activities but different resource requirements and resource availability levels. Each instance includes 14 projects consisting of 10 activities. Problem set B consisting of 84 instances focuses on the effects of different number of projects and activities. In these multi-project instances, three levels are set for the number of projects and seven levels are set for the number of activities. In problem set C, a multi-project environment that is heterogeneous in terms of project sizes, is emphasized by grouping projects consisting of different number of activities resulting in 27 instances.

The objectives studied are minCMAX/maxNPV. First, NSGA-II is employed to solve these sets of test problems. BFP is applied in two different modalities. In one modality, it is applied on the archive of non-dominated set of solutions obtained at the end of NSGA-II and is designated as BFP in the final stage. As an improvement routine, BFP is also applied in the intermediate stages, which is designated as BFP in the intermediate stages.

Due to space limitations, the results will be restricted to the test instance set A. In Table 1, the average values of 5 replications are presented as average Cmax (ACMAX), average NPV (ANVP), and average number of non-dominated solutions (ANS). The columns under With BFP in the final stage correspond to ACMAX, ANPV, and ANS obtained after BFP is applied to the archive of non-dominated solutions resulting from the application of NSGA-II. When comparing ACMAX and ANPV under these two different categories we observe that both objectives improve considerably. Another point to note is the reduction in ANS for the case of BFP in the final stage.

In Table 2, similar analysis is performed but this time under BFP in the intermediate stages. When we compare BFP in the intermediate stages under the category with the NSGA-II application reported in Table 1, we observe that both ACMAX and ANPV improve considerably. When we compare both categories in Table 2, then we observe slight improvement, if at all. This observation implies that improvements introduced by BFP in the intermediate stages do not leave much room for further improvement through BFP in the final stage. ANS values, on the other hand, remain approximately the same.

Another observation concerns the comparison between the ACMAX and ANPV results under the category With BFP in the final stage in Table 1 and under the category With BFP in the intermediate stages in Table 2. We observe no significant difference. We also observe an increase in the ANS values for the case of With BFP in the intermediate stages.

All the above observations and results replicate themselves for the test instance sets B and C.

Table 1: Results for Set A - BFP in the Final Stage

INSSUB	NSGA II			With BFP in the Final Stage		
	ACMAX	ANPV	ANS	ACMAX	ANPV	ANS
A11	61,39	8452,00	4,33	43,72	27448,15	1,33
A12	39,50	30205,81	2,22	29,67	65177,23	1,56
A13	38,33	35943,40	1,89	28,00	77852,86	1,00
A21	106,09	-3786,42	19,56	89,70	1313,29	13,67
A22	41,76	42472,89	1,67	31,50	91857,04	1,22
A23	39,06	46336,24	2,00	30,83	94562,36	1,11
A31	168,01	-9566,67	27,33	161,28	-9956,69	28,78
A32	45,52	31616,04	1,44	35,56	82375,22	1,00
A33	43,50	54445,20	2,22	31,70	125864,36	1,56

Table 2: Results for Set A - BFP in the Intermediate Stages

INSSUB	With BFP in the Intermediate Stages			With BFP in the Intermediate and Final Stages		
	ACMAX	ANPV	ANS	ACMAX	ANPV	ANS
A11	50,16	26130,72	3,78	43,39	27585,80	2,11
A12	29,05	68494,32	2,44	29,05	68494,32	2,44
A13	28,30	77474,89	2,44	28,30	77474,89	2,44
A21	89,65	-694,03	12,56	88,49	-773,11	12,33
A22	29,55	97170,35	2,89	29,32	97863,80	2,67
A23	29,66	100150,57	2,67	29,41	100415,73	2,33
A31	158,28	-9101,59	26,33	155,10	-9316,50	27,78
A32	35,81	85830,11	2,11	35,81	85884,65	2,11
A33	29,02	135491,89	2,33	29,06	135579,90	2,44

6 Conclusion

In this paper, we report on the results of the application of NSGA-II and the extension proposed here to a multi-objective multi-project RCPSP, where the objectives are the minimization of Cmax and the maximization of NPV. The extension proposed and tested is BFP in the final stage and BFP in the intermediate stages. Results reveal that BFP in the final stage shows significant improvement over the solution quality of pure NSGA-II. It is also revealed that BFP in the intermediate stages shows significant improvement over the solution quality of pure NSGA-II. Hence, the extension of NSGA-II through BFP has been successful. For future work, we intend to extend the work to cover further objectives such as minimization of the maximum outflow and minimization of average resource usage deviation; minimization of mean weighted tardiness of the projects in cases where due dates are assigned to projects; or minimization of mean flow time of the projects.

References

[1] A. Can and G. Ulusoy. Multi-project scheduling with 2-stage decomposition, Annals of Operations Research, 217: 95-116, 2015.

[2] K. Deb. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2): 182-197, 2002.

[3] R. Kolisch. Project Scheduling under Resource Constraints-Efficient Heuristics for Several Problem Classes, Physica-Verlag, Heidelberg, 1995.

[4] R. Kolisch. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. European Journal of Operational Research, 90: 320-333,1996.

[5] K.Y. Li and R. J. Willis. An iterative scheduling technique for resource-constrained project scheduling. European Journal of Operational Research, 56: 370-379, 1992.

[6] L. Ozdamar and G. Ulusoy. A note on an iterative forward / backward scheduling technique with reference to a procedure by Li and Willis. European Journal of Operational Research, 89: 400-407, 1996.