Evolutionary Algorithm for Project Scheduling under Irregular Resource Changes

Forhad Zaman, Saber Elsayed, Ruhul Sarker, Daryl Essam
School of Engineering and Information Technology
University of New South Wales
Canberra, Australia
{f.zaman, s.elsayed, r.sarker, d.essam}@unsw.edu.au

Carlos A. Coello Coello
Depto. de Sistemas
UAM-Azcapotzalco
Mexico
ccoello@cs.cinvestav.mx

***Abstract*—Over the last few decades, project scheduling problems have been solved under a set of resource constraints, which are assumed fixed throughout the project horizon. However, in real-life applications, resources may change over time due to maintenance or because the resources are needed for another project. Therefore, this research introduces a hybrid evolutionary framework, based on two multi-operator evolutionary algorithms, and a heuristic technique, for a multi-mode project scheduling under irregular resources changes. The framework simultaneously considers both algorithms and self-adaptively emphasizes the one which performs comparatively better. The heuristic considers two variants of handling techniques for irregular resources. One is based on inserting buffer activities to characterize resources unavailable, and another is based on a modified serial generation scheme, that determines the best modes of the activities at each time period based on irregular resources. The framework is tested by solving a set of test problems, with the renewable resources considered irregular over the project horizon. The results demonstrate that the multi-method algorithm has some advantages for scheduling a project, under both regular and irregular resources.**

I. Introduction

Over the last few decades, resource-constrained project scheduling problems (RCPSPs) have been widely studied by researchers and practitioners, due to their importance in many engineering applications [1]. The objective of an RCPSP is to schedule the project activities in a way that minimizes the overall completion time of the project, bearing in mind the satisfaction of resources and precedence constraints.

In standard (also called single-mode) RCPSPs, to implement an activity, a combination of resources for a certain duration is needed. On the other hand, in multi-mode RCPSP (MM-RCPSP), an activity can operate any executive mode among several non-preemptive modes. Each mode of activity represents different requirements of the combinations of resources and duration. This means that we need to determine the best schedule of the activities and their modes which can minimize the make-span while satisfying the precedence and resource constraints [2].

Over the years, a number of solution approaches have been developed for the single- and MM-RCPSPs. They can be broadly classified into three types: (i) mathematical based exact algorithms, (ii) heuristic-based solution approaches, and (iii) meta-heuristic based algorithms. From the many existing exact algorithms, branch and bound, branch and cut, a linear program, and mixed integer linear program are commonly used to solve many RCPSPs [2]. Although they are accurate and fast to converge, these algorithms are limited to small-scale RCPSPs. In fact, a few exact algorithms are available for MM-RCPSPs, such as tree-based branch and bound, constraint programming, and mixed integer linear programming based algorithms [3]. However, these algorithms are computationally expensive and only solve MM-RCPSPs with up to 20 activities.

The heuristic based algorithms are effective to solve the medium size RCPSP. For each problem, a dedicated heuristic is developed based on the different priority rules. For example, Zheng and Wang [4] developed an agent-based heuristic, where each agent represents a partial schedule of the activities which are iteratively updated towards an optimal solution. Liu et. al [5] developed an activity-list-based partitions algorithm where a local search is used to improve the quality of schedule. Chakrabortty et. al [6] proposed a solution framework based on six heuristic approaches that are used to solve RCPSPs with uncertain activity duration. However, solving MM-RCPSP using the heuristic approach is limited and computationally expensive.

Nowadays, many meta-heuristic algorithms [7], such as genetic algorithms (GAs) [8], differential evolution (DE) [9] and particle swarm optimization (PSO) [10] are used to solve both single mode and MM-RCPSPs. Although they are efficient and flexible to solve different types of RCPSPs, they are computationally expensive for large-scale problems. To overcome this drawback, a few hybrid methods that combine a meta-heuristic and a heuristic approach have been proposed. For example, Peteghem and Vanhoucke [11] proposed a hybrid GA that is empowered with a local heuristic, Coelho and Vanhoucke [12] proposed a two-stage solution approach, in which a first-stage is used to determine the best set of modes and the second-stage is used to schedule the activities, respectively. Another popular hybrid approach is to use a serial generation scheme (SGS) with a meta-heuristic algorithm [13]. In such approach, SGS reorders the activities based on their earliest starting and

This research has been funded by Australian Research Council Discovery Project DP170102416. The last author is on sabbatical leave from CINVESTAV-IPN. He gratefully acknowledges support from CONACyT grant no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*).

finishing times while satisfying their precedence and resource constraints. However, no algorithm performs better for a wide range of MM-RCPSPs.

Some multi-method algorithms consider two or more meta-heuristic algorithms in a framework, such as Elsayed et. al [14] and Zaman et. al [2] who developed a solution approach for solving single mode and MM-RCPSPs, respectively. Both frameworks are based on two meta-heuristic algorithms such as a multi-operator GA (moGA) and a multi-operator DE (moDE), and tested them by solving the well-known standard benchmark problems from PSPLIB [15]. It is found that the multi-method based algorithm performs better than a single algorithm.

The above-mentioned approaches solve the RCPSPs under a common assumption that the available resources are constant over the project horizon. In practice, some of the resources may not be available for a certain time period due to their maintenance or being used for a sister project. Therefore, the available resources become irregular over the project.

To the best of our knowledge, no algorithm solved RCPSPs under the irregular resources. Therefore, in this paper, we reformulate the MM-RCPSP taking into account resources may change over time. To solve this highly complex and non-smooth problem, a hybrid multi-method based evolutionary algorithm (H-MEA) is designed. It consists of two multi-operator evolutionary algorithms (moEAs), a linear programming approach and a SGS. Two moEAs are used to generate new individuals of a schedule of the activities ($\vec{x}$) and their modes ($\vec{y}$). The linear-programming approach rectifies any infeasible $\vec{y}$ to a feasible $\vec{y}$, while a modified SGS is used to obtain a good quality $\vec{x}$ by satisfying the irregular resources capacity and precedence constraints.

To handle irregular resources, two new approaches, such as (i) a buffer activity insertion (BAI), and (ii) an available resource-based scheduling (ARS), are developed. The BAI technique deliberately inserts some buffered activities in the network to portray the maintenance of the resources. On the contrary, the ARS does not add any additional activity, rather schedules the project based on the irregular resource changes. Both approaches have been tested using standard test problems taken from literature. From the analysis, it is found that the H-MEA with ARS has some merits in terms of flexibility and efficiency.

The rest of the paper is organized as follows: Section II describes the formulation of MM-RCPSP under irregular resources, Section III presents the proposed solution approach, Section IV shows the experimental results, and Section V states the conclusion and recommendations for future works.

II. Problem Description

MM-RCPSP is formulated for a single project that has a number of activities $A = \{1, 2, ..., D+2\}$, where D is the number of non-dummy activities of the project. The activity '1' and '(D+2)' are two dummy activities, those represent *start* and *finish* of the project. Each activity requires a number of renewable and non-renewable resources for a certain duration, which is subject to an operating mode. Each project has several execution modes, and each mode represents different requirements of the resources and duration.

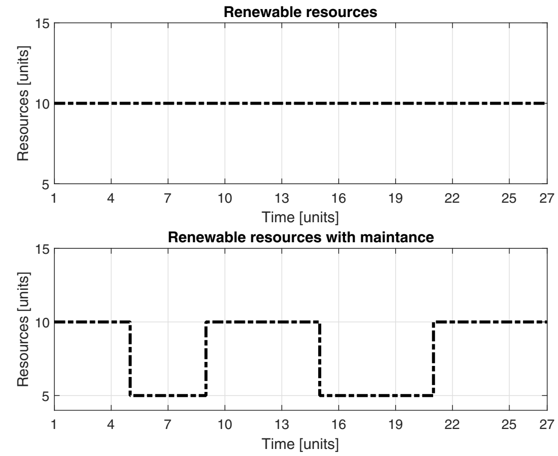

Fig. 1. Resources with and without considering any maintenance

The maximum number of resources are given for a project. The renewable resources of a project such as machines are repeatedly used throughout the project horizon. On the contrary, the non-renewable resources such as money cannot be repeated, and once a unit is used by an activity $j \in A$, that cannot be used by the other activities. The objective of a MM-RCPSP is to minimize the project duration by scheduling the activities with determining their best modes subject to resource and precedence constraints [1].

In this research, we consider the irregular availability of renewable resources over the project horizon. In other words, all the resources cannot be repeated at every time period, rather some of them are unavailable due to maintenance, and sharing with any other project, etc. The top graph in Fig. 1 shows the traditional availability of renewable resources, and the bottom one shows the irregular resources that are considered in this paper. It is seen that the total availability of the resources is not constant, but irregular which is more practical. Our objective is to minimize the project duration by scheduling the activities under the irregular resources. The mathematical model is shown below.

$$\min : f = FT_{(D+2),m},\, m \in M \tag{1}$$

Subject to:

$$FT_{i,m} \leq FT_{j,m} - d_{j,m}, \forall\, (j,i) \in A, \forall m \in M \tag{2}$$

$$\sum_{j \in A_t} r_{j,m,k} \leq R_{k,t}, \forall\, (j,i) \in A_t,\, \forall k \in K,\, \forall m \in M \tag{3}$$

$$\sum_{j=2}^{D} v_{j,m,n} \leq V_n, \forall m \in M, \forall n \in N \quad (4)$$

Eqn. (1) represents the objective function of the optimization problem which is to minimize the finish time (FT) of the *finish* dummy activity. The constraint in Eqn. (2) represents the temporal relationship between the activities, which means an activity cannot start until its predecessors finish, where $d_{j,m}$ is for the duration of the j^{th} activity at m^{th} mode.

Eqn. (3) represents the capacity constraint for the renewable resources, where $r_{j,m,k}$ is the required number of k^{th} renewable resources of the j^{th} activity at m^{th} mode, and $R_{k,t}$ is the available k^{th} renewable resources at t^{th} time period.

Eqn. (4) is for the capacity constraint of the non-renewable resource, where $v_{j,m,n}$ is for the required non-renewable resources of the j^{th} activity at m^{th} mode, and V_n is the total n^{th} type of non-renewable resources.

III. Proposed Approach

In this research, we propose a H-MEA based on two moEAs and two heuristics, for the MM-RCPSP under irregular resources. Its pseudo-code is shown in Algorithm 1.

It starts with an initial random population that contains random schedules of the activities, $\vec{x}_i$ and their modes, $\vec{y}_i$, $i = 1, 2, ..., N_P$. As both $\vec{x}_i$ and $\vec{y}_i$ may be infeasible, they are rectified using the two heuristics. The first heuristic is based on a linear programming approach that converts any infeasible $\vec{y}_i$ to a feasible one. While, the second heuristic is based on the modified forward and backward SGSs, that converts an infeasible $\vec{x}_i$ to a good quality feasible $\vec{x}_i$. To handle irregular resources, we use two techniques, namely (i) BAI, and (ii) ARS, those are discussed in subsection III-C.

Once feasible $\vec{x}_i$ and $\vec{y}_i$ are obtained, f_i is determined based on the Eqn. (1). Their offspring are generated based on the two moEAs. The number of offspring from a moEA is varied, which depends on its performance in previous generations. The one which generates better quality offspring than their parents is said to be better, and a higher number of offspring is generated using that algorithm. As the offspring of both $\vec{x}_i$and $\vec{y}_i$ may be infeasible, they are handled using the two heuristics. This process is continued until a predefined maximum number of generations is reached.

A. Pre-processing and Initial Generation

During the initial generation, we firstly do some data pre-processing. In it, we delete some infeasible modes, as they are not realistic. Generally, the requirements of the resources and duration of all non-preemptive modes for an activity are not comparable, which means no mode cannot be considered better than others. For example, one mode may have a smaller duration than other modes, but might have higher resources. However, if any problem contains akin modes, we delete the worst mode before starting the optimization process. For example, Fig. 2 shows two different modes of an activity, those are depicted using two pentagons with solid and dashed lines.

Algorithm 1 Pseudo-code of H-MEA

Require: Maximum number of generations N_G, and population size N_P.

Generate randomly both $\vec{x}_i$ and $\vec{y}_i, \forall i = 1, 2, ..., N_P$ using Eqn. (5), and (6), respectively.
for $g = 1; g \leq N_G; g{+}{+}$ **do**
  for $i = 1; i \leq N_P; i{+}{+}$ **do**
    Obtain feasible $\vec{y}_i$ using the linear-programming-based heuristic, as discussed in subsection III-B.
    Based on the new $\vec{y}_i$, determine $r_{j,m,k}$ and $v_{j,m,n}, \forall k, j = 1, 2, ..., D + 2$.
    Obtain a quality $\vec{x}_i$ using the one of the two handling techniques for the irregular resources, as discussed in subsection III-C.
    Calculate f_i as shown in Eqn. (1).
  end for
  Using the moEAs, generate new offspring of both $\vec{x}_i$ and $\vec{y}_i$, $\forall i = 1, 2, ..., N_P$, as discussed in subsection III-D.
  Based on the f_i, select the best N_P individuals of both $\vec{x}_i$ and $\vec{y}_i$ from their parents and offspring.
end for

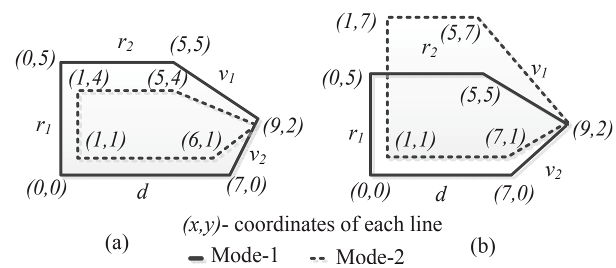

Fig. 2. An example of (a) comparable, and (b) non-comparable modes

Each pentagon represents a combination of the requirements of duration (d), two renewable (r_1 and r_2), and two non-renewable (v_1 and v_2) resources. The line length of each pentagon indicates the units of resource and duration are required in that mode. It is seen from Fig. 2(a), mode-2 is better than mode-1 as both duration and resources are minimum in mode-2. Therefore, we delete the mode-1 before starting the optimization process. On the contrary, if we consider Fig. 2(b), it is seen that both modes are non-comparable, as mode-2 has a lower d but higher r_1 comparing to mode-1. Therefore, we keep them in the optimization process.

Once removing the akin modes, we randomly generate $\vec{x}_i$ and $\vec{y}_i$ for the schedule of the activities and their modes, respectively, as:

$$\vec{x}_i = \mathbb{N} \cap \{x_{i,1}, x_{i,2}, \ldots, x_{i,D+2}\} \quad (5)$$

$$\vec{y}_i = \{m_{i,1}, m_{i,2}, \ldots, m_{i,D+2}\} = \mathbb{N} \cap [1, M_j] \quad (6)$$

The $\vec{x}_i$ and $\vec{y}_i$, $i = 1, 2, \ldots, N_P$ permutations of the activities and a corresponding random mode, respectively. The decision variable for the H-MEA is discrete, which is the conjunction of both $\vec{x}_i$ and $\vec{y}_i$, as:

$$\vec{z}_i = \{\vec{x}_i, \vec{y}_i\}, i \in N_P \tag{7}$$

where $\vec{z}_i$ is the i^{th} individual of a population, and the size of the decision variable is $2(D+2)$.

B. Heuristics

In this research, we use two-step heuristic procedures to rectify infeasible individuals to feasible ones. The first heuristic is based on a linear programming approach that converts any infeasible $\vec{y}_i$ to a feasible $\vec{y}_i$. To do this, firstly the $\vec{y}_i$ is encoded to a binary vector, as:

$$u_{i,j,m} = \begin{cases} 1 & \text{when } m = y_{i,j} \\ 0 & \text{otherwise} \end{cases}, \forall m = 1, 2, \ldots, M_j \tag{8}$$

The infeasible binary vector $\vec{u}_i$ is rectified by solving the following linear-programming model using the branch and bound algorithm. We use *intlinprog* function in Matlab to solve the following model, as [2]:

$$\min : \sum_{j=2}^{D+1} \sum_{m=1}^{M_j} rand_{i,j,m} d_{j,m} u_{i,j,m}, rand_{i,j,m} \in \{0, 1\} \tag{9}$$

Subject to:

$$\sum_{m=1}^{M_j} u_{i,j,m} = 1, \forall j \in D, \, u \in \{0, 1\} \tag{10}$$

$$\sum_{j=2}^{D+1} \sum_{m=1}^{M_j} u_{i,j,m} \times v_{j,m,m} \leq V_n, \forall n \in N \tag{11}$$

$$\sum_{m=1}^{M_j} u_{i,j,m} \times r_{j,m,k} \leq R_{k,t}, \forall k \in K, \forall j \in D \tag{12}$$

Eqn. (9) is the objective function of the linear-programming approach that minimizes the sum of the duration of all activities at their execution mode, where $rand_{i,j,m}$ is a random number between 0 and 1, that is used to generate a different feasible solution of $\vec{u}_i$ at each run. Eqn. (10) ensures an activity operates in only one mode where $u_{i,j,m}$ is a binary number. Eqns. (11) and (12) are for the capacity constraints of the non-renewable and renewable resources, respectively.

Once the updated $u_{i,j,m}$ is obtained, it is decoded as:

$$y_{i,j} = \begin{cases} m & \text{when } u_{i,j,m} = 1 \end{cases}, j = 1, 2, \ldots, D+2. \tag{13}$$

Based on the new $\vec{y}_i$, the $r_{j,k}$ and $v_{j,k} \forall k, j = 1, 2, ..., D+2$ are calculated. Then, the infeasible $\vec{x}_i$ is rectified based on the modified forward and backward SGSs. They are discussed in the following subsection.

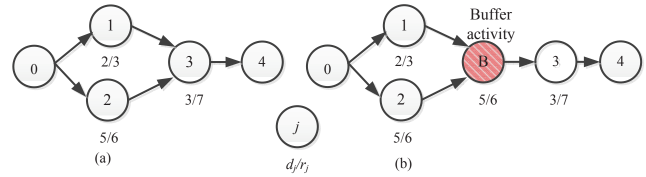

Fig. 3. An example to insert buffer activity for handling irregular resource

C. Handling Irregular Resources

As mentioned, the resources for a project can be irregular for many reasons, such as maintenance, being unavailable, or are used for another project. To handle this irregular resource, in this subsection, we discuss two new techniques, such as a BAI and ARS.

1) BAI: In this technique, we modify the original network and insert some buffer activities for the unavailability of the resources. For example, Fig. 3(a) shows a project with three non-dummy activities, and their required resource and duration are shown above their nodes. The project has a single resource (r_1), with a maximum capacity of 10 units. Let us assume, some units of r_1 will be used by a sister project during the implementation of the original project. Therefore, the capacity of r_1 will be irregular. To deal with this, we insert a buffer activity into the original network, as shown in Fig. 3(b). The buffer activity consumes some units of resource and duration those to be used by the sister project. All the activities including the buffer one are scheduled based on the forward and backward-SGS [2]. Therefore, scheduling the modified network gives an allocation of some resources for the sister project.

However, this technique is useful for a small-scale project and becomes complex for large problems with multiple resources that change over time. In addition, it is not a flexible approach as the time periods for the unavailability of the resource depends on the buffer activity being scheduled.

2) ARS: To overcome the above-motioned drawbacks in the BAI technique, we propose an alternative ARS approach that schedules the project based on the variable resource capacity constraint. In it, the capacity of each resource over the project horizon is revised based on their unavailability. Then, the activities (without a buffer) of the project are scheduled based on the modified forward and backward-SGSs.

In the modified forward SGS, the activities are scheduled based on their earliest starting times subject to satisfy their irregular resources and precedence constraints. In addition, at each time period, the duration of each activity is minimized by satisfying the available resource where the mode of the activity is optimized.

The modified backward SGS is used to further improve the quality of a schedule. In it, the activities are first sorted based on their finish times. Then, the activities are scheduled based on their latest finish time subject to satisfy the irregular resources constraint, while determining an appropriate mode to minimize the duration of each activity. Details of the forward and backward-SGS can be found in [2].x

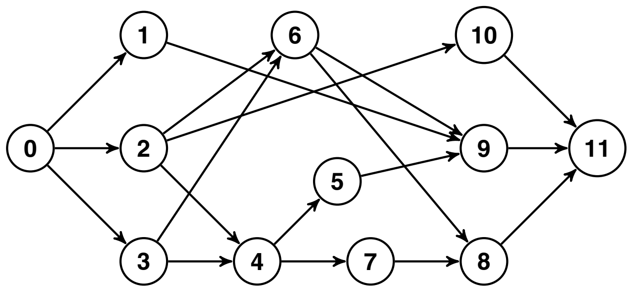

Fig. 4. Original network diagram of the test problem

D. moEAs

We use two moEAs, namely moGA and moDE to generate new solutions for both $\vec{x}_i$ and $\vec{y}_i$, $i = 1, 2, \ldots, N_P$. Both algorithms use multiple search operators; that is, moGA uses two crossover operators: (i) two-point and (ii) uniform, with a left-shift mutation operator considered, while moDE uses '*current-to-rand/bin with archive*' (DE_1) and '*current-to-rand/bin without archive*' (DE_2) [16].

At the start of the optimization process, all the search operators have the same preference to generate offspring, but this may change in the subsequent generations based on their ability to generate good quality solutions. Due to page limitation, the details are not described here but can be found in [14].

IV. Experimental Results

In this section, the performance of the proposed H-MEA is evaluated by solving a set of MM-RCPSPs under irregular renewable resources. We consider $j10$ benchmark set from PSPLIB [15], and a standard test problem from [17]. The $j10$ benchmark has 60 problems, with each problem has up to 10 instances. There are total 537 problems in $j10$ benchmark set. To compare the results, all the test problems are solved using three variants of the H-MEA. Their results are compared with each other, and also to the available results in recent literature.

The population size and the maximum number of schedules considered are 10 and 5000, respectively. Each test problem is run 30 times and their median results are reported. The algorithm is implemented in Matlab2018a on a Windows computer with a 3.4 GHz core i7 processor and 16 GB RAM.

A. Illustrated Problem: Discussion

In this subsection, we solve a simple MM-RCPSP with 10 non-dummy activities where each activity has three different modes, under an irregular renewable resource. The maximum availability of the resource is 10 units over the project horizon. The project network of the test problem is shown in Fig. 4, and its details can be found in [17].

First, the test problem is solved using H-MEA assuming that the renewable resources (10 units) are fixed and available over the project life. The Gantt chart of the schedule obtained is shown in Fig. 5, in which the project completion time is 27 days.

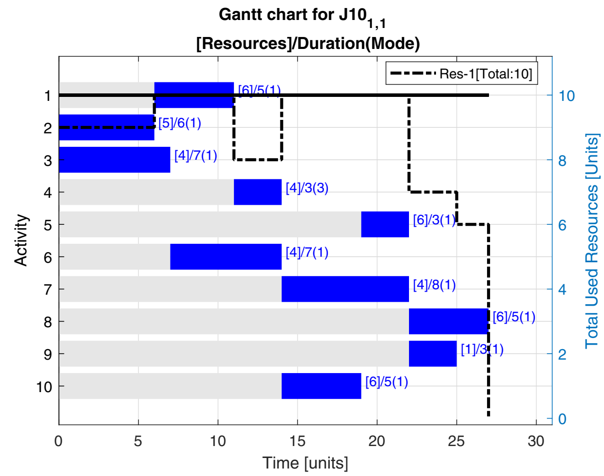

Fig. 5. Gantt chart for the test problem under regular resources

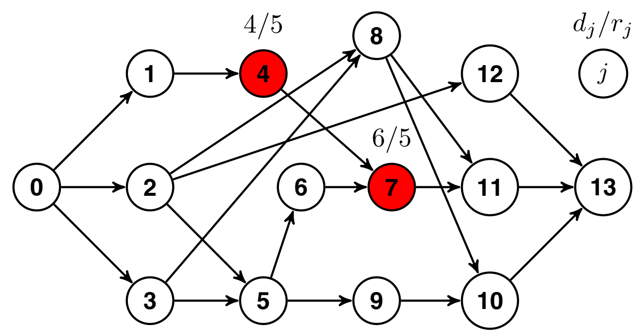

Fig. 6. Modified network diagram of the test problem

Then, we solve the test problem under the assumption that the resources are irregular due to periodic maintenance. Now, we solve the test problem using H-MEA with the irregular resources handled based on the BAI technique, as discussed in subsection III-C1. In it, we assume two maintenance events are required during the project implementation, i.e., one after finishing activity '1', another after finishing activity '6'. Fig. 6 shows the modified network where the two buffer activities in red are inserted due to the two maintenance events. The number of maintenance units and their duration are shown in above of each node.

After solving the modified test problem, the corresponding Gantt chart of the final solution shown in Fig. 7. It is seen that the make-span is increased from 27 to 32 days, which is obvious as two additional activities are incorporated in the network for the two periodic maintenance events. However, this approach has some drawbacks, such as flexibility and computational complexity. For example, the project manager cannot choose an arbitrary time period for the maintenance as it depends on the schedule of the buffered activities. In addition, for a higher-dimensional project, this approach may become complex.

To overcome the drawbacks mentioned above, we use the ARS approach to handle such irregular resources, as discussed

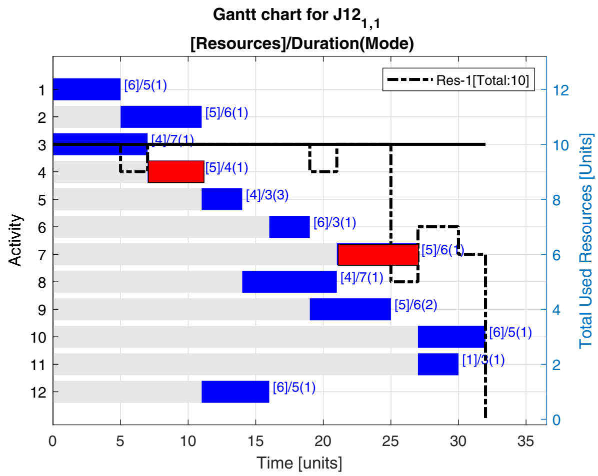

Fig. 7. Gantt chart for the modified test problem

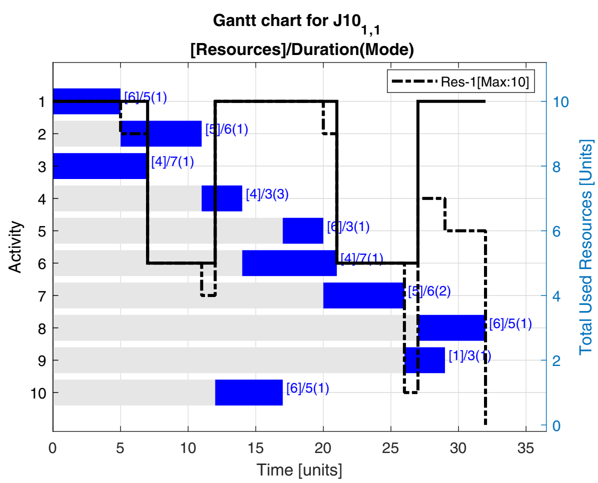

Fig. 8. Gantt chart for the test problem under irregular resources

in subsection III-C2. In this approach, we do not insert any additional activities into the network, but we consider irregular availability of the resources during the project horizon. To clarify, we assume the two maintenance events of the resource to be started at 7 and 21 days, respectively. Note that, these maintenance periods are also found in the BAI approach as shown in Fig. 7. With the new resource availability, the original network in Fig. 4 is solved using the H-MEA, and the obtained Gantt chart is shown in Fig. 8. It is seen that the make-span is 32, as found in the BAI technique. This indicates that the proposed ARS approach to handle irregular resources is appropriate, while it is relatively flexible and simple to formulate.

Let us consider a case where the project manager wants to start the two basic resources maintenance at day 4 and 14, respectively. This means no extra nodes, which represent maintenance, will be added to the network. Considering the new time periods for the resources maintenance, H-MEA solves the test problem and obtains the Gantt chart as shown

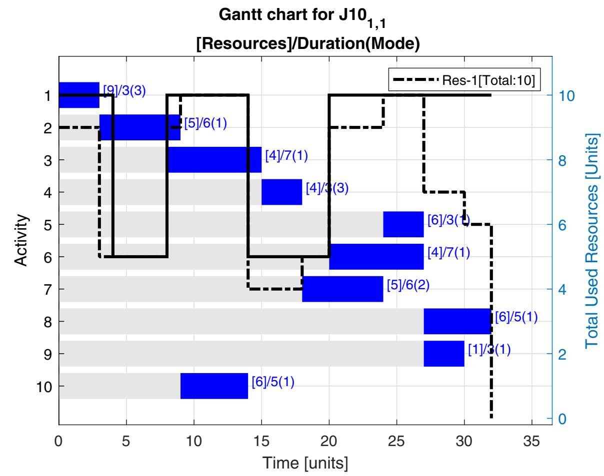

Fig. 9. Gantt chart for the test problem when two resource maintenance start at 4 and 14 days, respectively

in Fig. 9. It is seen that the make-span is still 32 even after increasing the second maintenance period from 5 to 6 days. Moreover, if we consider the best time periods for the basic maintenance, the make-span may be further reduced. Nevertheless, selecting the time periods arbitrarily is only possible in the ARS approach, while it may not be possible in the BAI technique, as discussed above.

B. Verification of H-MEA

In this subsection, we verify our proposed H-MEA algorithm by solving the standard benchmark of $j10$ under both regular and irregular resources. In addition, all the test problems are solved using the two variants of H-MEA, as:

- var1: a hybrid-GA
- var2: a hybrid-DE
- var3: the proposed H-MEA

For comparison purpose, the test problems are solved under the following two cases:

- case-1: the standard test problems where the resources are always available during the project horizon, and
- case-2: the modified test problems, where some of the resources are unavailable for a certain time period.

In case-2, we consider 5 units resources are unavailable from 5 to 8, and 15 to 20 days. As discussed in subsection IV-A, the irregular RCPSPs are handled based on the ARS approach.

We compare the average response value of each algorithm, that is calculated as [13]:

$$\Delta = \frac{1}{N}\sum_{n=1}^{N}\frac{f_n - OPT_n}{OPT_n} \qquad (14)$$

where f_n and OPT_n are the obtained make-span and optimal value of the n^{th} problem, respectively, N is the total problem number of problems, as $N = 537$ for $j10$.

Table I shows a comparison of the Δ found from the three variants of the H-MEA and those found in different recent

TABLE I
ARV FOR $j10$ BENCHMARK SET UNDER REGULAR AND IRREGULAR RESOURCES

Algorithm	Δ	
	case-1	case-2
WLZO [18]	0.28	NA
CLPE [19]	1.06	NA
CHA [20]	0.32	NA
ZLZH [21]	0.00	NA
TCGLS [22]	0.33	NA
RSS [23]	0.18	NA
EFEA [24]	0.14	NA
JRR [25]	0.28	NA
LZA [26]	0.09	NA
SEEDA [27]	0.09	NA
LCEDA [28]	0.12	NA
SLC [29]	0.05	NA
LHGA [30]	0.06	NA
MAN [31]	0.05	NA
CWC [32]	0.01	NA
SAAGA [18]	0.03	NA
HGFA [13]	0.02	NA
H-GA	0.28	18.04
H-DE	0.46	25.24
Prposed H-MEA	**0.00**	**17.96**

algorithms for MM-RCPSPs. The details of the state-of-the-art algorithms can be found in [13].

As the state-of-the-art algorithms are available for standard test problems (i.e., case-1), we have compared our results with them. It is found that the proposed H-MEA obtains optimal solutions for all test problems and outperforms two other algorithms. It is also evident that the other two variants of the H-MEA do not obtain the best solution, with H-GA relatively performs better than that of H-DE.

For case-2, we compare the results among the three variants of the H-MEA, as, to our knowledge, considering irregular resources in MM-RCPSPs has not been tackled in the literature yet. From the results, it is found that H-MEA outperforms the other two variants.

V. CONCLUSION AND FUTURE WORKS

In this research, we reformulated the MM-RCPSP under irregular resource changes over the project horizon. The changes of some resources due to maintenance and are used by another project. Considering such irregular resources, the MM-RCPSP has become complex and is a non-smooth optimization problem. To solve it, we developed a H-MEA that considers two moEAs, namely moGA and moDE, and two-stage heuristic approaches. The first stage heuristic rectifies any infeasible modes of the activities, while the second-stage obtains their quality schedule by satisfying the irregular resources and precedence constraints.

To deal with such irregular resources, two approaches, namely BAI and ARS techniques, were proposed. Both were tested by solving a set of standard test problems and their modified version to incorporate irregular availability of renewable resources. A comprehensive analysis was carried out by considering three variants of the proposed algorithm, in which it was revealed that the H-MEA with ARS, has some advantages in terms of solution quality and flexibility.

In the future, we will reformulate the problems by considering multiple projects that share some common resources. Considering a disruption in the shared resources would be another interesting area of research.

REFERENCES

[1] M. F. Zaman, S. M. Elsayed, T. Ray, and R. A. Sarker, "Scenario-based solution approach for uncertain resource constrained scheduling problems," *2018 IEEE World Congress on Computational Intelligence (IEEE WCCI)*, 2018.

[2] F. Zaman, R. Sarker, and D. Essam, "A new hybrid approach for the multimode resource-constrained project scheduling problems," in *The 48th International Conference on Computers and Industrial Engineering (CIE 48), The University of Auckland*, 2018.

[3] J. Cheng, J. Fowler, K. Kempf, and S. Mason, "Multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting," *Computers & Operations Research*, vol. 53, pp. 275 – 287, 2015.

[4] X. long Zheng and L. Wang, "A multi-agent optimization algorithm for resource constrained project scheduling problem," *Expert Systems with Applications*, vol. 42, no. 15, pp. 6039 – 6049, 2015.

[5] Z. Liu, L. Xiao, and J. Tian, "An activity-list-based nested partitions algorithm for resource-constrained project scheduling," *International Journal of Production Research*, vol. 54, no. 16, pp. 4744–4758, 2016.

[6] R. K. Chakrabortty, R. A. Sarker, and D. L. Essam, "Resource constrained project scheduling with uncertain activity durations," *Computers & Industrial Engineering*, vol. 112, no. Supplement C, pp. 537 – 550, 2017.

[7] F. Zaman, S. M. Elsayed, T. Ray, and R. A. Sarkerr, "Evolutionary algorithms for finding nash equilibria in electricity markets," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 536–549, Aug 2018.

[8] J. Alcaraz and C. Maroto, "A robust genetic algorithm for resource allocation in project scheduling," *Annals of Operations Research*, vol. 102, no. 1-4, pp. 83–109, 2001.

[9] M.-Y. Cheng, D.-H. Tran, and Y.-W. Wu, "Using a fuzzy clustering chaotic-based differential evolution with serial method to solve resource-constrained project scheduling problems," *Automation in Construction*, vol. 37, pp. 88–97, 2014.

[10] H. Zhang, X. Li, H. Li, and F. Huang, "Particle swarm optimization-based schemes for resource-constrained project scheduling," *Automation in Construction*, vol. 14, no. 3, pp. 393–404, 2005.

[11] V. V. Peteghem and M. Vanhoucke, "A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 201, no. 2, pp. 409–418, 2010.

[12] J. Coelho and M. Vanhoucke, "Multi-mode resource-constrained project scheduling using rcpsp and sat solvers," *European Journal of Operational Research*, vol. 213, no. 1, pp. 73–82, 2011.

[13] M. H. Sebt, M. R. Afshar, and Y. Alipouri, "Hybridization of genetic algorithm and fully informed particle swarm for solving the multi-mode resource-constrained project scheduling problem," *Engineering Optimization*, vol. 49, no. 3, pp. 513–530, 2017.

[14] S. Elsayed, R. Sarker, T. Ray, and C. C. Coello, "Consolidated optimization algorithm for resource-constrained project scheduling problems," *Information Sciences*, vol. 418-419, no. Supplement C, pp. 346 – 362, 2017.

[15] R. Kolisch and A. Sprecher, "Psplib - a project scheduling problem library," *European Journal of Operational Research*, vol. 96, no. 1, pp. 205–216, 1997.

[16] F. Zaman, S. M. Elsayed, R. Sarker, D. Essam, and C. A. C. Coello, "Multi-method based algorithm for multi-objective problems under uncertainty," *Information Sciences*, vol. 481, pp. 81 – 109, 2019.

[17] X. Qi, J. F. Bard, and G. Yu, "Disruption management for machine scheduling: The case of spt schedules," *International Journal of Production Economics*, vol. 103, no. 1, pp. 166 – 184, 2006.

[18] L. Wang, J. Liu, and M. Zhou, "An organizational cooperative coevolutionary algorithm for multimode resource-constrained project scheduling problems," *Lecture Notes in Computer Science*, vol. 8886, pp. 680–690, 2014.

[19] A. H. L. Chen, Y. C. Liang, and J. D. Padilla, "An entropy-based upper bound methodology for robust predictive multi-mode rcpsp schedules," *Entropy*, vol. 16, no. 9, pp. 5032–5067, 2014.

[20] C. W. Chiang and Y. Q. Huang, "Multi-mode resource-constrained project scheduling by ant colony optimization with a dynamic tournament strategy," *Proceedings - 3rd International Conference on Innovations in Bio-Inspired Computing and Applications, IBICA 2012*, pp. 110–115, 2012.

[21] L. Zhang, Y. Luo, and Y. Zhang, "Hybrid particle swarm and differential evolution algorithm for solving multimode resource-constrained project scheduling problem," *Journal of Control Science and Engineering*, vol. 2015, 2015.

[22] P. Jędrzejowicz and E. Ratajczak-Ropel, "Reinforcement learning strategy for solving the mrcpsp by a team of agents," *Smart Innovation, Systems and Technologies*, vol. 39, pp. 537–548, 2015.

[23] L. Wang and C. Fang, "An effective shuffled frog-leaping algorithm for multi-mode resource-constrained project scheduling problem," *Information Sciences*, vol. 181, no. 20, pp. 4804–4822, 2011.

[24] H. Li and H. Zhang, "Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints," *Automation in Construction*, vol. 35, pp. 431–438, 2013.

[25] O. S. Soliman and E. A. R. Elgendi, "A hybrid estimation of distribution algorithm with random walk local search for multi-mode resource-constrained project scheduling problems," *International Journal of Computer Trends and Technology*, vol. 8, pp. 57–64, 2014.

[26] C. Fang and L. Wang, "An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem," *Computers and Operations Research*, vol. 39, no. 5, pp. 890–901, 2012.

[27] J. Cui and L. Yu, "An efficient discrete particle swarm optimization for solving multi-mode resource-constrained project scheduling problem," *IEEE International Conference on Industrial Engineering and Engineering Management*, vol. 2015-January, pp. 858–862, 2014.

[28] H. Shen and X. Li, "Cooperative discrete particle swarms for multi-mode resource-constrained projects," *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2013*, pp. 31–36, 2013.

[29] O. Mirzaei and M. R. Akbarzadeh-T, "A novel learning algorithm based on a multi-agent structure for solving multi-mode resource-constrained project scheduling problem," *Lecture Notes in Electrical Engineering*, vol. 203 LNEE, pp. 231–242, 2012.

[30] R. M. Chen and C. M. Wang, "Controlling search using an s decreasing constriction factor for solving multi-mode scheduling problems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7906 LNAI, pp. 545–555, 2013.

[31] V. Van Peteghem and M. Vanhoucke, "An artificial immune system for the multi-mode resource-constrained project scheduling problem," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5482 LNCS, pp. 85–96, 2009.

[32] M. H. Sebt, M. R. Afshar, and Y. Alipouri, "An efficient genetic algorithm for solving the multi-mode resource-constrained project scheduling problem based on random key representation," *International Journal of Supply and Operations Management*, vol. 2, no. 3, pp. 905–924, 2015.