Journal of Quality Engineering and Production Optimization

Vol. 6, No. 1, Winter & Spring 2021

http://jqepo.shahed.ac.ir

Research Paper

DOI: 10.22070/JQEPO.2021.14832.1198

A Twofold Constructive Genetic Algorithm for Resource-Constrained Multi-Project Scheduling Problem

A. Rashidi-Pour [1] , M. Shakhsi-Niaei [1*]

[1] *Department of Industrial Engineering, College of Engineering, Yazd University, Yazd, Iran*

**** Corresponding Author:*** *M. Shakhsi-Niaei (Email: m.niaei@yazd.ac.ir)*

Abstract *– Resource-constrained multi-project scheduling problem (RCMPSP) arises in many project-based organizations, including construction and civil engineering companies. Numerous heuristic and meta-heuristic approaches have been proposed for a project scheduling problem with limited resources. In this paper, a twofold constructive genetic algorithm is proposed for the resource-constraint multiple projects scheduling problem, which benefits from a number of priority and several auxiliary rules which are fed into a serial schedule generation scheme (SGS), where auxiliary rules are used to break the tie situations where several activities have equal priority values. Numerical standard problems in different sizes are retrieved from the multi-project scheduling problem LIBRARY (MPSPLIB) website, and the numerical results are analyzed in different scenarios. Then, the genetic algorithm is used to improve the results where its parameters are tuned via Taguchi design of experiments (DOE). The results of this study showed that the performance of the proposed approach has significantly improved the solution of several problem instances and registered in the MPSPLIB.*

Keywords*– Project scheduling, Multiple projects, Priority rules, Tiebreaker, Genetic algorithm.*

I. INTRODUCTION

The resource-constrained project scheduling problem is a type of project scheduling problem that aims to minimize project completion time by considering the availability of limited resources (Demeulemeester and Herroelen, 2006). Blazewicz et al. (1983) showed that this problem is NP-hard; hence exact solution approaches fail to solve the large-sized problem instances in a timely manner. Instead, heuristic and meta-heuristic methods are used for large-size problems (Demeulemeester and Herroelen, 2006).

In heuristic methods, project activities are first prioritized according to a priority rule and then scheduled according to a scheduling scheme, e.g., serial or parallel schedule generation schemes (SGSs). When implementing priority rules, several activities may have the same priority value, which is called a tie, so some methods have been suggested to break the tie. For example, activities with equal priority values can be prioritized randomly or based on other auxiliary rules (Vázquez et al., 2015). Klein (2000) evaluated 24 priority rules for a single project problem using serial and parallel SGSs in both forward and backward directions. Browning and Yassine (2010) examined the parallel SGS with 20 different priority rules over a large number of RCMPSP instances. They used the FCFS (first come, first served) and GRES (greatest resource requirements) priority rules to break the tie and finally introduced the best priority rule for each size of the problem. Villafáñez et al. (2019) presented the P-SGS/MIN-SLK approach, which is a combination of

Manuscript Received:2- Jan -2021 & Revised:13-Feb-2021 & Accepted: 28-April-2021
ISSN: 2423-3781

parallel SGS and minimum activity total slack (MIN-SLK) as a priority rule on 140 RCMPSP instances. González et al. (2017) randomly reviewed 12 priority rules along with a random tie-breaker in 14 RCMPSP instances and reported three best methods among them. They used priority rules solutions as the initial population of the GA on and evaluated the performance of each rule by total makespan and average project delay. ElFiky et al. (2020) presented 17 priority rules on an RCMPSP case study and compared the results of each rule. Villafáñez et al. (2020) presented an algorithm to tackle the RCMPSP considering planned priorities where their approach suggests the project manager different project schedules based on initial priorities. Chen et al. (2019) designed a heuristic hybrid method and analyzed the performance of 20 priority rules for solving stochastic resource-constrained multi-project scheduling problems with new project arrivals (SRCMPSP-NPA) in which the durations of activities are stochastic. Van Eynde & Vanhoucke (2020) examined 256 different priority rules combining activity-based and project-based priority rules on both serial and parallel decoupled SGS in RCMPSP. They analyzed the performance of priority rules by two indicators, including APD & PDEL. To the best of our knowledge, none of the above studies has used auxiliary priority rules to break the prioritization tie. In other words, the activities with the same priority values have been randomly prioritized.

RCPSP and also RCMPSP are both NP-hard problems, and exact methods are not able to solve instances with more than 60 activities (Pellerin et al., 2020) because exact methods will have too long computational time for large-sized instances. In the last two decades, a growing number of metaheuristic algorithms have been presented to solve NP-hard optimization problems in order to find good (near-optimal) solutions in reasonable computational time. The Genetic Algorithm is one of the most popular population-based meta-heuristic algorithms, which was introduced by Holland (1975). Katoch et al. (2021) investigated the advantages and disadvantages of several meta-heuristic algorithms and suggested GA for several optimization applications, including project scheduling.

Linyi and Yan (2007) proposed a new particle swarm optimization using one-point crossover for the RCMPSP, where the serial SGS is used to generate the feasible solutions as the initial particles. They analyzed the performance of their model by comparing some priority rules. Tian et al. (2018) examined the RCMPSP with planned resource unavailability with an improved serial SGS and priority rules as a heuristic algorithm. They combined genetic algorithm, particle swarm optimization, and tabu search algorithm. Sonmez and Uysal (2015) combined backward-forward GA and simulated annealing to solve the RCMPSP and tested it on several portfolio instances. Ahmeti and Musliu (2021) presented a hybrid method which is a combination of constraint programming and a meta-heuristic algorithm based on local search. The meta-heuristic algorithm used in their research combines min conflicts and tabu search. Berthaut et al. (2018) proposed a hybrid metaheuristic based on scatter search (SS) that involves forward-backward improvement (FBI) and reversing the population at each iteration of the search. Uysal et al. (2021) proposed a GA-based algorithm for different sizes of RCMPSP without activity preemption. Gholizadeh-Tayyar et al. (2021) developed a GA to solve the RCMPSP instances and adjusted its parameters by Taguchi DOE. Sánchez et al. (2019) used a hybrid optimization approach that combines Ant Colony Optimization and a Hill-Climbing First Improvement algorithm to solve the RCMPSP. Peng et al. (2021b) modeled the RCMPSP as a bi-objective multi-mode scheduling problem and developed a two-phase algorithm based on NSGA-III. Nabipoor Afruzi & Aghaie (2019) considered the robust RCMPSP to deal with uncertainty in activity durations. They minimized the maximum total tardiness of the projects. Also, they proposed the adaptive bee genetic algorithm (ABGA) as a hybrid metaheuristic algorithm.

In this paper, a constructive solution of the multi-project scheduling problem with limited resources and activity preemption is generated using a number of priority and several auxiliary rules fed into a serial SGS. Then, these constructive solutions are improved by a genetic algorithm. The proposed approach has significantly improved the solution of several problem instances compared to existing rival meta-heuristic approaches. The main novelties of this paper can be summarized as:

- Considering some auxiliary rules as tie-breakers in priority rules and evaluating their performance,
- Proposing a new formulation for calculating the MS priority rule and
- forming the initial population of the proposed GA using several combined priority methods.

The rest of the paper is organized as follows. Section II describes the proposed approach, and the obtained results are shown in section III. Finally, the research is summarized and concluded in section IV.

II. RESEARCH METHOD

This section describes the research method and its details, as shown in Figure 1. The details of the relevant steps will be explained in separate subsections below.

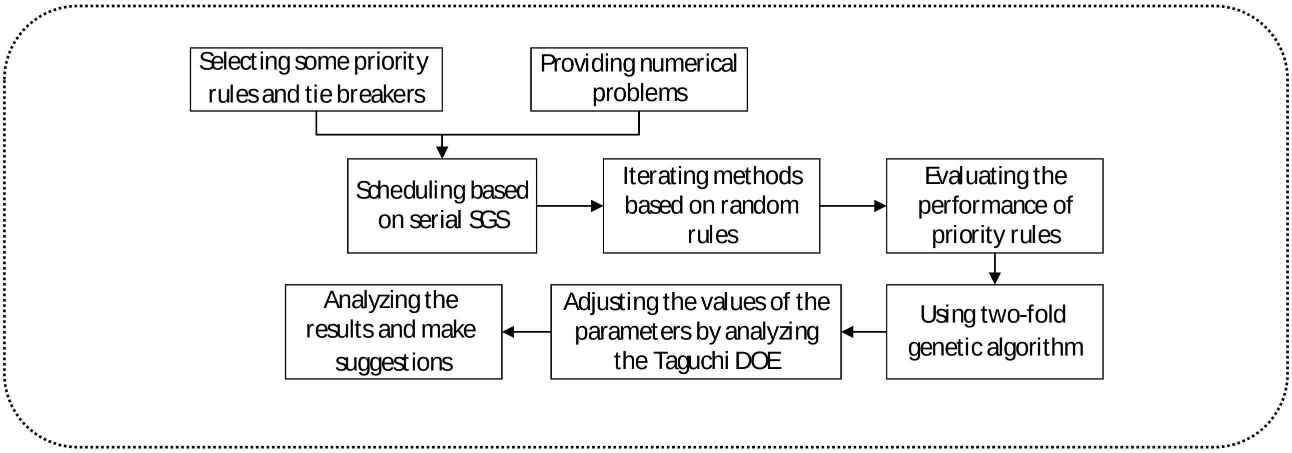

Fig. 1. The proposed method

A. Evaluating the performance of the priority rules

This section describes the first five steps shown in Figure 1. A large number of priority rules have been proposed by various researchers because there is no best-for-ever priority rule for all problems (Vázquez et al., 2015). González et al. (2017) showed that the rules of minimum late finish time (min LFT), earliest due date (EDD), and maximum successors (MS) performed better among several reviewed rules. In the EDD rule, activities that have a sooner due date are scheduled sooner. Similarly, in the Min LFT rule, activities that have a shorter late finish time are prioritized, and in the MS rule, activities that have more direct successors are scheduled earlier.

In this research, an improvement is proposed for the MS rule where the cumulative number of successors is considered instead of only direct successors. In other words, the MS value of an activity is calculated as the number of its direct successors in addition to the MS value of its successors. These calculations are conducted backward. It starts from the last project activity, where its MS value is equal to 0. As an example, Table I represents the calculated MS values of the activities shown in Figure 2.

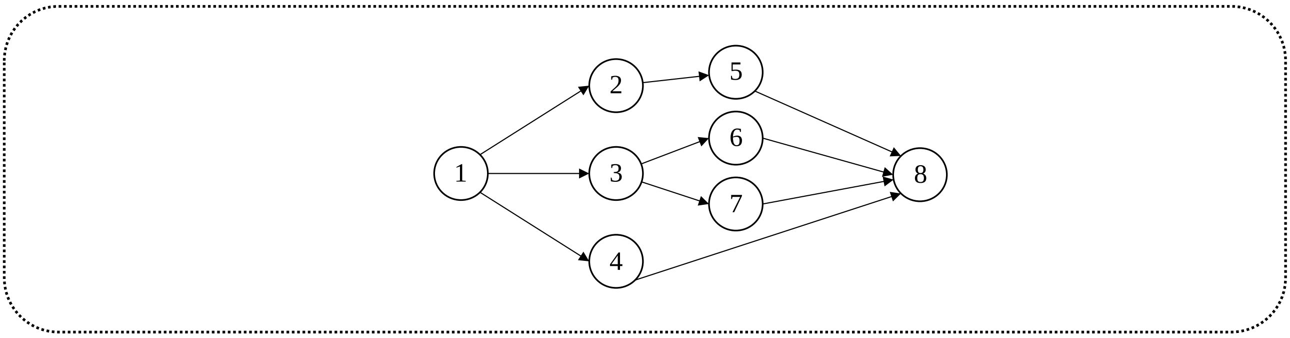

Fig. 2. Sample project

Table I. The process of calculating the MS factor for the project activities are shown in Figure 2

Activity	*Direct successor(s)*	*Number of direct successors*	*MS*
8	-	0	0
7	8	1	1+(0)=1
6	8	1	1+(0)=1
5	8	1	1+(0)=1
4	8	1	1+(0)=1
3	6 & 7	2	2+(1+1)=4
2	5	1	1+(1)=2
1	2 & 3 & 4	3	3+(2+4+1)=10

In any priority rule, other priority rules can be used to break the tie situation (Demeulemeester and Herroelen, 2006; Kolisch and Hartmann, 1999). In this paper, shortest process time (SPT), longest process time (LPT), smallest activity label (SAL), and random (RND) rules have been used as tie-breakers. Therefore, 12 twofold methods have been used to form priority lists, i.e., the combination of three main and four auxiliary/tie-breaker rules. Then, the precedence constraints are applied to the priority lists.

If the auxiliary rules cannot break the tie, i.e., the same priority values have been calculated for the relevant activities again, prioritization is determined randomly. To evaluate the performance of random methods, each problem is solved 100 times, and the average is considered as the final solution of each method.

To evaluate the performance of 12 twofold methods, 12 RCMPSP benchmark instances have been solved by each method. These problem instances are in different sizes, i.e., 2, 5, 10, and 20 projects with 32, 92, and 122 activities for each project, which are available in the multi-project scheduling problem LIBRARY on http://www.mpsplib.com. The details of these problem instances are shown in Table II.

In order to evaluate the performance of the priority rules, the RMSE index is used as Equation (1).

$$RMSE_k = \sqrt{(\sum_{p=1}^{12}(z_{k,p} - bestz_p)^2)/12} \quad (1)$$

In this equation, the index p represents various problem instances ($p = 1,2,...,12$). The $z_{k,p}$ shows the solution of problem p th by the k -th priority method. The best-observed solution of the problem p among 12 priority methods is represented by $bestz_p$.

In this paper, a serial SGS is used to schedule activities based on the priority list. In each iteration of the serial SGS, one activity is selected from the priority list, and the earliest possible time in the schedule is assigned to this activity, considering resource constraints and precedence relations. This process continues until all activities are scheduled and the last activity's finish time is considered the solution of the problem, i.e., the completion time of the project. Kim and Ellis (2010) analyzed the serial and parallel SGSs and showed the better performance of serial SGS due to its active scheduling solutions and shorter computational time.

Table II. List of benchmark problem instances

Problem name	*Number of activities in each project (j)*	*Number of projects*	*Total activities*
mp_j30_a2_nr2	32	2	64
mp_j30_a5_nr2	32	5	160
mp_j30_a10_nr3	32	10	320
mp_j30_a20_nr2	32	20	640
mp_j90_a2_nr3	92	2	184
mp_j90_a5_nr2	92	5	460
mp_j90_a10_nr3	92	10	920
mp_j90_a20_nr2	92	20	1840
mp_j120_a2_nr3	122	2	244
mp_j120_a5_nr2	122	5	610
mp_j120_a10_nr3	122	10	1220
mp_j120_a20_nr2	122	20	2440

B. Proposed twofold constructive genetic algorithm

GA is widely used for maintaining high-quality reactions to optimize issues and problems investigation (Alam et al., 2020). More specifically, many researchers have implemented GA variants for different types of project scheduling problems, e.g., Peng et al. (2021a&b), Khalilzadeh et al. (2020), Rahman et al. (2020), Zhang and Cui. (2021), and Gholizadeh-Tayyar et al. (2021). In this paper, we implemented GA as a widely used metaheuristic algorithm which has resulted in acceptable results compared to state-of-the-art existing metaheuristic algorithms.

Figure 3 shows the stages of the proposed GA. The best solutions obtained from the 12 constructive methods in addition to randomly generated chromosomes will form the initial population of the genetic algorithm. The fitness value of each chromosome is calculated as the project completion time based on the priority list and the serial SGS. The selection operator is based on a roulette-wheel selection mechanism. Two-point crossover and the swap mutation operators are used as crossover and mutation operators. The algorithm terminates when a predetermined number of generations has been produced.

The effective parameters of the genetic algorithm, including population size, number of generations, probability of crossover, and mutation, can affect the final solutions. In this paper, the probability of mutation operator is calculated by Equation (2) which is proposed by Hassanat et al. (2019).

$$mutation\ probability = 1 - crossover\ probability \tag{2}$$

Therefore, by setting the value of the crossover probability parameter, the value of the mutation probability parameter is also determined, vice versa.

Taguchi DOE is used to determine the best values for the parameters. Table III shows the different factors and alternative levels for each factor in the Taguchi method.

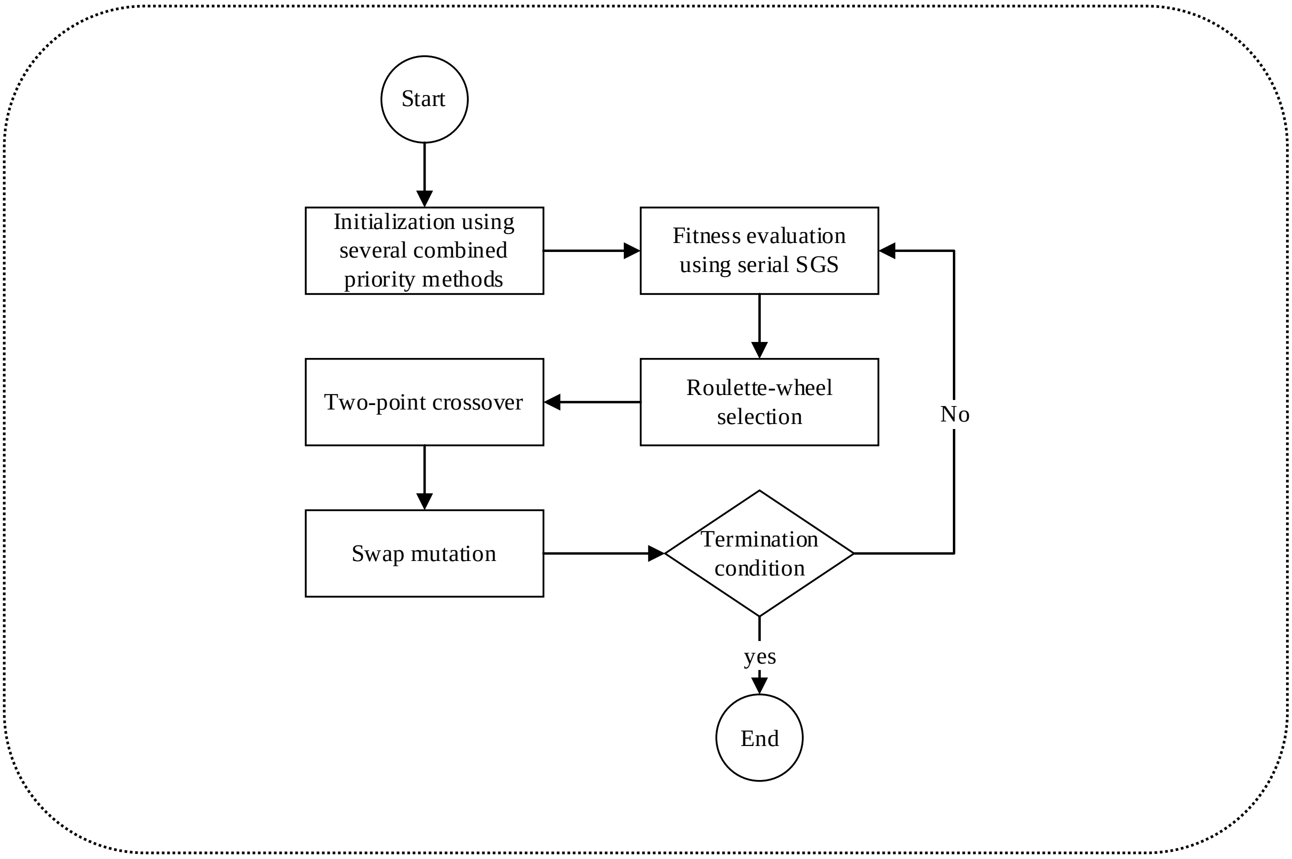

FIG. 3. PROPOSED TWOFOLD GA

Table III. Factors and their levels in the Taguchi DOE

Factors	***Population size***	***Generations***	***Crossover probability***
Levels	20 – 35 – 50	100 – 150 – 200	0.2 – 0.5 – 0.8

As shown in Table III, three factors are determined at three different levels, and three replications are carried out for each experiment. Therefore, each problem is solved 27 times by the genetic algorithm. After analyzing the diagrams of Taguchi method, the best value of each parameter is obtained, as shown in section IV. It should be noted that Minitab software has been used to design and analyze the Taguchi experiments.

III. NUMERICAL RESULTS

This section discusses the results of the research method steps mentioned in the previous section. Table IV shows the value of the RMSE index for the 12 twofold priority methods. The lower value of the RMSE indicates the better performance of the method. According to Table IV, methods 10 and 12 have the lowest RMSE and therefore have the best overall performance.

Table IV. The value of the RMSE index for 12 methods

Twofold priority methods	*Description*		*RMSE*
	Main rule	*Auxiliary rule*	
1	LFT	SPT	8.997
2	LFT	LPT	8.862
3	LFT	SAL	9.342
4	LFT	RND	8.994
5	EDD	SPT	7.340
6	EDD	LPT	7.470
7	EDD	SAL	7.364
8	EDD	RND	7.405
9	MS	SPT	5.217
10	MS	LPT	4.871
11	MS	SAL	5.470
12	MS	RND	4.799

The values of the RMSE index of 12 priority methods are shown in descending order in Figure 4. As it turns out, the values of this index can be categorized into three groups. Methods 9 to 12 work best which are related to the MS priority rule.

For further investigation, the number of times that each priority method has deviated more than 5% from the best-observed solution has been counted for each method. Figure 5 represents the number of more-than-5% deviations. The first three diagrams of each method (blue, red, and yellow) are related to the problems with the number of 32, 92, and 122 activities, respectively, while the last diagram shows the total more-than-5% deviations. Obviously, a method with a lower number of significant deviations has a better performance. As shown in this figure, methods 9, 10, 11, and 12 had the lowest number of significant deviations from the best solution and so performed better.

The results of Taguchi DOE are shown in the form of a means plot and signal-to-noise-ratios plot. As an example, figure 6 shows these plots for the problem mp_j90_a10_nr3.

The initial population number, the number of generations, and the crossover probability are respectively represented by *popsize*, *maxiter* and *pc* symbols.

According to the signal-to-noise-ratio plot, any factor level that has the highest value is the best level of that parameter (factor). The two plots in Figure 6 are inversely related. Thus, in the means plot, unlike signal-to-noise ratios, the level that has the lowest value is considered as the best parameter level. It should be noted that if several levels of a parameter are reported as the best levels, one of them is randomly selected because there will be no change in the final solution.

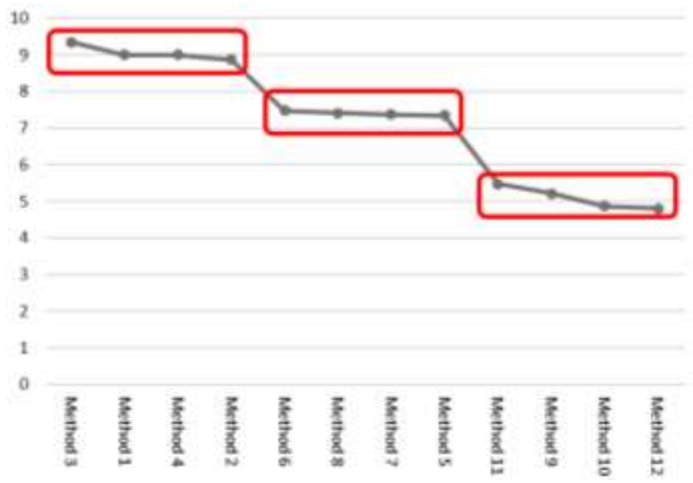

Fig. 4. The value of the RMSE in different priority methods

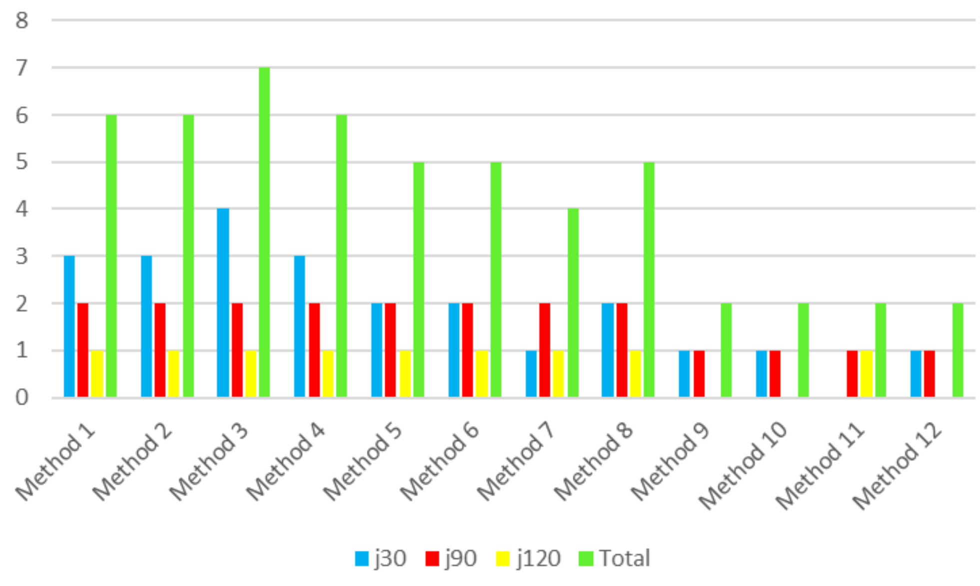

Fig. 5. The number of significant deviations

The best values of the parameters for the above-mentioned example are shown in Table V. It should be noted that in the number of generations parameter (*maxiter*) both 100 and 200 generations produce the best answer; therefore, 100 generations are randomly selected.

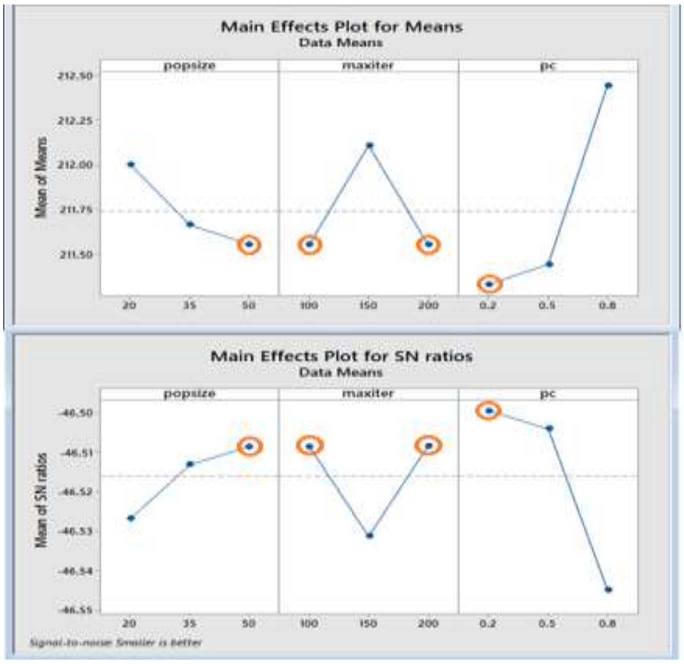

Fig. 6. Main effects plots for means (above) and signal to noise ratios (below)

Table V. The best combination of genetic algorithm parameters in problem mp_j90_a10_nr3

Parameter	Level value
popsize	50
maxiter	100
pc	0.2

The same process has been conducted for all numerical problem instances.

After setting the values of the genetic algorithm parameters of each problem, the numerical problem instances have been solved using the genetic algorithm.

Table VI shows the results of the proposed twofold constructive genetic algorithm, compared to the three best available solutions of the MPSPLIB website, where the bold numbers indicate that the proposed genetic algorithm has attained or improved the best available solutions. As shown in Table VI, in 6 cases, better results are obtained, and in 2 cases, our solution is equal to the best available solutions. It should be noted that in order to validate our solutions, they are uploaded on http://www.mpsplib.com under the name RashidiPour/Shakhsi-Niaei, which is validated and then registered. Figure 7 compares the results of this study with the top three solutions on the MPSPLIB website.

Table VI. genetic algorithm solutions compared to the three best MPSPLIB solution

Problem	*Total makespan (TMS) obtained in this research*	*Three best TMSs on the MPSPLIB website*			*Computational time (seconds)*
		Best	*Second*	*Third*	
mp_j30_a2_nr2	59	58	59	59	133
mp_j30_a5_nr2	78	79	79	79	418
mp_j30_a10_nr3	242	97	242	243	680
mp_j30_a20_nr2	277	278	282	283	2000
mp_j90_a2_nr3	114	114	114	114	95
mp_j90_a5_nr2	114	114	114	114	325
mp_j90_a10_nr3	210	213	213	213	2460
mp_j90_a20_nr2	163	164	164	168	8969
mp_j120_a2_nr3	278	273	275	277	367
mp_j120_a5_nr2	179	164	165	176	2403
mp_j120_a10_nr3	138	142	142	142	2460
mp_j120_a20_nr2	202	204	204	207	9220

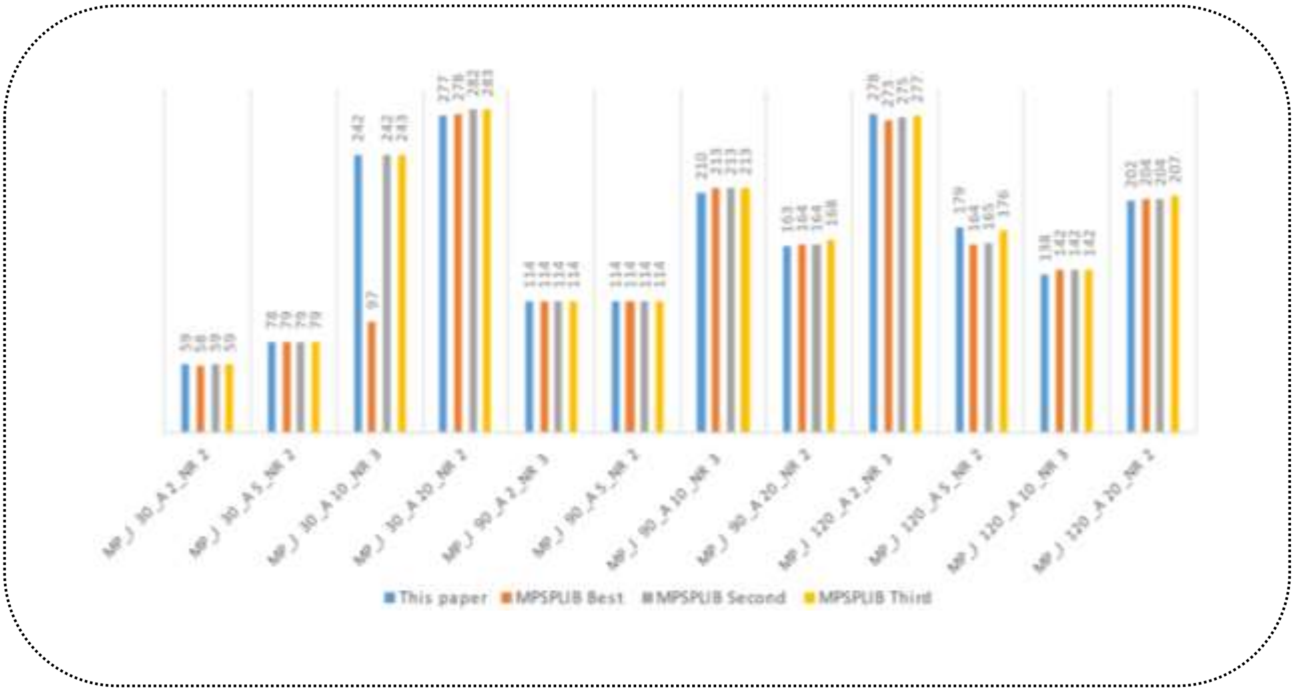

Fig. 7. Comparison of the proposed GA and the three best solutions of MPSPLIB

Figure 8 illustrates the graphical representation of the solution for the problem mp_j30_a2_nr2 generated by the MPSPLIB website.

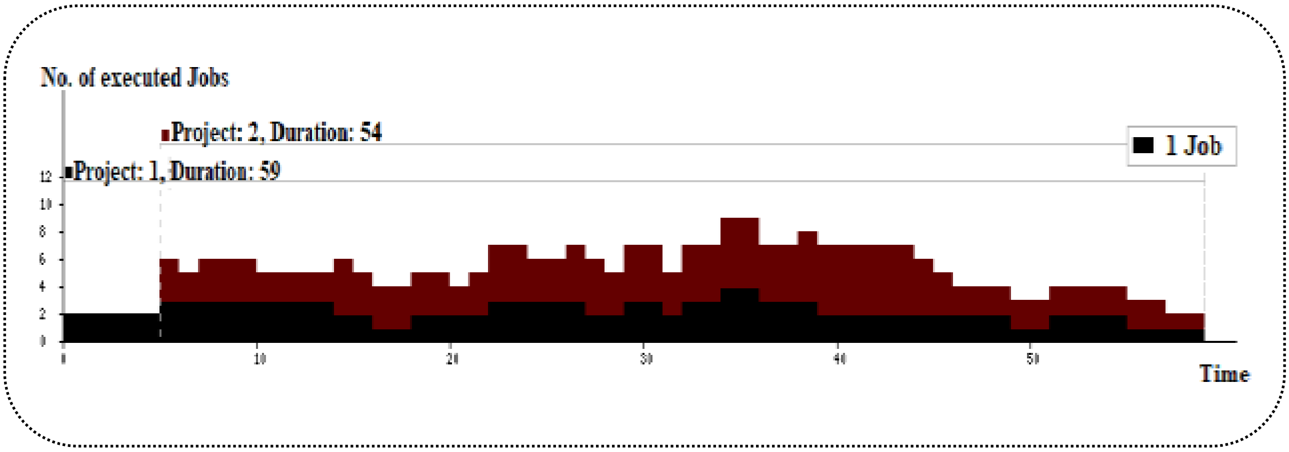

FIG. 8. UPLOADED ANSWER OF PROBLEM mp_j30_a2_nr2

For further investigation, the results of this study (TMS) are compared with the results of three rival methods shown in Table VII, i.e., PSGSMINSLK (Villafáñez, 2019), ACO, and ACO+HC+SMT (Sánchez et al., 2019). As shown in this table, the proposed GA has attained/outperformed the best result of three rival approaches in 10 out of 12 problem instances.

Table VII. genetic algorithm solutions compared to three rival approaches

Problem	*PSGSMINSLK*	*ACO*	*ACO+HC+SMT*	*This research*
mp_j30_a2_nr2	61	64	NA	59
mp_j30_a5_nr2	80	NA	81	78
mp_j30_a10_nr3	243	252	248	242
mp_j30_a20_nr2	278	NA	284	277
mp_j90_a2_nr3	115	115	114	114
mp_j90_a5_nr2	114	NA	NA	114
mp_j90_a10_nr3	213	NA	NA	210
mp_j90_a20_nr2	168	NA	167	163
mp_j120_a2_nr3	**275**	NA	NA	278
mp_j120_a5_nr2	181	190	**178**	179
mp_j120_a10_nr3	146	152	145	138
mp_j120_a20_nr2	221	NA	218	202

IV. CONCLUSION

In this study, the performance of some twofold priority rules for the resource-constrained multi-project scheduling problem is investigated using a serial schedule generation scheme. For this purpose, 12 twofold priority methods are formed based on three main priority rules, including min LFT, EDD, and MS, and four auxiliary priority rules as tie-breakers, including SPT, LPT, SAL, and RND. The performance of these methods is evaluated using the RMSE index and the number of significant deviations from the best-obtained solutions. Then, 12 standard problem instances with different sizes with 64 up to 2440 activities in the whole project portfolio have been tested. Based on the results of this analysis, it is suggested to use the proposed MS as the main rule and LPT or RND as the auxiliary rule to break the tie situations in the MS method. Considering the number of deviations from the best answer, methods related to the proposed MS priority rule produce the best answer, including:

- Maximum successors as main and shortest process time as an auxiliary rule,
- Maximum successors as main and longest process time as an auxiliary rule,
- Maximum successors as main and smallest activity label as an auxiliary rule, and
- Maximum successors as main and random as an auxiliary rule.

After comparing the performance of the described priority methods, a genetic algorithm is used to improve the solutions of problem instances. In order to set the GA parameters, 27 experiments for each problem are designed using Taguchi DOE. Then, the best values of these parameters are obtained.

In order to evaluate the quality of the results, the following comparisons have been conducted:

- The three best results available at the MPSPLIB website have been compared with those of this research. As shown in table VI, the present study has improved the best available solutions in 6 problems and also in 2 problems has obtained the best available solutions.
- The results of three state-of-the-art methods, i.e., PSGSMINSLK (Villafáñez, 2019), ACO, and ACO+HC+SMT (Sánchez et al., 2019) have been compared with those of this research. As shown in table VII, the proposed approach has attained/outperformed the best result of three rival approaches in 10 out of 12 problem instances.

For future research, the parallel SGS can be used in similar analyzes, and the results can be compared. Moreover, other priority rules can be compared with the best method in the present study. Finally, other meta-heuristic algorithms can be implemented and be compared.

REFERENCES

Ahmeti, A., & Musliu, N. (2021). Hybridizing Constraint Programming and Meta-Heuristics for Multi-Mode Resource-Constrained Multiple Projects Scheduling Problem. *Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling-PATAT*, 188–206.

Alam, T., Qamar, S., Dixit, A., & Benaida, M. (2020). Genetic Algorithm: Reviews, Implementations, and Applications. International Journal of Engineering Pedagogy, 10(6), 57–77.

Berthaut, F., Pellerin, R., Hajji, A., & Perrier, N. (2018). A path relinking-based scatter search for the resource-constrained project scheduling problem. *International Journal of Project Organisation and Management*, 10(1), 1–36.

Blazewicz, J., Lenstra, J. K., & Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete applied mathematics*, 5(1), 11–24.

Browning, T. R., & Yassine, A. A. (2010). Resource-constrained multi-project scheduling: Priority rule performance revisited. International Journal of Production Economics, 126(2), 212–228.

Chen, H., Ding, G., Zhang, J., & Qin, S. (2019). Research on priority rules for the stochastic resource constrained multi-project scheduling problem with new project arrival. *Computers & Industrial Engineering*, 137, 106060.

Demeulemeester, E. L., & Herroelen, W. S. (2006). *Project scheduling: a research handbook*. Springer Science & Business Media.

ElFiky, H., Owida, A., & Galal, N. M. (2020). Resource Constrained Multi-Project Scheduling Using Priority Rules: Application in the Deep-Water Construction Industry. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 353–363.

Gholizadeh-Tayyar, S., Okongwu, U., & Lamothe, J. (2021). A Heuristic-Based Genetic Algorithm for Scheduling of Multiple Projects Subjected to Resource Constraints and Environmental Responsibility Commitments. *Process Integration and Optimization for Sustainability*, 1–22.

González, G., Calderón, C., González, D., & Trujillo, J. (2017). Priority Rules Performance Analysis as a GA Initial Population for RCMPSP. *Proceedings of International Conference on Industrial Engineering and Operations Management*, 1201–1211.

Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., & Prasath, V. B. (2019). Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information*, 10(12), 390–425.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80, 8091–8126.

Khalilzadeh, M., Hosseini, S., & Ghaeli, R. (2020). A genetic algorithm-based method for solving multi-mode resource-constrained project scheduling problem in uncertain environment. *Journal of Project Management*, *5*(2), 79–86.

Kim, J. L., & Ellis Jr, R. D. (2010). Comparing schedule generation schemes in resource-constrained project scheduling using elitist genetic algorithm. *Journal of construction engineering and management*, 136(2), 160–169.

Klein, R. (2000). Bidirectional planning: improving priority rule-based heuristics for scheduling resource-constrained projects. *European Journal of Operational Research*, 127(3), 619–638.

Kolisch, R., & Hartmann, S. (1999). Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis. In *Project scheduling* (pp. 147–178). Springer, Boston, MA.

Linyi, D., & Yan, L. (2007). A particle swarm optimization for resource-constrained multi-project scheduling problem. *Proceedings of International Conference on Computational Intelligence and Security (CIS 2007)*, 1010–1014.

Nabipoor Afruzi, E., & Aghaie, A. (2019). A Hybridized Metaheuristic Algorithm to Solve the Robust Resource Constrained Multi-Project Scheduling Problem. *AUT Journal of Modeling and Simulation*, 51(1), 15–32.

Pellerin, R., Perrier, N., & Berthaut, F. (2020). A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 280(2), 395–416.

Peng, W., Mu, J., Chen, L., & Lin, J. (2021a). A novel non-dominated sorting genetic algorithm for solving the triple objective project scheduling problem. *Memetic Computing*, *13*(2), 271–284.

Peng, W., Zhang, J., & Chen, L. (2021b). A bi-objective hierarchical program scheduling problem and its solution based on NSGA-III. *Annals of Operations Research*, Article in press.

Rahman, H. F., Chakrabortty, R. K., & Ryan, M. J. (2020). Memetic algorithm for solving resource constrained project scheduling problems. *Automation in Construction*, *111*, 103052.

Sánchez, M. G., Gil, A. F., & Castro, C. (2019). Integrating a SMT solver based local search in ant colony optimization for solving RCMPSP. *Proceedings of IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 1–6.

Sonmez, R., & Uysal, F. (2015). Backward-forward hybrid genetic algorithm for resource-constrained multiproject scheduling problem. *Journal of Computing in Civil Engineering*, 29(5), 04014072-1–04014072-9.

Tian, J., Dong, X., & Han, S. (2018). Optimizing for a Resource-constrained Multi-project Scheduling Problem with Planned Resource Unavailability. *Proceedings of 3rd International Conference on Modelling, Simulation and Applied Mathematics (MSAM 2018)*, 243–248.

Uysal, F., Sonmez, R., & Isleyen, S. K. (2021). A graphical processing unit‐ based parallel hybrid genetic algorithm for resource-constrained multi‐ project scheduling problem. Concurrency and Computation: Practice and Experience, *Article in press*.

Van Eynde, R., & Vanhoucke, M. (2020). Resource-constrained multi-project scheduling: benchmark datasets and decoupled scheduling. *Journal of Scheduling*, 23(3), 301–325.

Vázquez, E. P., Calvo, M. P., & Ordóñez, P. M. (2015). Learning process on priority rules to solve the RCMPSP. *Journal of Intelligent Manufacturing*, 26(1), 123–138.

Villafáñez, F., Poza, D., López-Paredes, A., Pajares, J., & del Olmo, R. (2019). A generic heuristic for multi-project scheduling problems with global and local resource constraints (RCMPSP). *Soft Computing*, 23(10), 3465–3479.

Villafáñez, F., Poza, D., López-Paredes, A., Pajares, J., & Acebes, F. (2020). Portfolio scheduling: an integrative approach of limited resources and project prioritization. *Journal of Project Management*, 5(2), 103–116.

Zhang, Y., & Cui, N. (2021). Project scheduling and material ordering problem with storage space constraints. Automation in Construction, 129, 103796.