Pro-Reactive Approach for Project Scheduling under Unpredictable Disruptions

Forhad Zaman*[1], Saber Elsayed[1], Ruhul Sarker[1], Daryl Essam[1], and Carlos A. Coello Coello[2]

[1]School of Engineering and Information Technology, University of New South Wales, Australia
Email: `{f.zaman, s.elsayed, r.sarker, d.essam}@adfa.edu.au`

[2]*CINVESTAV-IPN, Departamento de Computación*, Mexico
Email: `ccoello@cs.cinvestav.mx`

Abstract*—Existing solution approaches for handling disruptions in project scheduling use either proactive or reactive methods. However, both techniques suffer from some drawbacks which affect the performance of the optimization process in obtaining good quality schedules. Therefore, in this paper, we develop an auto-configured multi-operator evolutionary approach, with a novel pro-reactive scheme for handling disruptions in multi-mode resource constrained project scheduling problems (MM-RCPSPs). In this paper, our primary objective is to minimize the makespan of a project. However, we also have secondary objectives such as maximizing the free resources (FRs) and minimizing the deviation of activity finishing time. As the existence of FR may lead to a sub-optimal solution, we propose a new operator for the evolutionary approach and two new heuristics to enhance the algorithm's performance. The proposed methodology is tested and analyzed by solving a set of benchmark problems, with its results showing its superiority with respect to state-of-the-art algorithms in terms of quality of the solutions obtained.

***Index Terms*—Scheduling, resource-constrained project scheduling problems, multi-mode, disruption, evolutionary algorithms**

I. INTRODUCTION

The objective of a standard resource-constrained project scheduling problem (RCPSP) is to find the best sequence of activities, by satisfying all the resource limitations and precedence constraints while minimizing project completion time. In such problems, single-mode resources are considered and are recognized as single-mode RCPSPs (SM-RCPSPs). Multi-mode RCPSPs (MM-RCPSPs) are an extension of SM-RCPSPs in which each activity has a number of non-preemptive execution modes, each of which may be different in terms of resource requirements and duration. Thus in addition to all the specifications of the SM-RCPSPs, an MM-RCPSP aims to find the best schedule of activities and their best execution modes so that completion time is minimized [1]. However, some of these resources may be unavailable at the time of execution due to an unexpected breakdown, which is known as a resource disruption. Resource disruptions are a critical issue in real-world projects [2]. MM-RCPSP is a well-known NP hard problem [3], [4]. One of the difficulties with the existing approaches for MM-RCPSPs is that they do not perform consistently over a wide range of problems. When considering MM-RCPSPs with unknown disruptions, they are much harder to solve and there is a lack of effective approaches for this type of problem. As of the literature, the approaches are either of proactive or reactive type and both of them have drawbacks [5], [6]. The assumptions made in those approaches usually resulted in sub-optimal solutions. We believe that an appropriate design and linking of these two approaches will allow us to reduce the effect of those assumptions. The main motivation of this research is that MM-RCPSPs with disruptions is a difficult practical problem, and there is a research gap on the development of its solution approach.

In this paper, MM-RCPSPs with unknown disruptions are considered. The aim is to propose an algorithm for solving MM-RCPSPs and then extend that algorithm to deal with MM-RCPSPs with disruptions. Evolutionary Algorithms (EAs) are a popular choice for solving RCPSPs for their various advantages [7]. However it is well-known that no single EA ensures consistent performance over a wide range of problems. This means that if one algorithm performs well for a set of problems, it may perform badly for some other problems. To mitigate this issue, multi-method and/or multi-operator algorithms are considered under a single algorithm framework in which each individual algorithm has the opportunity to be evolved, with its own set of parameters and operators, using its own sub-population and then share information with other sub-populations to achieve a common goal. The choice of the algorithms and operators are done based on their complementary properties. So a multi-method based evolutionary framework that works by configuring two multi-operator evolutionary algorithms (mo-EAs), namely multi-operator GA (mo-GA) and multi-operator DE (mo-DE), is developed in this paper. Such a framework has shown good performance in solving different optimization problems, including SM-RCPSPs [7] and MM-RCPSPs [8]. The proposed framework is automatically configured during the solution process, in which more emphasis is placed on the well-performing search operators and/or algorithm. To enhance the performance of the framework, two heuristics are also integrated, where the first one is based on a linear programming approach and the second one is based on modified forward-backward serial generation schemes (SGSs) [9].

We develop a new 'pro-reactive' mechanism to deal with

---

*Corresponding author

disruptions in MM-RCPSPs. In the first step of this mechanism, we propose a new proactive approach to generate an initial schedule by keeping reasonable free resources (FR) over the entire project life cycle. Such FRs can be used as a buffer for any potential disruption at a later stage. As maintaining FR may lead to a sub-optimal schedule, we propose a new selection operator that chooses the best schedule with both minimum makespan and maximum FR, which is different from existing approaches. This initial schedule is used for project execution. If any disruption occurs during the project execution, the initial schedule is rescheduled for the remaining part (i.e., the activities not executed yet) of the project, and a new reactive approach is used where the preserved FRs are utilized as much as possible. This is the way the proactive and reactive schedules are linked in our 'pro-reactive' approach.

Technically, in the proactive approach, we consider MM-RCPSPs as a constrained optimization problem with a primary objective of minimizing makespan and a secondary objective of maximizing FR. In the reactive approach, three objectives are considered where the primary objective is to minimize the makespan, the second objective is to maximize FR and the third objective is to minimize the deviation of the finish time of the remaining activities. Note that the proactive approach does not need the third objective and the reactive approach considers the secondary objective to keep a possible buffer by assuming another disruption may occur in the future. The optimization problem either for the proactive or for the reactive approach has been solved, for optimizing the primary objective, using the evolutionary framework discussed earlier. If there are any ties, the other objectives are used to break them.

The performance of the proposed algorithm is evaluated on well-known MM-RCPSP benchmark problems, with and without disruptions. To analyze the effect of a disruption on an initial schedule, a set of parametric tests with different types of disruptions has been conducted. In comparison with the recent literature, the proposed algorithm shows its superiority in terms of minimizing makespan, even after a series of disruptions.

The remainder of this paper is organized as follows: the existing literature and proposed mathematical formulations are presented in Sections II and III respectively. The proposed hybrid ensemble algorithm is provided in Section IV. The computational experimental results are reported in Section V. A discussion and analysis of results are provided in Section VI. Finally, our conclusions and some possible paths for future work are described in Section VII.

II. OVERVIEW OF EXISTING RESEARCH

Over the last few decades, many solution approaches have been proposed for solving RCPSPs. These approaches can be categorized as exact methods, heuristics, meta-heuristics and hybrid approaches. Exact methods were the main driver of research in the past and include: mixed integer linear programming (MILB), neighborhood search, and tree-based branch and bound (BB). These methods mainly dealt with SM-RCPSPs [2]. However, they are computationally expensive for large-scale SM-RCPSPs and even for small-scale MM-RCPSPs [2], [10]. Heuristics have been successfully applied to solve small to medium size MM-RCPSPs [11], [12]. For the third category, several meta-heuristic algorithms, such as genetic algorithms (GAs) [13], particle swarm optimization (PSO) [14] and differential evolution (DE) [15], have shown good performance in solving SM-RCPSPs and MM-RCPSPs [16]. Hybrid algorithms which combine two or more approaches, such as local search methods with a metaheuristic algorithm [17], [18], and multiple evolutionary algorithms [19], [7], [20], [21], [22], have also shown encouraging results in solving different complex project scheduling problems [23], [24], [25].

In most SM-RCPSPs and MM-RCPSPs solution approaches, it is assumed that resource requirements and activity durations are known and deterministic. However it is a well-known fact that project activities are subject to considerable uncertainties due to various practical issues. In this context, a number of studies have been reported and they may be categorized as either proactive or reactive or both [5], [6], [4]. The aim of the proactive approach is to generate a baseline schedule or robust scheduling, by considering the uncertainty involved, before execution of a project. It is assumed that this scheduling is the only schedule available for project execution. In the reactive approach, given a baseline schedule, if any unexpected delay occurs during the execution phase, a rescheduling or repairing of the schedule is conducted. In this case, it is assumed that no proactive strategy is considered in generating the baseline schedule. When considering both the proactive and reactive approaches (pro-reactive), the project is executed based on the proactive schedule and if any disturbances occur during execution, the reactive approach is applied to revise the schedule.

In the literature, proactive approaches have usually been proposed to deal with uncertain activity duration in SM-RPSPs [26], [27] and single-mode multi-project scheduling problems [28], [29]. These are considered as stochastic problems with different objectives and have been solved with a variety of approaches. For example, metaheuristic approaches have been applied for cost-based flexibility [30]. Also a chance-constrained programming approach with a new robustness measure such as expected solution stability has been adopted for these problems [26]. However, Ma et al. [31] considered splitting activities for maximizing robustness where the project deadline is known. A genetic algorithm was designed to solve the underlying optimization problem. Note that not all activities can be split practically, and it requires a trade-off between the benefits of activity splitting and the increased setup times of those splitting components. Capa and Ulusoy [29] considered a multi-project scheduling problem with uncertainty in the amount of resources' used and not in activity duration as is usually considered by most studies. A tabu search algorithm [32] was used to generate a proactive schedule under resources' uncertainty. In it, a surrogate approach inserts time buffers between activities to overcome any changes in their uncertain duration. The choice of robustness or flexibility as an objective would increase a project's makespan, which may not be useful if a lower number of disturbances (or no disturbance at all) occur during the project's execution.

Similar to the proactive approach, most reactive approaches have been developed for SM-RCPSPs. They may consider that the project deadline is either known [33], or to be determined [34]. Among the reactive approaches, hybrid branch and cut [5], tabu search [6], BB algorithms [4] and variable neighborhood search [33] are all well-known approaches. The minimization of the deviation of activity finish times from the baseline schedule is a popular objective in pro-reactive approaches [34].

Compared to SM-RCPSPs, studies in proactive and reactive scheduling in MM-RCPSPs are scarce [35], [36]. In these studies, the reactive scheduling basically deals with resource disruptions during a project's execution. Re-scheduling is usually considered as a single objective optimization problem which is solved using exact methods and heuristics [4], [37]. Elloumi et al. [38] developed a multi-objective optimization model for MM-RCPSPs with the objectives of minimizing makespan and minimizing a disruption measure, such as the finish time deviation and any changes in resources. They solved the developed model using an evolutionary algorithm and a multi-objective heuristic and obtained the solutions within a reasonable computational time. Although the use of the proactive approach to generate the baseline schedule and then the reactive approach to revise a schedule if any disruption occurs during project execution, would provide the highest possible benefits for MM-RCPSPs, we could not find such sort of approach in the literature.

Although, multi-objective optimization has been applied to MM-RCPSPs to generate a baseline schedule with different objectives, the concept of proactive scheduling is not considered in those studies [39]. A very limited number of studies have considered multiple objectives during the optimization process for reactive schedules in MM-RCPSPs [38]. However, no studies were found that considered multiple objectives for both proactive and reactive scheduling for MM-RCPSPs.

III. PROBLEM DESCRIPTION AND FORMULATION

In this section, we discuss the problem formulation for both the proactive and reactive approaches in the context of MM-RCPSPs. We consider a single project with a set of non-dummy activities and each of these activities requires some resources for a certain duration. The project has some predefined renewable resources which can be reused in each time period and their availability is constant throughout the project's duration. Also the project has some non-renewable resources which cannot be reused, such as money. In fact, each activity has several alternative execution modes that may include both renewable and non-renewable resources. From the different possible modes available, an activity can be executed in only one mode.

As shown in Fig. 1, a sample project (MM-RCPSP) has three non-dummy activities (1, 2 and 3, represented as nodes with circles), with 1 and 3 having single modes and 2 having three different ones, and two dummy ones (0 and 4, represented as nodes) which are referred to as the *start* and *finish* times of the project. The resource requirements and duration of each activity are shown under each circle. It can be seen that when activity 2 considers modes 1 and 3, the project's duration is 8 and 6, respectively. Therefore, the objective of MM-RCPSP is to find the best schedule of activities, with their best execution modes, while satisfying its precedence and resource constraints.

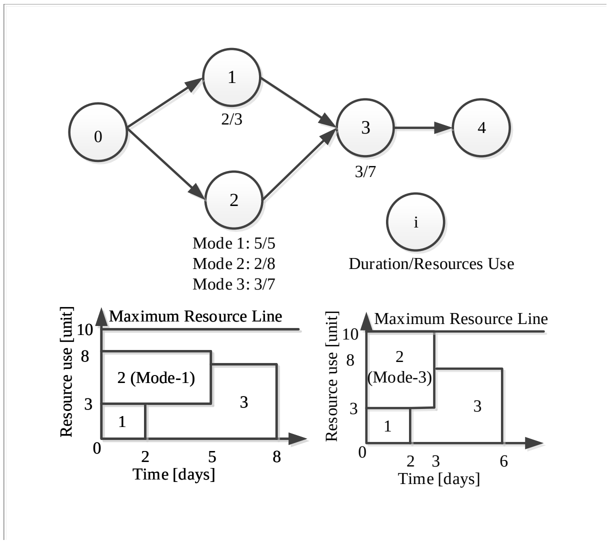

Fig. 1. Example of MM-RCPSP with its Gantt chart

A. Proactive Model

To develop the proposed pro-reactive approach, we formulate the optimization model for proactive scheduling in this sub-section. In this scheduling, we consider two objectives, the first is the primary objective with the aim of minimizing makespan and the aim of the secondary objective is to maximize the free resources (FR) in the whole project. We assume that a disruption might occur in the renewable resources and they would be unavailable for a certain period of time during the project duration. By maintaining FR as much as possible, it can be used to minimize the effect on project completion of a disruption when it occurs. For the purpose of modelling, its acronyms are defined and listed in Table I. The mathematical model is presented below.

$$\min : f_1 = FT_{D+2} \tag{1}$$

$$\max : f_2 = \sum_{t=0}^{f1} w_t \min\{FR_{1,t}, \ldots, FR_{K,t}\} \tag{2}$$

Where,

$$FR_{k,t} = \left(R_k - \sum_{j \in A_t} r_{j,m_j,k} \right), \forall k, t \tag{3}$$

Subject to:

$$FT_1 = 0 \tag{4}$$

$$FT_{i,m_i} \leq FT_{j,m_j} - d_{j,m_j}, \forall i \in P_j, \forall m_j \in M_j \tag{5}$$

Table I
LIST OF ACRONYMS

MM-RCPPSP Model:	
D	Number of non-dummy activities in a project.
j	An activity in a project, i.e., $j = 1, 2, \dots, D + 2$
A	Set of non-dummy activities in a project, i.e., $j = 2, 3, \dots, D + 1$.
A_t	Set of activities running in the t^{th} time period, i.e., $A_t \in A$.
P_j	Set of immediate predecessor activities for activity j.
M_j	Number of modes of the j^{th} activity.
m_j	Active mode of the j^{th} activity.
FT_{j,m_j}	Finish time of the j^{th} activity in the m_j^{th} mode.
ST_{j,m_j}	Start time of the j^{th} activity in the m_j^{th} mode.
K, N	Number of renewable and non-renewable resources in a project, respectively.
R_k, V_n	Numbers of available k^{th} renewable and n^{th} nonrenewable resources, respectively.
$r_{j,m_j,k}$, $v_{j,m_j,n}$	Numbers of k^{th} renewable and n^{th} nonrenewable resources required by the j^{th} activity in the m_j^{th} mode.
d_{j,m_j}	Duration of the j^{th} activity in the m_j^{th} mode.
f_1	Primary objective for project duration.
f_2	Secondary objective for FR over the project horizon.
Recovery Model:	
$\hat{f}_1$	Project duration after a disruption.
w_t	Weight value of the t^{th} time period for maximizing FR.
t_d	Start time of a disruption.
t_r	Recovery time.
$\hat{A}$	Set of activities which need to be rescheduled after a disruption.
$\hat{FT}_j$	Finish time of the j^{th} activity after disruption
Δ	Sum of deviations of finish times of affected activities, from their finish times in nominal schedule.
$\hat{R}_{k,t}$	Maximum number of the k^{th} renewable resource in the t^{th} time period after a disruption, i.e., $\hat{R}_{k,t} \leq R_k$
$R_{k,t}^{dis}$	Unit of the k^{th} resource breakdown at the t^{th} time.
Algorithm:	
P, G	Population size and maximum number of generations, respectively.
i	An individual in a population, i.e., $i \in N_P$.
g	Current generation number, i.e., $g \in N_G$.
$\vec{x}_i$	i^{th} individual of activities' sequences.
$\vec{y}_i$	i^{th} individual of modes of activities.

Hat (^) on a variable indicates its values after disruption.

$$\sum_{j \in A_t} r_{j,m_j,k} \leq R_k, \forall k \in K, \forall m_j \in M_j \tag{6}$$

$$\sum_{j=2}^{D} v_{j,m_j,n} \leq V_n, \forall m_j \in M_j, \forall n \in N \tag{7}$$

Eq. (1) represents the primary objective, that is to minimize a project's duration, where FT_{D+2} is the finish time of the *finish* dummy activity. Eq. (2) represents the secondary objective that is used to maximize the sum of the minimum number of free renewable resources (FR) available after they are used. In it, FR is calculated over a project's horizon, starting from zero hours, and ending at f_1, which is the project duration as per Eq. (1). The weight values (w_t) in Eq. (2) are determined based on the risk zones over a project's duration, which are associated with the most common time periods during which some resources can be unavailable due to maintenance, shared with another project and holidays, such as reduced workforce over Christmas. Alternatively, the potential risk zones are estimated based on historical data on similar projects and expert opinion. Eq. (3) shows the equation to determine the FR of the k^{th} unit at the t^{th} time period.

The constraint in Eq. (4) indicates that the project must be started at zero hour, where FT_1 is the finish time of the *start* dummy activity. Eq. (5) ensures the temporal relationships among activities, i.e., that no activity starts until its predecessors have finished, Eq. (6) ensures that the renewable resources used by the activities at any time must not be greater than their maximum limits and Eq. (7) means that the total number of non-renewable resources used by all the non-dummy activities, must not be greater than their capacities.

B. Reactive Model

A project is executed based on proactive scheduling. When the project encounters some unexpected disruptions, the schedule needs to be revised for recovery. In the literature, researchers have suggested two types of models for the recovery stage: (i) preempt-repeat and (ii) preempt-resume [4]. In the former, all the affected activities start from the beginning with their already completed portions discarded. In other words, in this approach, the activities which are affected due to any disruption are restarted from the beginning, and so any partially completed portions are scraped. On the other hand, in preempt-resume, their completed portions are considered and only the remaining ones are rescheduled. Although both approaches have their own pros and cons depending on a project's structure, for simplicity, we adopt the former.

In our recovery model, the primary objective is to minimize a project's duration, while satisfying the resource availability and precedence constraints of the affected activities. Also two other objectives are considered to maximize FR after a disruption and minimize deviations of the activities' finish times from their nominal ones. The third objective is important in the recovery stage, as it aims to ensure that the start and finish times of all of the activities obtained from the proactive schedule, are not significantly different when a disruption is encountered. This optimization model is defined as:

$$\min : \hat{f}_1 = \hat{FT}_{D+2} \tag{8}$$

$$\max : \hat{f}_2 = \sum_{t=t_d}^{f1} w_t \min \{FR_{1,t}, \dots, FR_{K,t}\} \tag{9}$$

$$\min : \Delta = \sum_{j=1}^{D+2} \left| FT_j - \hat{FT}_j \right| \tag{10}$$

Subject to:

$$\hat{FT}_{j,m_j} \geq t_d, \forall j \in \hat{A} \tag{11}$$

$$\hat{FT}_{i,m_i} \leq \hat{FT}_{j,m_j} - d_{j,m_j}, \forall i \in P_j, \forall m_j \in M_j \tag{12}$$

$$\sum_{j \in A_t} r_{j,m_j,k} \leq \hat{R}_{k,t},\ \forall k \in K,\ \forall m_j \in M_j \tag{13}$$

$$\sum_{j=2}^{D} v_{j,m_j,n} \leq V_n,\ \forall m_j \in M_j,\ \forall n \in N \tag{14}$$

Eq. (8) minimizes a project's duration after a disruption. Eq. (9) maximizes FR for another possible disruption, similarly to Eq. (2). Eq. (10) minimizes the differences between the activities' finish times, before and after a disruption. The constraint in Eq. (11) ensures that any activity, $j \in \hat{A}$ cannot start before the start time of a disruption (t_d), where $\hat{A}$ represents the set of activities that need to be rescheduled after the disruption. The constraint in Eq. (12) defines the precedence constraints, Eq. (14) the non-renewable resources and Eq. (13) the renewable resource capacity constraints, which reduce after a disruption and then resume their maximum capacity after the recovery period.

IV. PROPOSED SOLUTION APPROACH

SM-RCPSP is a well-known NP hard problem. However, MM-RCPSP is much harder to solve as compared to SM-RCPSP [4]. When proactive or reactive strategies are included with MM-RCPSP, it is not easy to design an effective and efficient solution approach. Note that existing approaches cannot even guarantee consistent performance for standard MM-RCPSPs. Considering this complexity of solving MM-RCPSPs, we propose a hybrid ensemble approach for solving them. The main reason for such a choice is that an appropriately designed ensemble approach utilizes the strength of multiple single operator-based algorithms, which ensures effective and efficient problem solving [7], [8]. To enhance the performance of the ensemble approach, we have also hybridized it with different local search procedures. The proposed framework and its components are discussed below.

A. Framework

In this research, we propose a pro-reactive approach, based on a hybrid ensemble algorithm (H-EA), for solving MM-RCPSPs with disruptions. The framework consists of two-stages: (i) proactive and (ii) reactive. Both stages use a H-EA that considers two mo-EAs (mo-DE and mo-GA), which evolve solutions towards the optimal sequence of activities. They also use two-stage heuristics which ensure that these solutions are always feasible, in terms of the precedence and resource constraints.

The framework starts with a new proactive approach in which H-EA begins with an initial population, of size N_P, where the decision variables (also called activities) are represented as discrete vectors, $\vec{x}_i$, and their corresponding modes $\vec{y}_i, i = 1, 2, ..., N_P$, are shown in subsection IV-B. Then for any infeasible solution, heuristics are applied to convert them to a feasible one. First a linear programming-based heuristic obtains a feasible $\vec{y}_i$. Subsequently the modified forward-backward SGS is used to convert infeasible $\vec{x}_i$ to feasible $\vec{x}_i\ i \in N_P$ (more details are given in subsections IV-C and IV-D).

Once feasible individuals are obtained their f_1 and f_2 are calculated and then the solutions are sorted according to a priority-based selection operator, as discussed in subsection IV-G, with the offspring of both $\vec{x}_i$ and $\vec{y}_i$ generated using an ensemble algorithm, as discussed in subsection IV-E. If any new $\vec{x}_i$ or $\vec{y}_i$ are infeasible it is converted to a feasible one, as previously stated. The evolutionary process continues until the predefined maximum number of schedules, N_G, is reached.

If any disruption occurs a reactive approach takes place. Firstly information about the disruption, such as its start and recovery times (t_d and t_r, respectively) and the reduced renewable resources ($\hat{R_{k,t}}$), is collected. Then the activities affected and their finish times are recorded from the initial scheduling. Based on the successors' matrix, as discussed in subsection IV-F1, some of the affected activities are allowed to continue with their current implementation, while others are rescheduled. H-EA is again used to reschedule the affected activities with half of the initial population taken from the final population of the initial solutions, as discussed in subsection IV-F. The pseudo-code of the proposed framework for MM-RCPSPs with disruptions is given in Algorithm 1 and its details are discussed in the following subsections.

Algorithm 1 Pseudocode of our proposed pro-reactive approach

```
Require: $G$ and $P$.
Procedure proactive
Initial population: generate $\vec{z}_i = \{\vec{x}_i, \vec{y}_i\}, \forall i = 1, 2, ..., P$, as shown in subsection IV-B .
for $g = 1; g \leq G; g++$ do
    for $i = 1; i \leq P; i++$ do
        Obtain a feasible $\vec{y}_i$ using a linear-programming-based-heuristic, as shown in subsection IV-C.
        Calculate $r_{j,m_j,k}, v_{j,m_j,k}, \forall k$ and $d_{j,m_j}, \forall j$ based on $\vec{y}_i$.
        Obtain a quality $\vec{x}_i$ after applying the MM-forward and backward-SGS, as discussed in subsection IV-D.
        Calculate $f_{1,i}$ and $f_{2,i}$ for $\vec{x}_i$, as shown in Eqs. (1) and (2), respectively.
    end for
    Select the best $P$ individuals of both $\vec{x}_i$ and $\vec{y}_i$ based on the new selection operator, as shown in subsection IV-G.
    Generate new offspring of both $\vec{x}_i$ and $\vec{y}_i$, $\forall i = 1, 2, ..., P$ using H-EA, as discussed in subsection IV-E.
end for
Start implementing the best $\vec{x}$ with minimum $f_1$.
Procedure reactive
if a disruption occurs then
    Pass the best $\vec{x}$ and the corresponding $\vec{y}$, $t_d$ and $t_r$ of the disruption to Algorithm 4.
end if
```

Line 2 in Algorithm 1 shows an initial population of the proposed H-EA and lines 3 and 4 are *for-loops* executing the algorithm up to G and P, respectively. In line 5 the linear-programming approach is used to obtain feasible modes of the activities, by minimizing their durations, while satisfying the non-renewable resources. Line 6 calculates the required resources and duration of each activity, based on their new active modes, line 7 uses a modified forward and backward SGS to improve $\vec{x}$ while satisfying the precedence and resource constraints, and line 8 evaluates the objective values of the multi-objective MM-RCPSP. Line 10 uses a new selection operator to select the best set of $\vec{x}$ and $\vec{y}$, and line 11 generates a new set of $\vec{x}$ and $\vec{y}$ using the ensemble algorithm. This process continues with line 13 selecting the best $\vec{x}$ at the end of G generations for implementation.

Line 15 monitors the project as to whether it is subject to a disruption. Once a disruption occurs line 16 starts Algorithm 4 to reschedule the affected activities.

B. Initial Population and Representation

For H-EA, we consider discrete decision variables which represent $\vec{x}_i$ and $\vec{y}_i$, and that are randomly generated as:

$$\vec{z}_i = \{\vec{x}_i, \vec{y}_i\}, \forall i = 1, 2, \ldots, N_P \tag{15}$$

$$\vec{x}_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,D+2}\} \tag{16}$$

$$\vec{y}_i = \{m_{i,1}, m_{i,2}, \ldots, m_{i,D+2}\} \tag{17}$$

$$x_{i,j} = \begin{cases} 1 & \text{if } j = 1 \\ D+2 & \text{if } j = D+2 \\ \mathbb{N} \cap \{2, 3, \ldots, D+1\} & \text{otherwise} \end{cases} \tag{18}$$

$$m_{i,j} = \mathbb{N} \cap [1, M_j], \forall j \tag{19}$$

Eq. (15) represents the i^{th} individual in an initial population which is generated by a random permutation (indicated as $\mathbb{N}$) of all the non-dummy activities, as shown in Eq. (18) for $\vec{x}_i$, and a random permutation (indicated as $\mathbb{N}$) of their modes, as shown in Eq. (19) for $m_{i,j} \in \vec{y}_i$. The number of decision variables of an individual is $N_x = 2 \times (D+2)$.

C. Linear Programming Model

As a $\vec{y}_i$ may not be feasible in terms of the resources required by the activities of $\vec{x}_i$, a linear programming approach that helps not only to obtain feasible $\vec{y}_i$, but may also reduce a project's duration, is proposed. To achieve this, we first encode a binary vector $\vec{u}_i^0$ from $\vec{y}_i$ as:

$$u^0_{i,j,m_j} = \begin{cases} 1 & \text{when } y_{i,j} = m_j \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

The size of $\vec{u}_i$ is $D \sum_{j=2}^{D+1} M_j$ (see Fig. 2).

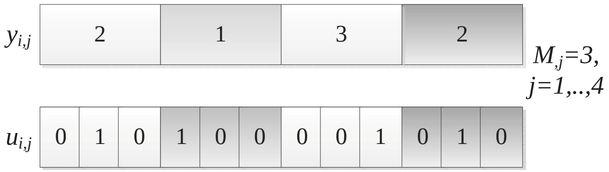

Fig. 2. Example of encoding approach of $\vec{u}_i$ from $\vec{y}_i$

Considering $\vec{u}_i^0$, we solve a linear programming model of MM-RCPSP using the simplex method, by relaxing the binary requirements, with the model presented below [9]:

$$\min : \sum_{j=2}^{D+1} \sum_{m_j=1}^{M_j} d_{j,m_j} u_{i,j,m_j}, \forall u \in \{0, 1\} \tag{21}$$

Subject to:

$$\sum_{m_j=1}^{M_j} u_{i,j,m_j} = 1, \forall j = 2, 3, \ldots, D+1 \tag{22}$$

$$0 \leq \Re\left(u_{i,j,m_j}\right) \leq 1, \forall m_j, \forall j = 2, 3, \ldots, D+1 \tag{23}$$

$$\sum_{j=2}^{D+1} \sum_{m_j=1}^{M_j} u_{i,j,m_j} \times v_{j,m_j,n} \leq V_n, \forall n \in N \tag{24}$$

$$\sum_{m_j=1}^{M_j} u_{i,j,m_j} \times r_{j,m_j,k} \leq R_k, \forall k \in K, \forall j = 2, 3, \ldots, D+1 \tag{25}$$

Eq. (21) represents the objective function of the linear programming model (to minimize the sum of the duration of all activities), Eqs. (22) and (23) ensure that an activity can operate in only one mode, Eqs. (24) and (25) are used to satisfy the capacity of the non-renewable and renewable resources, respectively, under their operating modes.

It is important to note that the above model always produces a unique $\vec{u}_i$ under any circumstance, but in H-EA we need N_P different solutions to avoid getting stuck in a local optimum. So the best solution (mode) generated by solving the above mentioned model is assigned to the elite $\vec{x}_i$, while Eq. (26) is used to obtain a different $\vec{u}_i$ for the non-elite $\vec{x}_i$ by:

$$\min : \sum_{j=2}^{D+1} \sum_{m_j=1}^{M_j} rand_{j,m_j} d_{j,m_j} u_{i,j,m_j}, \forall u \in \{0, 1\} \tag{26}$$

where $rand_{j,m_j}$ is a random number between 0 and 1, and Eq. (26) is a modified version of Eq. (21). The logic behind Eq. (26), is to use different values for each solution, so that a different $\vec{u}_i$ are obtained.

Once all $\vec{u}_i, i = 1, 2, \ldots, N_P$ are obtained, they are decoded to obtain $\vec{y}_i$,∀ i=1,2,...,N$_P$ as:

$$y_{i,j} = m_j \text{ when } u_{i,j,m_j} = 1, \forall j = 1, 2, \ldots, D+2 \tag{27}$$

D. MM-Forward-Backward-Serial Generation Scheme (SGS)

As a schedule, $\vec{x}_i, i \in N_P$ may not be feasible in terms of satisfying the precedence and resource constraints, forward-backward-SGS is commonly used [40] to obtain a feasible $\vec{x}_i$ from an infeasible one. However, traditional SGS does not incorporate variant modes of activities in it's scheduling. As a result, although the obtained $\vec{x}_i$ are feasible, its quality many not be the best [9].

In this research, we use a modified MM-forward-backward-SGS to obtain a feasible and high quality $\vec{x}_i, i \in N_P$ from an infeasible one. In forward-SGS, activities are scheduled based on their earliest start times, while backward-SGS schedules them based on their latest finish times, subject to resource and precedence constraints. To improve the quality of $\vec{x}_i$ we incorporate the activities' modes in the heuristic procedure, whereby each activity is scheduled with its required resources and duration optimized by determining an appropriate mode.

The pseudocode of the modified MM-forward-backward-SGS is shown in Algorithm 2. It begins with an infeasible $\vec{x}_i$ with each activity's duration and required resources determined, based on the feasible $\vec{y}_i$ obtained from the linear programming model. Line 2 in Algorithm 2 uses a *while-loop* to ensure that all activities are scheduled, while line 3 schedules each activity which satisfies the precedence constraints. Lines 6 to 14 are used to find the best mode for that activity, to minimize duration while satisfying the resource constraints. The combination of the resources and duration of activity is optimized by searching for an appropriate mode. If the activity satisfies the precedence constraints, but not the resource capacity ones, it is scheduled to the next possible time period, as shown in line 16, while if it cannot satisfy all the constraints, the next activity of $\vec{x}_i$ is considered and scheduled, as shown in line 20.

Line 23 uses MM-backward-SGS to further improve the schedule of $\vec{x}_i$, as described in Algorithm 3. It begins with the updated $\vec{x}_i$ and $\vec{y}_i$ obtained from line 22 in Algorithm 2, with the activities sorted in descending order based on their finish times and then all those in the sorted $\vec{x}_i$ are rescheduled. If an activity satisfies its precedence and resource constraints then it is scheduled as far right as possible (up to the project duration), with its used resources and duration minimized by searching for an appropriate mode. Then if even after changing its mode it still does not satisfy the resource constraints, it is scheduled to the next position left. Once all the activities of $\vec{x}_i$ are rescheduled, they are re-ordered based on Algorithm 2.

Algorithm 2 Pseudocode of MM-forward-backward SGS

```
Require: $\vec{x}_i, i \in P$, $R_k$, $V_n$ $r_{j,m_j,k}$, and $v_{j,m_j,n}$, $\forall k, n, m_j, j$.
 1: Set $j = 1$, $t = 0$ and $count = 1$.
 2: while $count \leq D + 2$ do
 3:     if precedence constraints of $x_{i,j}$ satisfied then
 4:         Set Scheduled($x_{i,j}$) = False.
 5:         do
 6:             Set ModeFound = False.
 7:             for $m_j = y_{i,j}; m_j \in [1 : M_j]$ do
 8:                 if Eqs. (5) to (7) satisfied then
 9:                     Set ModeFound = True, and
                        Scheduled($x_{i,j}$) = True.
10:                     Calculate $FT_{j,m_j} = FT_{h,m_h} + d_{j,m_h}$,
                        $\forall h \in P_j$, $\forall j$
11:                 end if
12:             end for
13:             if ModeFound = True then
14:                 Update $x_{i,j}$ and $y_{i,j}$ based on minimum
                    $FT_{j,m_j}$, and set $count = count + 1$.
15:             else
16:                 Set, $t = t + 1$
17:             end if
18:         while Scheduled($x_{i,j}$) = True
19:     else
20:         Set $j = j + 1$.
21:     end if
22: end while
23: Update $\vec{x}_i$ and $\vec{y}_i$ using the MM-backward-SGS, as shown
    in Algorithm 3.
24: Return $\vec{x}_i$ and $\vec{y}_i$.
```

Algorithm 3 Pseudocode of MM backward SGS

```
Require: Updated $\vec{x}_i, i \in P$, $R_k$, $V_n$ $r_{j,m_j,k}$, and $v_{j,m_j,n}$,
    $\forall k, n, m_j, j$ from Algorithm 2.
 1: Sort activities of $\vec{x}_i$ in descending order, based on their
    finish times.
 2: Set SuccessBackward = False.
 3: for $j = 1; j \leq D + 2; j = j + 1$ do
 4:     Set $t_{start}$: minimum start times of $x_{i,j}$'s
        successors.
 5:     if $t_{start}$ > start time of $(x_{i,j}) + d_{i,j}$ then
 6:         Push $x_{i,j}$ as far as possible by tuning its mode while
            satisfying resource and precedence constraints, as
            shown in Steps 7 to 12 in Algorithm 2.
 7:         Set SuccessBackward = True.
 8:     end if
 9: end for
10: if SuccessBackward = True then
11:     Repeat steps 1 to 22 in Algorithm 2 for $\vec{x}_i$ to re-sort
        its activities.
12: end if
13: Return updated $\vec{x}_i$ and $\vec{y}_i$.
```

E. Ensemble Algorithm

To generate offspring of both $\vec{x}_i$ and $\vec{y}_i$, $i \in N_P$, the proposed H-EA algorithm is based on two mo-EAs, (i) mo-GA and (ii) mo-DE, that are executed in a single framework. It sets its initial probabilities of generating a new offspring to 1, as $prob_1 = prob_2 = 1$, so that both algorithms are used to achieve a quick convergence. However after a certain number of generations, called a cycle (CS), these probabilities are updated based on each algorithm's success rate (SR) in generating a better offspring than its parent, as:

$$prob_a = \max\left(0.1, \min\left(0.9, \frac{\sum_{g=1}^{CS} SR_{a,g}}{\sum_{a=1}^{2}\sum_{g=1}^{CS} SR_{a,g}}\right)\right) \quad (28)$$

where $SR_{a,g}$ is the success rate of the the a^{th} algorithm in the g^{th} generation, calculated as:

$$SR_{a,g} = \begin{cases} \frac{|F_{a,g}^{parent} - F_{a,g}^{offspring}|}{F_{a,g}^{parent}} & F_{a,g}^{parent} \neq F_{a,g}^{offspring} \\ 1 & \text{otherwise} \end{cases} \quad (29)$$

where $F_{a,g}^{parent}$ and $F_{a,g}^{parent}$ are the fitness values of the best individual of the parent and offspring, respectively, in the g^{th} generation obtained by the a^{th} algorithm.

To generate an offspring, both algorithms use multiple search operators. mo-GA uses two crossover operators: (i) two-point and (ii) uniform and a left-shift mutation operator, and mo-DE uses two search operators, DE1 and DE2, which are based on '*current-to-rand/bin with archive*' and '*current-to-rand/bin without archive*', respectively. The search operators in both algorithms evolve different numbers of individuals, based on their performance in previous generations, with the better-performing one evolving more individuals than the other. Their performance is calculated based on their success rates in generating better offspring than their parents, as shown in Eq. (29).

It is also worth mentioning that both of mo-DE's search operators deal with a real-valued $\vec{z}_i^{cont}$, in which an encoding approach is used to generate a real-valued individual, i.e., $\vec{z}_i^{cont}$ from a discrete one $\vec{z}_i, i \in N_P$. Then after generating the offspring of $\vec{z}_i^{cont}$, a decoding approach is applied to convert its equivalent to an integer individual, i.e., $\vec{z}_i, i \in N_P$. All these search operators and the encoding and decoding approaches are described in [7].

F. Disruption and Recovery Stage

In this research, we consider a disruption in a MM-RCPSP which may occur during its implementation, and define it as a renewable resource breakdown for a certain time period, after which the initial schedule may need to be revised. For rescheduling we follow a successors' matrix to give priority to some of the activities, as discussed in the following subsections.

1) Rescheduling using Successors' matrix : At the time of rescheduling it is assumed that the start and recovery times of a disruption are known and based on its start and finish times, the activities which might be interrupted can also be known, while even after a disruption some activities can continue to be implemented under the reduced resources. As there is a question regarding which activities can run and which might be rescheduled, a reasonable answer is to consider an activity to continue, if it finishes most of its tasks before a disruption begins. However when the start times of some of the affected activities are the same, there is a conflict in terms of decision-making. Let us consider an example of a project's network. Its Gantt chart is shown in Fig. 3, in which the project suffers a disruption that starts at 5 days and recovers at 7. During the recovery period the availability of the resource is reduced from 10 units to 5. It can be seen in the Gantt chart that activities 2 and 3 are running after the disruption, so one of these can continue while the other needs to be interrupted. Let us consider the two possible solutions shown in Fig. 4, in which it can be seen in the lefthand figure that the revised duration of the project is 15 when activity 2 is allowed to run and 3 is stopped until the disruption is recovered. On the other hand in the Gantt chart at the right in Fig. 4, the project's duration is 16 when activity 3 runs and 2 is interrupted. This is because activity 2 has two successors, while 3 has only one and activities 4 and 5 cannot start until activity 2 is completed. Therefore in this case the activity with the largest number of successors is considered in the rescheduling phase, for which we use a successors' matrix that represents the number of successors of each activity.

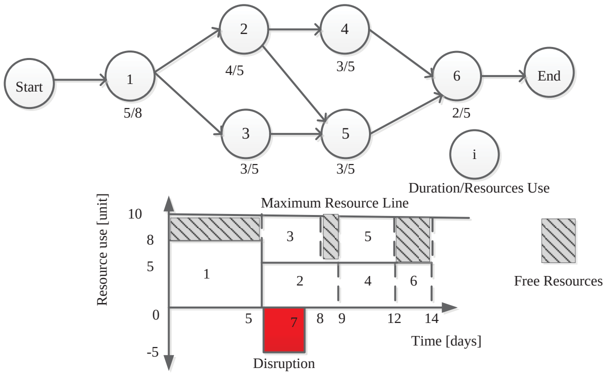

Fig. 3. Example of a single-mode RCPSP under disruption

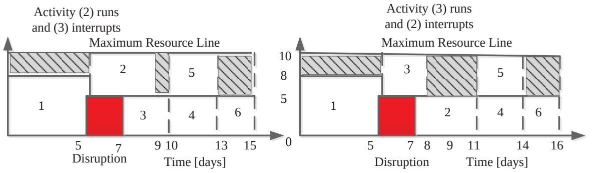

Fig. 4. Two possible solutions after disruption

2) Rescheduling of Affected Activities: After a disruption the affected activities are revised, based on the reduced resources. Also some of the affected activities are allowed to run based on their successors' matrix, as discussed in subsection IV-F1. The activities which need to be rescheduled are passed to Algorithm 4, in which half of the individuals of the initial population are randomly generated, as shown in subsection IV-B, while the remaining ones are taken from the current population that was generated using Algorithm 1. In subsequent generations the offspring of both the affected $\vec{xx}_i$ and $\vec{yy}_i$ are generated using the ensemble algorithm, as shown in subsection IV-E. Each infeasible $\vec{xx}_i$ and $\vec{yy}_i$ is rectified based on the MM-forward and MM-backward SGS techniques and the linear programming approach, respectively. Both the $\vec{xx}_i$ and $\vec{yy}_i$ are sorted, based on $\hat{f}_1$, Δ, and $\hat{f}_2$, as shown in Eqs. (8), (10), and (9), respectively, a process which is continued until the stopping criteria are met. The pseudo-code for rescheduling the affected activities is shown in Algorithm 4.

Algorithm 4 Pseudocode of H-EA for rescheduling activities after a disruption

```
Require: Initial schedule of $\vec{x}$ with their modes $\vec{y}$, start time, finish time, $t_d$ and $t_r$.
Determine: $\vec{xx}_i = x_{i,j}$ when the finish time of $x_{i,j} > t_d, j = 1, 2, ..., D+2, i \in P$, and their corresponding $\vec{yy}$.
Set and update, $\vec{zz} = \{\vec{xx}, \vec{yy}\}$ based on the successors' matrix.
Calculate, $\hat{R}_{k,t} = R_k - R^{dis}_{k,t}$.
Initial population: consider $P/2$ individuals of $\vec{zz}$ from the $G^{th}$ generation of Algorithm 1, with the remaining ones randomly generated, as shown in subsection IV-B.
for $g = 1; g \leq G; g++$ do
    for $i = 1; i \leq P; i++$ do
        Considering new $\hat{R}_{k,t}$, apply linear-programming, MM-forward and MM-backward SGS to obtain a high quality $\vec{zz}_i$, as shown in Steps 5 to 7 in Algorithm 1.
        Determine $\hat{f}_{1,i}$, $\Delta_i$ and $\hat{f}_{2,i}$ as shown in Eqs. (8), (10) and (9), respectively.
    end for
    Select the best $\vec{zz}$, based on Eq. (31).
    Generate new offspring of both $\vec{xx}$ and $\vec{yy}$ using the ensemble algorithm, as shown in subsection IV-E.
    if $\Delta = 0$ then
        Terminate the algorithm.
    end if
end for
```

G. Selection Operator

As mentioned earlier we formulate MM-RCPSP with the objectives of minimizing makespan and maximizing FR. In this case the individuals in a population are sorted based on these two objective values. In the traditional selection approaches similar priority is given to all objectives. In this research we propose an approach for sorting and selecting the best individuals in a population based on a lexicographical order, as discussed below.

RCPSP is a highly constrained discrete optimization problem in which it is highly likely that there will be some solutions with the same fitness value. Here we sort the individuals using the objectives' priorities. In the initial scheduling there are two objectives, the project duration (f_1) and FR (f_2), as shown in Eqs. (1) and (2), respectively. To obtain a high-quality schedule we first sort the individuals based on their f_1 values and then those with the same f_1, with the maximum f_2 preferred; for example an individual is selected from its parent ($\vec{z}_i$) and offspring ($\vec{Z}_i$) as:

$$\vec{z}_{i,g+1} = \begin{cases} \vec{z}_{i,g} & \text{if } f_{1,i} < F_{1,i} \\ \vec{z}_{i,g} & \text{if } f_{1,i} = F_{1,i} \text{ and } f_{2,i} > F_{2,i} \; i \in N_P \\ \vec{Z}_{i,g} & \text{otherwise} \end{cases} \tag{30}$$

where $f_{1,i}$ and $F_{1,i}$ are the project duration for the i^{th} parent and its offspring, respectively, and $f_{2,i}$ and $F_{2,i}$ are FR of the i^{th} parent and its offspring, respectively.

For the recovery stage the above selection operator is slightly modified, where the first priority is to minimize $\hat{f}_{1,i}$ (makespan after a disruption), and then Δ_i (deviations of the finish times after a disruption), and finally $\hat{f}_{2,i}$ (FR after a disruption) is used to maximize FR. Therefore Eq. (30) is updated as:

$$\vec{zz}_{i,g+1} = \begin{cases} \vec{zz}_{i,g} & \text{if } \hat{f}_{1,i} < \hat{F}_{1,i} \\ \vec{zz}_{i,g} & \text{if } \hat{f}_{1,i} = \hat{F}_{1,i} \text{ and } \Delta_{z,i} < \Delta_{Z,i} \\ \vec{zz}_{i,g} & \text{if } \hat{f}_{1,i} = \hat{F}_{1,i}, \Delta_{z,i} = \Delta_{Z,i} \text{ and } \hat{f}_{2,i} > \hat{F}_{2,i} \\ \vec{ZZ}_{i,g} & \text{otherwise} \end{cases} \tag{31}$$

where $\hat{f}_{1,i}$, $\hat{f}_{2,i}$ and Δ_i can be obtained from Eqs. (8), (9), and (10), respectively.

V. COMPUTATIONAL EXPERIMENTS

The performance of the proposed H-EA is evaluated by solving well-known MM-RCPSPs from well-known PSPLIB with their data including the network complexity (the average number of non-redundant arcs per node including the dummy activities), resource factor (the average portion the resources used) and resource strength (scaling parameter expressing resource availability) [41]. The benchmark sets up to $j30$, each of which has 60 instances containing 10 problems, are solved using H-EA for 5000 schedules. The results obtained are compared with respect to recently published algorithms in the literature. For a fair comparison with [4], we consider a disruption in selected problems and solve them using our proposed recovery approach. For comparison purposes, we solve these problems using the following three variants of H-EA:

- **var1**: solves problems without considering any FR;
- **var2**: solves problems considering FR over the project's horizon; and
- **var3**: solves the problems considering FR in the selected risk zones.

var1 is one of the traditional approaches where no FR is intentionally maintained over the time of scheduling, i.e., the second objective function in Eq. (2) is discarded from the optimization model during the pro-active solution approach. On the other hand, **var2** and **var3** are two alternative considerations of the second objective function in the optimization model. **var3** is our proposed approach that maximizes FR in the potential risk zones in terms of future resource disruptions.

Each variant is run 30 times with N_P set to 10 and N_G to 500, and their average results are used to compare the performance of the algorithm, which was implemented in Matlab 2018a on a desktop computer with a 3.4 GHZ Intel Core i7 processor and 16 GB in RAM.

A. MM-RCPSP with Disruption

In this subsection, we consider a disruption in an MM-RCPSP, with the initial schedules (those obtained by solving a problem without disruption) revised using H-EA and compare it with similar disruptions in some selected problems [4]. We consider a disruption in the renewable resources that starts at day 10 and recovers at 12, during which time 2 and 5 units of the first and second renewable resources, respectively, are lost as:

$$\hat{R}_{k,t} = \begin{cases} R_k - 2 & \text{if } k = 1 \text{ and } t = 10 \text{ to } 12 \\ R_k - 5 & \text{if } k = 2 \text{ and } t = 10 \text{ to } 12 \\ R_k & \text{otherwise} \end{cases} \tag{32}$$

Each problem is initially solved using the three variants (i.e., var1, var2 and var3) of the proposed H-EA to generate nominal schedules, with the primary objective of minimizing project duration. Also we consider a secondary objective of maximizing FR using var2 and var3, with the risk zones for var3 arbitrarily set as (4 to 6), (10 to 12) and (14 to 16) so that the disruption lies in a risk zone. For var3 in Eq. (2), we consider the weight as:

$$w_t = \begin{cases} 100 & \text{if } t = (4 \text{ to } 6), (10 \text{ to } 12), \text{ and } (14 \text{ to } 16) \\ 1 & \text{otherwise} \end{cases} \tag{33}$$

where $w_t = 100$ indicates that FR needs to be maintained with a high priority and $w_t = 1$ means that this is less important.

Table II shows the results for f_1, $\hat{f}_1$ and Δ obtained from the three variants and a BB algorithm in [4]. It can be seen that the optimal solutions to all the test problems are achieved by the proposed H-EA. After a disruption all the nominal solutions from the three variants are revised, with the activities before a disruption not considered. From the revised schedules it is clear that the $\hat{f}_1$ obtained by the BB algorithm significantly increased, while the proposed H-EA does not have this drawback, i.e., it obtains the same $\hat{f}_1$ as f_1 for most problems. The reason for this is its ability to efficiently select the best set of modes so that the schedule does not alter, even after a disruption. For some problems, e.g., $j10_{2,6}$, $j10_{48,1}$, $j20_{48,9}$, and $j30_{55,1}$, $\hat{f}_1$ is slightly increased when it is solved using var1, it reduces using var2 and is minimum using var3. This is because FR is maintained in both var2 and var3, while FR is maintained at the risk zones in var3, so that the initial schedule is not affected, even after a disruption.

Furthermore, in Table II it is clear that var3 of H-EA is the best for minimizing Δ. In fact when the initial solution is generated using it, a disruption does not have any impact on the solutions to some problems. Fig. 5 shows the average performance with respect to f_1, $(\hat{f}_1 - f_1)$ and Δ, in which it can be seen that the proposed H-EA with var3 is the best approach for obtaining both the initial and revised schedules.

B. Multiple MM-RCPSPs with Disruptions

In this subsection we discuss solving multiple test problems with disruptions. As the difficulty of a test problem depends largely on the network's complexity (NC), resource factor (RF) and resource strength (RS), based on these factors we categorize the problems as one of four types, i.e., T/1, T/2, T/3 and T/4 [4], with each of their instances solved using the three variants of the proposed H-EA.

Table II
SINGLE DISRUPTION IN SELECTED PROBLEMS

Problem	f_1		$\hat{f}_1 - f_1$				Δ		
	BB [4]	H-EA	BB [4]	var1	var2	var3	var1	var2	var3
$j10_{2,6}$	16	16	6	4	4	**3**	21	21	**18**
$j10_{20,1}$	12	12	6	3	0	0	6	0	0
$j10_{48,1}$	17	**16**	0	1	1	**0**	4	4	**0**
$j10_{55,1}$	19	**18**	0	0	0	0	0	0	0
$j10_{62,1}$	17	**15**	4	0	0	0	0	0	0
$j20_{9,8}$	18	18	16	3	3	3	18	15	15
$j20_{10,5}$	23	**22**	13	4	4	4	36	**34**	36
$j20_{27,1}$	17	17	13	0	0	0	0	0	0
$j20_{48,9}$	26	26	6	2	1	**0**	19	5	**2**
$j20_{64,10}$	22	22	17	0	0	0	2	2	2
$j30_{9,1}$	31	31	3	0	0	0	13	8	**6**
$j30_{10,5}$	33	33	3	0	0	0	0	0	0
$j30_{27,10}$	29	29	1	0	0	0	0	0	0
$j30_{55,1}$	29	29	3	1	0	0	10	2	2
$j30_{64,10}$	36	36	3	0	0	0	2	2	**1**

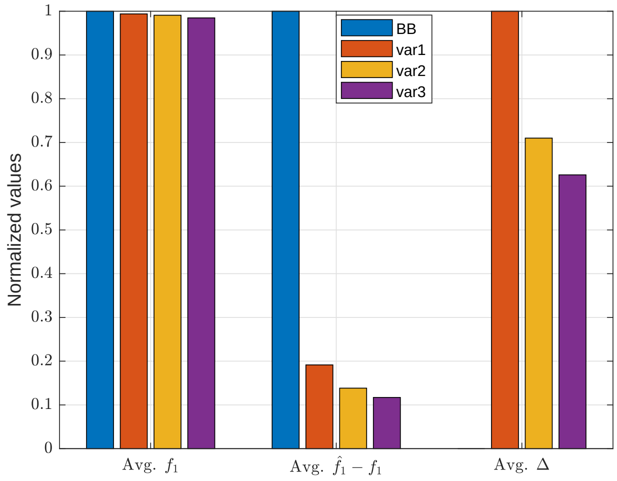

Fig. 5. Comparison of solution approaches for rescheduling activities after disruption

For each test problem we consider the sample disruption shown in Eq. (32) and the initial schedules are revised, based on the reduced resources. Table III shows the average results of f_1, $\hat{f}_1$ and Δ for each set of instances in the $j10$, $j20$ and $j30$ test problems. Although the initial f_1 obtained from all H-EA variants are the same, they increase differently after a disruption and it can be seen that var3 is the best in terms of the minimum $(\hat{f}_1 - f_1)$ and Δ for most problems.

C. MM-RCPSP with Series of Disruptions

In this subsection we evaluate the performance of H-EA for solving some selected test problems with a series of disruptions. We assume that when a disruption occurs, another one cannot occur until the resources have returned to their normal level. We consider the following disruptions:

- case 1 (two disruptions): $t_d = 10$ and 15, and $t_r = 2$;

Table III
RESULTS FOR GROUP-WISE TEST PROBLEMS WITH SINGLE DISRUPTION

Prob.	Type	f_1	$\hat{f}_1$			Δ		
		(avg.)	var1	var2	var3	var1	var2	var3
$j10$	T/1	19.36	21.84	21.04	21.04	11.44	11.21	9.97
	T/2	14.84	15.17	15.11	15.05	1.05	0.79	0.53
	T/3	19.82	22.87	22.01	21.94	13.14	12.91	12.58
	T/4	24.16	26.28	25.76	25.70	13.90	10.80	10.02
$j20$	T/1	23.72	25.51	25.84	25.40	18.39	26.43	19.82
	T/2	23.28	23.33	23.35	23.35	1.05	0.88	0.68
	T/3	28.26	29.70	29.54	29.28	22.56	20.44	16.62
	T/4	31.84	32.80	32.54	32.32	16.58	17.30	13.98
$j30$	T/1	30.20	31.00	30.90	30.85	22.60	19.60	19.15
	T/2	30.25	30.25	30.25	30.28	0.55	0.63	0.30
	T/3	35.62	36.52	36.34	36.18	28.54	25.38	18.02
	T/4	38.44	39.08	39.00	38.86	23.24	19.14	14.60

Table IV
VALUES OF $(\hat{f}_1 - f_1)$ WITH A SERIES OF DISRUPTIONS

	Case 1			Case 2			Case 3		
	var1	var2	var3	var1	var2	var3	var1	var2	var3
$j10$	2.40	1.40	**0.00**	4.20	4.00	**3.80**	2.60	0.00	0.00
$j20$	1.80	1.60	**1.40**	2.00	0.40	0.40	0.00	0.00	0.00
$j30$	0.20	0.40	**0.00**	0.60	0.40	**0.20**	1.20	1.40	**0.80**

- case 2 (three disruptions): $t_d = 8$, 13 and 18, and $t_r = 2$; and
- case 3 (five disruptions): $t_d = 10$, 15, 20, 25 and 30, and $t_r = 2$.

For simplicity each disruption indicates the breakdowns of 2 and 5 units of the first and second type of renewable resources, respectively. Each case is solved using the three above variants of the proposed H-EA, with their average deviations of makespan reported in Table IV and the best result for each test problem highlighted in **boldface**. It can be seen that var3 of H-EA is the best algorithm for rescheduling, as $\left(\hat{f}_1 - f_1\right)$ is minimum, even after a series of disruptions.

VI. DISCUSSION AND ANALYSIS

In this subsection we analyze the effect of different parameters and disruptions on the performance of H-EA. Due to page limitations further analysis on the secondary objective and successor's matrix are presented in supplementary material.

A. Impacts of Different Disruptions

In this subsection we analyze the impact of a schedule with different types of disruptions with differing values of (i) t_d and (ii) t_r. The former refers to the start time of the first disruption and the second to its recovery time. For simplicity we consider a single disruption in three random problems, $j10_{2,7}$, $j20_{9,5}$ and $j30_{18,6}$, which are solved using the proposed H-EA with var3. The initial makespans (f_1) of these problems without any disruption are 25, 26 and 25, respectively.

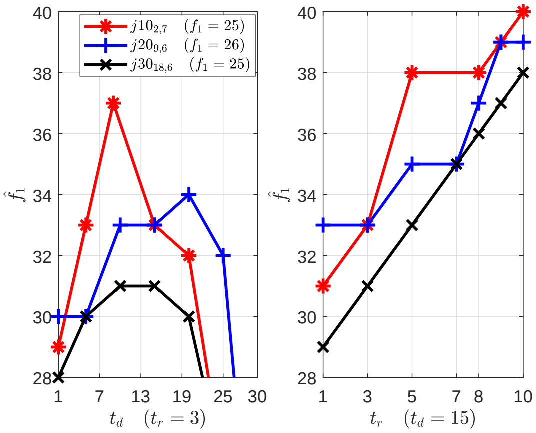

Fig. 6. Changes in makespans by changing start times and duration of disruptions

A single disruption with a complete loss of renewable resources is considered and the test problems solved for different values of t_d, with the value of t_r remaining fixed as 3. It can be seen in Fig. 6 (lefthand graph) that $\hat{f}_1$ approximately follows a semicircle when t_d increases, with the maximum $\hat{f}_1$ found when the disruption appears in the middle of the project's duration. This is realistic because if a project encounters a disruption in an early or late stage, it has adequate time to recover, or its duration is not significantly affected, as only a few activities need to be rescheduled, respectively. However, a disruption occurring in the middle of a project is more severe, because it affects many activities and there is insufficient time to recover within the initial makespan.

We then fixed the value of t_d at 15 and varied those of t_r from 1 to 10. Fig. 6 (righthand graph) shows that $\hat{f}_1$ almost monotonically increases with increasing values of t_r, because all the activities remain off during these periods.

B. Verification of the Proposed H-EA

In this subsection we verify H-EA by solving a set of standard test problems without considering any disruptions. The results are compared with those published in the recent literature, with full elaborations of the algorithms provided in [19]. We compare the algorithm's performance in terms of average deviation from the known optimal solutions, as:

$$\text{average deviation} = \frac{1}{N}\sum_{n=1}^{N}\frac{f_{1,n} - OPT_n}{OPT_n} \tag{34}$$

where OPT_n and $f_{1,n}$ are the optimal value and makespan obtained for the n^{th} problem, respectively, with N being the total number of problems in a particular benchmark set. Here, we consider $N = 537$ for $j10$, $N = 548$ for $j12$, $N = 552$ for $j14$, $N = 551$ for $j16$, $N = 553$ for $j18$, and $N = 555$ for $j20$. All the data and optimal values of these problems can be found in PSPLIB [41].

The results reported in Table V reveal that the proposed H-EA obtains 100% optimal solutions for both the $j10$ and $j12$ benchmark sets, and the best-quality ones for the $j14$, $j16$ and $j20$ problems. Although H-EA does not obtain the best solutions for the $j18$ problems, it achieves very competitive results. Nevertheless, the mean values of the average deviations (%) for all the benchmark sets is minimum for our algorithm.

Table V
ARV FOR THE WELL-KNOWN MM-RCPSP FOR 5000 SCHEDULES

Algorithms	$j10$	$j12$	$j14$	$j16$	$j18$	$j20$	**Mean.**	**Rank**
WLZO [42]	0.28	0.79	1.18	2.75	NR	NR	1.25	19.75
JSA [43]	1.16	1.73	2.60	4.07	5.52	6.74	3.64	20.50
CLPE [35]	1.06	0.41	3.43	3.36	2.12	5.78	2.69	19.08
CHA [44]	0.32	NR	NR	NR	NR	2.05	1.19	21.67
AGA [45]	0.24	0.73	1.00	1.12	1.43	1.91	1.07	17.50
ZLZH [46]	0.00	NR	NR	NR	NR	1.82	0.91	18.25
TCGLS [47]	0.33	0.52	0.93	1.08	1.32	1.69	0.98	17.00
RSS [48]	0.18	0.65	0.89	0.95	1.21	1.64	0.92	15.42
EFEA [49]	0.14	0.24	0.77	0.91	1.30	1.62	0.83	14.25
JRR [50]	0.28	0.41	0.54	0.75	0.92	1.55	0.74	13.50
LCSFLA [45]	0.10	0.21	0.46	0.58	0.94	1.40	0.62	11.50
LZA [51]	0.09	0.13	0.40	0.57	1.02	1.10	0.55	9.92
SEEDA [52]	0.09	0.12	0.36	0.42	0.85	1.09	0.49	8.42
LCEDA [53]	0.12	0.14	0.43	0.59	0.90	1.28	0.58	10.67
CYDP [54]	NR	NR	NR	NR	NR	0.97	0.97	20.33
SLC [55]	0.05	0.21	0.46	0.82	1.21	1.62	0.73	12.08
LHGA [56]	0.06	0.17	0.32	0.44	0.63	0.87	0.42	8.17
MAN [57]	0.05	0.09	NR	0.22	**0.18**	0.80	0.27	7.67
CWC [58]	0.01	NR	NR	NR	NR	0.71	0.36	16.42
VPVAIS [41]	0.02	0.07	0.20	0.39	0.52	0.70	0.32	4.67
VPVGA [59]	0.01	0.09	0.22	0.32	0.42	0.57	0.27	4.58
SAAGA [42]	0.03	0.07	0.21	0.32	0.40	0.56	0.27	4.17
HGFA [19]	0.02	0.08	0.13	0.28	0.29	0.41	0.20	3.00
Prposed H-EA	**0.00**	**0.00**	**0.11**	**0.08**	0.31	**0.41**	**0.15**	**1.50**

The overall performance of H-EA was statistically evaluated using a Friedman test in which the benchmark sets of $j10$ to $j20$ problems are considered as samples. The mean rank in Table V indicates that H-EA is the best algorithm for solving all the test problems.

VII. CONCLUSIONS AND FUTURE WORK

In the literature most studies were conducted using a single objective optimization with either a proactive or reactive approach. There were also some variations. In this paper we have discussed a few alternative models and analyzed their appropriateness for the problem environment considered in this research. Although makespan minimization is the prime objective, we have considered two other objectives with justification for their inclusion in our model. These objectives can be optimized either sequentially with priority settings for obtaining a single optimal solution or simultaneously generating multiple alternative solutions. As the primary objective is clearly defined it is appropriate to consider the sequential optimization of these objectives. Analyzing the drawbacks and complementary properties of proactive and reactive approaches, we have realized that their appropriate integration can do a better job. The proposed integration is a two-stage approach where the sequential optimization of the objectives would be more appropriate as the single solution obtained in the first stage can be easily used as input to the second stage for its refinement.

In this research we designed an integrated pro-reactive approach for MM-RCPSPs with disruption. To handle disruption we proposed a new technique that represented MM-RCPSP using a tiered set of objectives, with the objectives of minimizing makespan and maximizing FR. FR was effectively maintained over the project's horizon, which resulted in many initial schedules that were not significantly changed after a disruption.

To solve the optimization problem we proposed a new H-EA with two mo-EAs (mo-GA and mo-DE) and a two-stage heuristics approach. The mo-EAs were used to generate best-quality schedules of the activities and their modes, and the heuristics were used to ensure that the solutions were feasible. The first heuristic obtained feasible modes from any infeasible ones and the second rectified the schedule of activities by satisfying the resource capacity and precedence constraints.

The performance of the proposed H-EA was evaluated by solving well-known MM-RCPSPs from PSPLIB of the $j10$ to $j30$ benchmark sets, while its performance with disruptions was evaluated by comparing the results obtained by its three variants and a state-of-the-art algorithm, which showed that with maximizing FR in the risk zones, it was the best. Also the proposed algorithm was verified by solving the standard benchmark sets without disruption and the results revealing that H-EA was statistically better than many other existing ones. Furthermore a number of parametric tests were carried out to show the effectiveness of the algorithm under different types of disruptions.

Based on the analysis, if a project encounters a disruption at an early stage of project execution, it is possible to reschedule with minimum changes. A disruption occurring in the middle of a project is more severe, as there may not be sufficient time to recover without increasing the initial makespan. For a later stage disruption the makespan would usually be increased unless the disruption information is known well in advance. The consideration of maximizing free resources (FR) as a secondary objective is the key for improving the schedule under disruption. If FR is maximized for a later stage of the project duration, or over selected risk zones (when known from historical data), the benefit is maximum as compared to maximizing FR for all zones. So the prediction of risk zones based on historical data would be beneficial for such schedules.

As part of our future research work, we aim to develop new algorithms for solving RCPSPs involving complex practical issues, such as dynamic disruptions and uncertainty, and for other problems, such as multiple projects with common resource requirements, with or without additional complexities.

ACKNOWLEDGMENTS

This research has been funded by Australian Research Council Discovery Projects DP170102416 (for research) and DP190102637 (further work for revision of this paper). The last author gratefully acknowledges financial support from CONACyT grant no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*) and from a project from the 2018 SEP-Cinvestav Fund (application no. 4).

REFERENCES

[1] L. Y. Tseng and S. C. Chen, "Two-phase genetic local search algorithm for the multimode resource-constrained project scheduling problem," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 848–857, 2009.

[2] R. Chakrabortty, "Managing uncertainties and disruptions in resource constrained project scheduling," Ph.D. dissertation, The University of New South Wales, Canberra, Australia, 2017.

[3] H. Vahdani and A. Shams, "Multi-mode capital-constrained project payment scheduling model considering bonus-penalty structure," *International Journal of Management Science and Engineering Management*, vol. 15, no. 1, pp. 17–25, 2020.

[4] R. K. Chakrabortty, R. A. Sarker, and D. L. Essam, "Multi-mode resource constrained project scheduling under resource disruptions," *Computers & Chemical Engineering*, vol. 88, pp. 13 – 29, 2016.

[5] G. Zhu, J. F. Bard, and G. Yu, "A two-stage stochastic programming approach for project planning with uncertain activity durations," *Journal of Scheduling*, vol. 10, no. 3, pp. 167–180, Jun 2007.

[6] F. Deblaere, E. Demeulemeester, and W. Herroelen, "Reactive scheduling in the multi-mode RCPSP," *Computers and Operations Research*, vol. 38, no. 1, pp. 63–74, 2011.

[7] 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.

[8] F. Zaman, S. Elsayed, R. Sarker, D. Essam, and C. A. Coello Coello, "Evolutionary algorithm for project scheduling under irregular resource changes," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, June 2019, pp. 403–410.

[9] 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, pp. 1–12.

[10] 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.

[11] 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.

[12] 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.

[13] A. Alipour, A. A. Khodaiari, A. Jafari, and R. Tavakkoli-Moghaddam, "Uncertain production scheduling optimization in open-pit mines and its ellipsoidal robust counterpart," *International Journal of Management Science and Engineering Management*, vol. 13, no. 4, pp. 245–253, 2018.

[14] 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.

[15] 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.

[16] F. Zaman, S. Elsayed, R. Sarker, D. Essam, and C. A. Coello Coello, "An evolutionary approach for resource constrained project scheduling with uncertain changes," *Computers Operations Research*, vol. 125, p. 105104, 2021.

[17] 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.

[18] 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.

[19] 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.

[20] 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)*, pp. 1–8, 2018.

[21] F. Zaman, S. M. Elsayed, R. Saker, and D. Essam, "Resource constrained project scheduling with dynamic disruption recovery," *IEEE Access*, vol. 8, pp. 144 866–144 879, 2020.

[22] F. Zaman, S. Elsayed, R. Sarker, and D. Essam, "Hybrid evolutionary algorithm for large-scale project scheduling problems," *Computers Industrial Engineering*, vol. 146, p. 106567, 2020.

[23] X. Yu, W. Chen, T. Gu, H. Zhang, H. Yuan, S. Kwong, and J. Zhang, "Set-based discrete particle swarm optimization based on decomposition for permutation-based multiobjective combinatorial optimization problems," *IEEE Transactions on Cybernetics*, vol. 48, no. 7, pp. 2139–2153, 2018.

[24] Z. Zhang, W. Chen, H. Jin, and J. Zhang, "A preference biobjective evolutionary algorithm for the payment scheduling negotiation problem," *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[25] H. Zhao, Z. Zhan, Y. Lin, X. Chen, X. Luo, J. Zhang, S. Kwong, and J. Zhang, "Local binary pattern-based adaptive differential evolution for multimodal optimization problems," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3343–3357, 2020.

[26] P. Lamas and E. Demeulemeester, "A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations," *Journal of Scheduling*, vol. 19, pp. 409–428, 2016.

[27] J. Xiong, Y.-w. Chen, K.-w. Yang, Q.-s. Zhao, and L.-n. Xing, "A hybrid multiobjective genetic algorithm for robust resource-constrained project scheduling with stochastic durations," *Mathematical Problems in Engineering*, vol. 2012, p. 786923, 2012.

[28] X. Wang, Q. Chen, N. Mao, X. Chen, and Z. Li, "Proactive approach for stochastic rcmpsp based on multi-priority rule combinations," *International Journal of Production Research*, vol. 53, no. 4, pp. 1098–1110, 2015.

[29] C. Capa and G. Ulusoy, "Proactive project scheduling in an RD department a bi-objective genetic algorithm," *IEOM 2015 - 5th International Conference on Industrial Engineering and Operations Management, IEEE Proceeding*, vol. 15091335, pp. 1–6, 2015.

[30] M. Brcic, M. Katic, and N. Hlupic, "Planning horizons based proactive rescheduling for stochastic resource-constrained project scheduling problems," *European Journal of Operational Research*, vol. 273, no. 1, pp. 58–66, 2019.

[31] Z. Ma, Z. He, N. Wang, Z. Yang, and E. Demeulemeester, "A genetic algorithm for the proactive resource-constrained project scheduling problem with activity splitting," *IEEE Transactions on Engineering Management*, vol. 66, no. 3, pp. 459–474, 2019.

[32] O. Lambrechts, E. Demeulemeester, and W. Herroelen, "Time slack-based techniques for robust project scheduling subject to resource uncertainty," *Annals of Operations Research*, vol. 186, no. 1, pp. 443–464, Jun 2011.

[33] W. Zheng, Z. He, N. Wang, and T. Jia, "Proactive and reactive resource-constrained max-NPV project scheduling with random activity duration," *Journal of the Operational Research Society*, vol. 69, no. 1, pp. 115–126, 2018.

[34] O. Lambrechts, E. Demeulemeester, and W. Herroelen, "Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities," *Journal of Scheduling*, vol. 11, pp. 121–136, 2008.

[35] A. Chen, Y. C. Liang, and D. Padilla, "An entropy-based upper bound methodology for robust predictive multi-mode RCPSP schedules," *Entropy*, vol. 16, pp. 5032–5067, 2014.

[36] A. Chen, Y. C. Liang, and D. Padilla, "An experimental reactive scheduling framework for the multi-mode resource constrained project scheduling problem," *Lecture Notes in Engineering and Computer Science*, vol. 2228, pp. 853–858, 2017.

[37] F. Deblaere, E. Demeulemeester, and W. Herroelen, "Reactive scheduling in the multi-mode RCPSP," *Computers & Operations Research*, vol. 38, pp. 63 – 74, 2011.

[38] S. Elloumi, P. Fortemps, and T. Loukil, "Multi-objective algorithms to multi-mode resource-constrained projects under mode change disruption," *Computers and Industrial Engineering*, vol. 106, pp. 161–173, 2017.

[39] M. Salimi and A. A. Najafi, "Modeling and solution procedure for a preemptive multi-objective multi-mode project scheduling model in resource investment problems," *Journal of optimization in Industrial Engineering*, vol. 11, pp. 169–183, 2018.

[40] S. Elsayed, M. F. Zaman, and R. Sarker, "Automated differential evolution for solving dynamic economic dispatch problems," in *Intelligent and Evolutionary Systems*. Springer, 2016, pp. 357–369.

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

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

[43] J. Józefowska, M. Mika, R. Różycki, G. Waligóra, and J. Węglarz, "Simulated annealing for multi-mode resource-constrained project

scheduling," *Annals of Operations Research*, vol. 102, no. 1, pp. 137–155, Feb 2001.

[44] 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.

[45] J. Alcaraz, C. Maroto, and R. Ruiz, "Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms," *Journal of the Operational Research Society*, vol. 54, no. 6, pp. 614–626, 2003.

[46] 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, p. 923791, 2015.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] A. Lova, P. Tormos, M. Cervantes, and F. Barber, "An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes," *International Journal of Production Economics*, vol. 117, no. 2, pp. 302–316, 2009.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

Forhad Zaman received his Ph.D. degree from the University of New South Wales (UNSW), Canberra, ACT, Australia, in 2017. He is currently an Adjunct Lecturer with the School of Engineering and Information Technology, University of New South Wales, Canberra. His current research interests include power system operation and control, active bidding strategy in the electricity market, evolutionary algorithms, multiobjective optimization, handling real-life uncertainty, and game theory.

Saber Elsayed received the Ph.D.degree in Computer Science from the University of New South Wales Canberra (UNSW Canberra), Australia, in 2012. Currently, he is a Senior Lecturer with the School of Engineering and IT, UNSW Canberra. His research interests include evolutionary algorithms, constraint-handling techniques in evolutionary optimization, scheduling, big data and cyber security using computational intelligence (CI). Dr. Elsayed won several IEEE-CEC competitions. He is currently serving as the chair of the IEEE CI Society (ACT Chapter), and has held organizational roles in several conferences.

Ruhul Sarker received his Ph.D. in 1992 from Dalhousie University, Halifax, Canada. He is currently a Professor in the School of Engineering and IT, Canberra, Australia. His main research interests are Evolutionary Optimization, and Applied Operations Research. He is the lead author of the book *Optimization Modelling: A Practical Approach*. Prof. Sarker is a member of IEEE and INFORMS.

Daryl Essam received his B.Sc. degree in computer science from University of New England, Australia in 1990 and Ph.D. degree from University of New South Wales, Australia, in 2000. Since 1994, he has been with the Canberra campus, UNSW, where he is currently a Senior Lecturer. His research interests include genetic algorithms, with a focus on both evolutionary optimisation and large scale problems.

Carlos A. Coello Coello (M'98-SM'04-F'11) received a PhD in computer science from Tulane University, USA, in 1996. He is currently Professor with Distinction (CINVESTAV-3F Researcher) at the Computer Science Department of CINVESTAV-IPN, in Mexico City, Mexico. Dr. Coello has authored and co-authored over 450 technical papers and book chapters. He has also co-authored the book *Evolutionary Algorithms for Solving Multi-Objective Problems* (Second Edition, Springer, 2007). His publications currently report over 53,700 citations in Google Scholar (his h-index is 93). Currently, he is currently EIC of the IEEE Transactions on Evolutionary Computation and serves in the editorial board of several other international journals. He received the 2007 National Research Award from the Mexican Academy of Sciences in the area of Exact Sciences, the Medal to the Scientific Merit 2009, granted by Mexico City's congress and the National Medal of Science and Arts in the area of Physical, Mathematical and Natural Sciences. He is a Fellow of the IEEE, and a member of the ACM, Sigma Xi, and the Mexican Academy of Science. His major research interests are: evolutionary multi-objective optimization and constraint-handling techniques for evolutionary algorithms.