A NEW HYBRID APPROACH FOR THE MULTIMODE RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEMS

Forhad Zaman1*, Saber Elsayed, Ruhul Sarker and Daryl Essam
School of Engineering and Information Technology
The University New South Wales, Canberra, Australia
{f.zaman, s.elsayed, r.sarker, d.essam}@unsw.edu.au

ABSTRACT

A multi-mode resource constrained project scheduling problem (MM-RCPSP) aims to minimize the duration of a project by scheduling its activities and their operating modes, while satisfying their precedence and resources constraints. Although many solution approaches, including evolutionary algorithms (EAs), have been developed to solve this problem, the existing processes to determine feasible modes of the activities are computationally expensive. In addition, no-single algorithm in literature performs consistently better for wide-range of test problems. Therefore, this paper proposes a new hybrid solution approach, based on a multi-method EA and three heuristic approaches. The multi-method EA considers two EAs in which they are self-adaptively used during the search process, with more emphasis is placed on a better performing EA. The first heuristic is designed based on a linear-programming approach that finds the best set of feasible modes for the activities, while other two based on multimode forward and backward schedule generation schemes, those are used to improve the solution quality generated from the EAs. A set of benchmark problems are solved, with the solutions obtained compared with those of recently reported results in the literature. The results demonstrate that the proposed approach is superior to many algorithms compared in this paper.

Keywords: Multi-mode resource constrained project scheduling problem, Evolutionary Algorithm, Heuristics.

* Corresponding Author

1. INTRODUCTION

A project consists of a set of predefined activities, which each has a certain duration and budget for its required resources. However, the number of available resources during the project completion is limited. In addition, some activities cannot be started until their predecessors are completed. Therefore, the activities must be scheduled in such a way, so that the total duration of the project is minimum, while satisfying their predecessors and resource constraints. This scheduling is called a Resource Constrained Project Scheduling Problem (RCPSP) [1]. The applications of RCPSP are found in diverse areas, such as, planning, manufacturing, construction, and software implementation [2].

Recently, RCPSPs have been extended to multi-mode RCPSP (MM-RCPSP), in which the duration of each activity of the project, is considered as a function of its level and types of resources committed to it. MM-RCPSP is more complicated than the single-mode RCPSP because it determines not only the best schedule of the activities but their best set of modes [3]. These problems have been proven to be an NP-hard problem [4]

Over the last few decades, RCPSPs have been found to be one of the challenging optimization problems, and many solution approaches have been developed for solving such complex problems. They are broadly categorized into three types, namely exact procedures, heuristic and meta-heuristic [5]. Among them, although exact methods, such as integer programming, linear-programming, and branch and bound algorithms have produced some promising results, they were unsuccessful for large-scale RCPSP [2]. To overcome this drawback, some heuristic methods, such as priority rule-based methods, were developed. Although some heuristics were widely used because of their easy implementation and computational speed, they failed to solve a wide-range of RCPSPs [2]. Meaning, one heuristic may perform well for a particular RCPSP but may not show similar performance in other RCPSPs [3]. On the other hand, the meta-heuristic algorithms, such as genetic algorithm (GA) [6], differential evolution (DE) [7], and particle swarm optimization (PSO) [8] have shown outstanding performance in solving many RCPSPs.

However, none of the algorithms perform better for solving a wide-range of RCPSP, as some of them are better for a problem, but are inferior in others [2]. To overcome this, Elsayed, Sarker, Ray and Coello [2] proposed a consolidated optimization algorithm (COA), based on two efficient multi-operator based evolutionary algorithms (MOEAs): a MOGA and MODE, for solving a wide-range of single-mode RCPSPs. However, they did not solve MM-RCPSP.

During last few decades, various meta-heuristic algorithms have been developed for solving MM-RCPSP; for example, several variants of GA [9, 10] were developed, in which a local search was employed to improve the GA's performance. Peteghem and Vanhoucke [5] proposed another variant of GA for MM-RCPSP, in which they applied a heuristic based on the resource scarceness matrix that represented the used and available resources after each activity scheduled. Coelho and Vanhoucke [11] developed a two-stage solution process for MM-RCPSP, in which the first stage determined the feasible modes and second one scheduled the activities. Wang and Fang [12] a shuffled frog-leaping algorithm, in which they determined a set of feasible modes randomly and then the activities list was obtained using the frog-leaping algorithm. Sebt, Afshar and Alipouri [3] proposed a hybrid GA with fully informed PSO (HGFA), in which a heuristic approach was used to improve the solution quality.

Most of the above mentioned algorithms used a common heuristic based on schedule generation scheme (SSGS), to improve the performance of the proposed algorithms. In SSGS, the activities were scheduled based on their possible earliest starting and finishing time while satisfying their temporal and resource constraints, with operating the same modes obtained from a prior process. Nevertheless, changing the modes of an activity may have a significant contribution to satisfy the resource constraints and to reduce project duration. In addition, some of the above algorithms used another heuristic to determine the best set of feasible modes iteratively. In other words, the heuristic iteratively changed a random activity's mode until the resources constraints were satisfied. However, this approach is computationally expensive for large-scale problems.

In this paper, we propose a new hybrid-multi-operator EA (H-MOEA) for solving a broad-range of MM-RCPSPs. In it, two heuristic approaches with two MOEAs are considered. The first heuristic is based on a linear programming approach, to obtain feasible modes by minimizing the duration of each activity. The second heuristic is to determine a quality solution, by scheduling activities based on their starting, finishing and selecting an alternative mode. Both MOEAs are used to generate new individuals for both activities' sequences and their modes, and they perform sequentially one after another in two different sub-populations, with one is given more emphasis than another when performs better. A number of well-known multimode test problems were taken from PSPLIB [13] and were solved using the proposed H-MOEA. The results obtained are compared with the recently published results in different literature, which shows that H-MOEA performs best.

The remainder of this paper is organized as follows: the definition and mathematical formulation of MM-RCPSP are discussed in Section 2, the proposed H-MOEA is in Section 3, the experimental results are in Section 4, and the conclusion and suggestions for future works are in Section 5.

2. MULTIMODE RCPSP

Traditional RCPSP schedules project activities while the sum of the resource requirements at any time must not exceed the total capacity, and their temporal between activities are satisfied. The objective of a RCPSP is to minimize the total make-span and it is often represented in activity-on-node format, which uses a directed acyclic graph [4]. In it, a project is represented with a set of nodes that correspond to a set of activities and their arcs represent the temporal relationships between activities.

Unlike a single mode RCPSP, a MM-RCPSP that have multiple modes for each activity. The decision manager can select any allowed mode for any activity while satisfying the resource constraints. Changing the mode of an activity changes the requirements of resources and duration. However, it is also noted that the mode of an activity must not be changed during its execution. In other words, if an activity starts with say the m^{th} mode, it must finish with the m^{th} mode.

In addition, for MM-RCPSP, there are two types of resources: renewable and non-renewable. The renewable resources, such as machine are available throughout the project horizon. On the contrary, the total available non-renewable resources are fixed for a project and if an activity uses N units non-renewable resources, the other activities can only use the remaining units.

2.1. Mathematical model

A MM-RCPSP consists of $D + 2$ activities with first and last activities being two dummy activities, refer to 'start' and 'finish' of the project. Each non-dummy activity (say,) has M_j possible modes, and their requirements for renewable and non-renewable resources, and durations are $r_{j,m}$, $v_{j,m}$ and $d_{j,m}$ where $m \in M$.

The objective of a MM-RCPSP is to minimize the total project duration, by achieving a best set of sequences of the activities and their optimal modes, while satisfying the renewable and non-renewable resources throughout, the project horizon.

The optimization problem for a MM-RCPSP is shown below, in which its parameters and variables are defined in Table 1.

$$\text{Minimize } FT_{(D+2),m}, m \in M \quad (1)$$

Subject to:

$$FT_{1,m} = 0, \forall m \quad (2)$$
$$FT_{i,m} \leq FT_{j,m} - d_{j,m}, \forall (j,i) \in A, \forall m \in M \quad (3)$$
$$\sum_{j \in A_t} r_{j,m,k} \leq R_k, \forall (j,t), \ \forall k \in K, \forall m \in M \quad (4)$$
$$\sum_{j=1}^{D+2} v_{j,m,n} \leq V_n, \ \forall m \in M, \forall n \in N \quad (5)$$

As for many RCPSP, the objective function in Eqn. (1) represents to minimize the finish time of the last dummy activity. Constraint (2) ensures that the first dummy activity finishes first without spending any duration. Constraint (3) maintains the temporal constraints between the activities, which means an activity cannot start until its predecessor finishes. Constraint (4) and (5) are two capacity constraints for renewable and non-renewable resources, respectively.

Table 1: List of Parameters and Variables

Notations:	Description
D	Number of non-dummy activities.
A	Set of activities, i.e., $i = 1,2, \dots, D$
M_j	Number of modes of the j^{th} activity, i.e., $m = 1,2, \dots, M$
$FT_{i,m}$	Finishtime of the i^{th} activity when it operates at mode $m \in M_j$.
K and N	Number of types of renewable and non-renewable resources, respectively.
R_k and V_n	Total number of available of the k^{th} type renewable and the n^{th} type non-renewable resource, respectively.
$r_{j,m}$ and $v_{j,m}$	Required number of renewable and non-renewable resources of the j^{th} activity when operates in the m^{th} mode.
$d_{j,m}$	Duration of the j^{th} activity when operates in the m^{th} mode.

3. NEW HYBRID-MULTI-OPERATOR EA

In this section, we present the proposed solution method, named H-MOEA, for MM-RCPSP. It is based on two multi-operator EAs, in conjunction with two-stage heuristic approaches. The first-stage considers a linear programming approach to obtain feasible modes from infeasible ones, while the second one obtains quality schedules during the evolutionary process.

H-MOEA starts with an initial population, in which the decision variables are considered as 'discrete' to represent activities sequences and their modes, as shown in subsection 3.1. Then, the random modes are passed to the first-stage-heuristic to obtain feasible modes, as described in subsection 3.2. With the feasible modes, the new resource requirements and duration of each activity are calculated. Then, each individual of the activities' schedule is updated, based on the proposed MM-forward- and backward-SSGS, as discussed in subsection 3.3. Subsequently, the individuals are sorted based on the minimum finish time of the last dummy activity. Then, new individuals of both activities' schedules and their modes are generated based on COA, as discussed in subsection 3.4. As some of the new individuals may contain infeasible modes, a first-stage-heuristic is used to rectify them. Then, the individuals of the activities' sequences are updated based on the forward and backward scheduling approaches. This process is continued until the predefined maximum number of schedules is reached. The pseudo-code of the proposed solution approach is given below, while its details are discussed in the subsequent subsections.

Algorithm 1: Pseudo-Code of the Proposed Solution Approach

Requirements: Population size and maximum number of generations, N_P and N_G respectively.
Initial population: Random representations of activities sequences ($\vec{x}_i, \forall i = 1,2,\ldots,N_P$) and their random modes ($\vec{y}_i, \forall i = 1,2,\ldots,N_P$).

1. $\boldsymbol{for}\ (k = 1{:}\,N_G)\ \boldsymbol{do}$ ➢ k represents the current generation number.
2. $\boldsymbol{for}\ (i = 1{:}\,N_P)\ \boldsymbol{do}$ ➢ i represents the current individual.
3. Apply linear-programming-based-heuristic to obtain feasible modes from infeasible ones, as shown in subsection 3.2.
4. Calculate new resource requirements and duration of the activities based on their updated modes.
5. Apply MM-forward and backward-SSGS to obtain a high quality schedule, as discussed in subsection 3.3.
6. Calculate the project duration of the $\vec{x}_i$ schedule, based on the finishtime of the $D + 2$ dummy activity.
7. $\boldsymbol{end\ for}\ (\boldsymbol{i})$
8. Sorted the individuals based on the minimum project durations, and $\boldsymbol{if}, k > 1$, select the best N_P individuals of both $\vec{x}_i$ and $\vec{y}_i$ from parents and offspring based on their minimum project durations.
9. Generate new individuals of both $\vec{x}_i$ and $\vec{y}_i, \forall i = 1,2,\ldots,N_P$ using a COA, as discussed in subsection 3.4.
10. $\boldsymbol{end\ for}\ (\boldsymbol{k})$

3.1. Initial Population and Representation

The decision variables of a MM-RCPSP are its activities' sequence and their operating modes. For example, if a MM-RCPSP has D number of activities and each has M number of modes, the representation of the decision variables is $\vec{z}_i = \{1,2,\ldots,D+2; m_1, m_2, \ldots, m_{D+2}\}, i \in N_P$, where '1' and 'D+2' are the two dummy activities for the 'start' and 'finish' of the project. $m_j \in M_j$ is the operating mode of the j^{th} activity. For simplicity, we consider the decision variable, as, $\vec{z} = \{\vec{x}, \vec{y}\}$, where $\vec{x}$ and $\vec{y}$ represent the decision variables for the activities' sequence and their modes, respectively.

In the initial population, N_P number of random individuals of $\vec{z}$ are generated, as:

$$\vec{x}_i \in perm\binom{\Delta}{\Delta}, \Delta = \{2,3,\ldots,D+1\}, \forall i = 1,2,\ldots,N_P \quad (6)$$

$$y_{i,j} = \mathbb{N} \cap [1, M_j], \forall j \in D, \forall i = 1,2,\ldots,N_P \quad (7)$$

$$\vec{z}_i = \{\vec{x}_i, \vec{y}_i\}, \forall i = 1,2,\ldots,N_P \quad (8)$$

where $perm\binom{\Delta}{\Delta}$ indicates the permutation or random combinations of the non-dummy activities and $\mathbb{N} \cap [1, M_j]$ represents a random integer number between 1 to M_j.

3.2. Obtain Feasible Modes

As the random modes may not always be feasible in terms of renewable and non-renewable resources requirements, we propose a linear programming approach that obtains the best quality feasible modes. To do this, we first formulate MM-RCPSP to minimize the duration of each activity, as:

$$\text{Minimize: } \sum_{j=2}^{D+1}\sum_{m=1}^{M_j} d_{j,m}w_{j.m}\,,\ \forall w \in \{0,1\} \tag{9}$$

Subject to:

$$\sum_{m=1}^{M_j} w_{j,m} = 1, \forall j \in \{2,3,\dots,D+1\} \tag{10}$$

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

$$\sum_{m=1}^{M_j} w_{j,m} \times r_{j,m,k} \leq R_k,\ \forall k \in K, \forall j \in \{2,3,\dots,D+1\} \tag{12}$$

(13)

$$0 \leq \mathcal{R}(w_{j,m}) \leq 1, \forall m \in M, \forall j \in \{2,3,\dots,D+1\} \tag{14}$$

where $w_{j,m}$ is a binary number, and $w_{j,m} = 1$ indicates that the j^{th} activity operates with the m^{th} mode. Eqn. (10) ensures that an activity can only operate a single mode, Eqns. (11) and (12) satisfy the non-renewable and renewable sources of the activities under their operated mode. However, it is important to note that the objective function of Eqn. (9) is linear and solving the problem using a linear programming approach always gives a unique and optimal solution. However, to maintain diversity for solving MM-RCPSP using H-MOEA, we need N_P number of different feasible modes (i.e., $\vec{y}_i, i = 1,2,\dots,N_P$) for N_P individuals of $\vec{x}_i, i = 1,2,\dots,N_P$. Therefore, the objective function of the above formulation is set differently for different individuals. Eqn. (11) is used for the best individual, while the following objective function is used for the rest of the individuals.

$$\text{Maximize: } \sum_{j=2}^{D+1}\sum_{m=1}^{M_j}\sum_{k=1}^{K} r_{j,m.k}w_{j.m}rand_{j,m}\,, \qquad rand_{j,m} \in \{0,1\}, \forall w \in \{0,1\} \tag{15}$$

Eqn. (15) is an alternative objective function of Eqn. (9), in which the usages of renewable-sources are maximized. Since most of the real-life modes of an activity have conflicting relationships between duration and resources, when the usages of resources are increased, the duration likely decreased. Moreover, the objective function of Eqn. (15) is multiplied by a random variable, that generates a random number between 0 and 1 for each mode of each activity. In other words, the random numbers are treated as weights of a mode of an activity. Once the above problem is solved, each time a new set of feasible modes is obtained. Note that we use the non-feasible set of modes obtained from the evolutionary process, as initial points when the above model is solved.

3.3. Obtain Feasible Schedules

As previously mentioned, when a new individual is generated from the initial population or subsequent generations, the individual may not be feasible in terms of the precedence and resource constraints of the activities. Although the feasible modes of the activities could be obtained from the first-stage heuristic as discussed in subsection 3.1, the precedence and resources constraints may not be satisfied. To obtain a feasible schedule from an infeasible one, the following MM-forward- and -backward-SSGS are used.

Algorithm 2: MM-forward SSGS

1. Get an individual of schedule, say $\vec{x} = \{x_1, x_2, \dots, x_{D+2}\}$.
2. Set, $j = 1, t = 0$.
3. ***while*** $(\vec{x} \neq \{\})$ → $\vec{x}$ *is not empty*
4. ***If*** the precedence of the x_j are satisfied
5.   ***do***
6.   At the t^{th} position, check the available renewable and non-renewable resources with those to be committed for the x_j^{th} activity, considering the current mode and another alternative random mode.
7.   ***If*** no such feasible mode is found
8.   Set $t = t + 1$
9.   ***Else***
10.   Get the feasible mode with minimum duration of the x_j^{th} activity.
11.   Schedule the x_j^{th} activity at the t^{th} position, and record its starttime, finishtime, and mode.
12.   Remove x_j from $\vec{x}$.
13.   ***EndIf***
14.   ***while*** (x_j is scheduled)
15. ***Else***
16.   Set, $j = j + 1$
17. ***EndIf***
18. ***EndWhile***

In the MM-forward-SSGS, the activities of $\vec{x}$ are scheduled, based on their earliest possible starting times, subject to satisfying their resource and precedence constraints. As shown in Algorithm 2, a valid activity is pushed to its earliest left possible position, by tuning its mode and satisfying the constraints. On the other hand, MM-backward-SSGS is used to sort all activities based on their finish times. In it, an activity is pushed to its latest right by tuning its mode and satisfying the constraints. In fact, when MM-backward-SSGS is applied to an individual, subsequently, its MM-forward-SSGS is applied to resort the activities.

3.4. Generating New Schedules and Modes

As said, although several EAs have been recently developed for RCPSPs [2], there are still some drawbacks to improve solution quality. Particularly, an EA often requires excessive parameter tunning to obtain the best quality solutions for a RCPSP [2]. As mentioned earlier, COA Elsayed, Sarker, Ray and Coello [2] showed good performance in solved single mode RCPSPs. In this research, we extend COA for solving MM-RCPSPs. COA starts with a single population and its individuals are evolved using MOGA and MODE. Initially, their probabilities are set to 1, which means both algorithms have equal opportunities to evolve the same number of individuals. However, their probabilities are updated in subsequent generations, based on their success rate on generating successful offspring from their parents. Therefore, the better performing algorithm gets a higher chance to evolve more individuals, and vice-versa. Based on the probability of an algorithm, new individuals of both schedules and modes are generated. The details of COA can be found in [2].

3.5. Selection Operator

Once the feasible modes of all new individuals are obtained (see subsection 3.2), the individuals are passed to the MM-forward- and backward-SSGS (see subsection 3.3) in order to obtain a better quality solutions. Then, the project durations of all individuals are determined, based on the finishtime of the dummy activity. In the selection operator, the better individuals between parents and offspring are selected, based on their minimum project durations.

3.6. Stopping Criterion

H-MOEA terminates if the total number of schedules is greater than a predefined maximum number of schedules. Note that the algorithm counts two schedules per individual evaluation, with one for MM-forward-SSGS and another for MM-backward-SSGS.

4. COMPUTIONAL EXPERIMENTS

The performance of the proposed algorithm is evaluated by solving the well-known benchmark sets of J10, J12, J14, J16, J18 and J20. Each set contains 60 instances with 10 problems in each instance.

For a fair comparison, the algorithm runs 30 times, and each terminates after 5000 schedules as the stopping criterion. The population size of the algorithm is set to 10 and its other parameters could be found in [1].

The algorithm is implemented in Matlab 2018a in a desktop computer with a 3.4 GHZ Intel Core i7 processor and 16 GB Ram.

4.1. Comparisons with State-of-the-art Algorithms

In this subsection, the well-known benchmark sets for MM-RCPSP are solved using the proposed H-MOEA. The results obtained are compared with the known optimal results in PSPLIB [13]. In addition, the performance of H-MOEA is demonstrated by comparing the obtained results with those reported in different recent publications. The names of the state-of-the-art algorithms are set based on the authors name and their proposed algorithms, with a full list can be found in [3].

To compare the results, the average percentage of deviations (Δ) from their known optimal values are calculated, as [2]:

$$\Delta = 100 \times \frac{1}{N} \sum_{n=1}^{N} \frac{Makespan_n - OPT_n}{OPT_n} \tag{16}$$

where OPT_n is the known optimal value of the n^{th} instance and $Makespan_n$ is its obtained project duration from H-MOEA. N is the total number of problems of each benchmark set, i.e., $N = 537$ for J10, $N = 548$ for J12, $N = 552$ for J14, $N = 551$ for J16, $N = 553$ for J18 and $N = 555$ for J20.

The comparison results for all benchmark sets are listed in Table 2 where the algorithms are sorted based on ascending performances for the J20 instance. As seen from that, the proposed H-MOEA obtains 100% optimal solutions for both J10 and J12. In fact, the minimum Δ are found in J14, J16 and J20 in comparison to those obtained from the state-of-the-art algorithms. Although the obtained results for J18 is not the best, it's very competitive comparing to other algorithms. Nevertheless, the average Δ for all the benchmark sets is found to be minimum in our algorithm.

Table 2: Average Δ for the well-known MM-RCPSP for 5000 schedules

Algorithm	J10	J12	J14	J16	J18	J20	Avg. Δ
WLZO [14]	0.28	0.79	1.18	2.75	NR	NR	1.25
JSA [15]	1.16	1.73	2.60	4.07	5.52	6.74	3.64
CLPE [16]	1.06	0.41	3.43	3.36	2.12	5.78	2.69
CHA [17]	0.32	NR	NR	NR	NR	2.05	1.19
AGA [10]	0.24	0.73	1.00	1.12	1.43	1.91	1.07
ZLZH [18]	0.00	NR	NR	NR	NR	1.82	0.91
TCGLS [19]	0.33	0.52	0.93	1.08	1.32	1.69	0.98
RSS [20]	0.18	0.65	0.89	0.95	1.21	1.64	0.92
EFEA [21]	0.14	0.24	0.77	0.91	1.30	1.62	0.83
JRR [22]	0.28	0.41	0.54	0.75	0.92	1.55	0.74
LCSFLA [12]	0.10	0.21	0.46	0.58	0.94	1.40	0.62
LZA [23]	0.09	0.13	0.40	0.57	1.02	1.10	0.55
SEEDA [24]	0.09	0.12	0.36	0.42	0.85	1.09	0.49
LCEDA [25]	0.12	0.14	0.43	0.59	0.90	1.28	0.58
CYDP [26]	NR	NR	NR	NR	NR	0.97	0.97
SLC [27]	0.05	0.21	0.46	0.82	1.21	1.62	0.73
LHGA [28]	0.06	0.17	0.32	0.44	0.63	0.87	0.42
MAN [29]	0.05	0.09	NR	0.22	**0.18**	0.80	0.27
CWC [30]	0.01	NR	NR	NR	NR	0.71	0.36
VPVAIS [31]	0.02	0.07	0.20	0.39	0.52	0.70	0.32
VPVGA [5]	0.01	0.09	0.22	0.32	0.42	0.57	0.27
SAAGA [32]	0.03	0.07	0.21	0.32	0.40	0.56	0.27
HGFA [3]	0.02	0.08	0.13	0.28	0.29	0.41	0.20
Proposed H-MOEA	**0.00**	**0.00**	**0.11**	**0.08**	0.31	**0.41**	**0.15**

Fig. 1 shows a sample Gantt chart for the J10 benchmark of its 37^{th} instance and its 9^{th} problem. It is seen that H-MOEA obtains the optimal solution (which is known) by obtaining its optimal modes. It is also seen that the usages of non-renewable resources are cumulatively increased throughout the project duration. At the end, total available resources have been fully utilized. On the contrary, the total number of renewable resources are available over the project horizon. At each time period, the summation of resources committed by the activities scheduled at that time period, must not exceed the available units.

4.2. Statistical Test

In this subsection, the above algorithms are statistically compared shown in Table 2. To do this, we use a Friedman ranking test, in which the test problems are considered as samples. Table 3 shows the mean rank of all algorithms, in which it is shown that the proposed H-MOEA is the best algorithm for solving all of the considered test problems.

Table 3: Mean Ranks of the algorithms from Freidman Test

Alg.	WLZO	JSA	CLPE	CHA	AGA	ZLZH	TCGLS	RSS	EFEA	JRR	LCSFLA	LZA
Rank	19.75	20.50	19.08	21.67	17.50	18.25	17.00	15.42	14.25	13.50	11.50	9.92
Alg.	SEEDA	LCEDA	CYDP	SLC	LHGA	MAN	CWC	VPVAIS	VPVGA	SAAGA	HGFA	**H-MOEA**
Rank	8.42	10.67	20.33	12.08	8.17	7.67	16.42	4.67	4.58	4.17	3.00	**1.50**

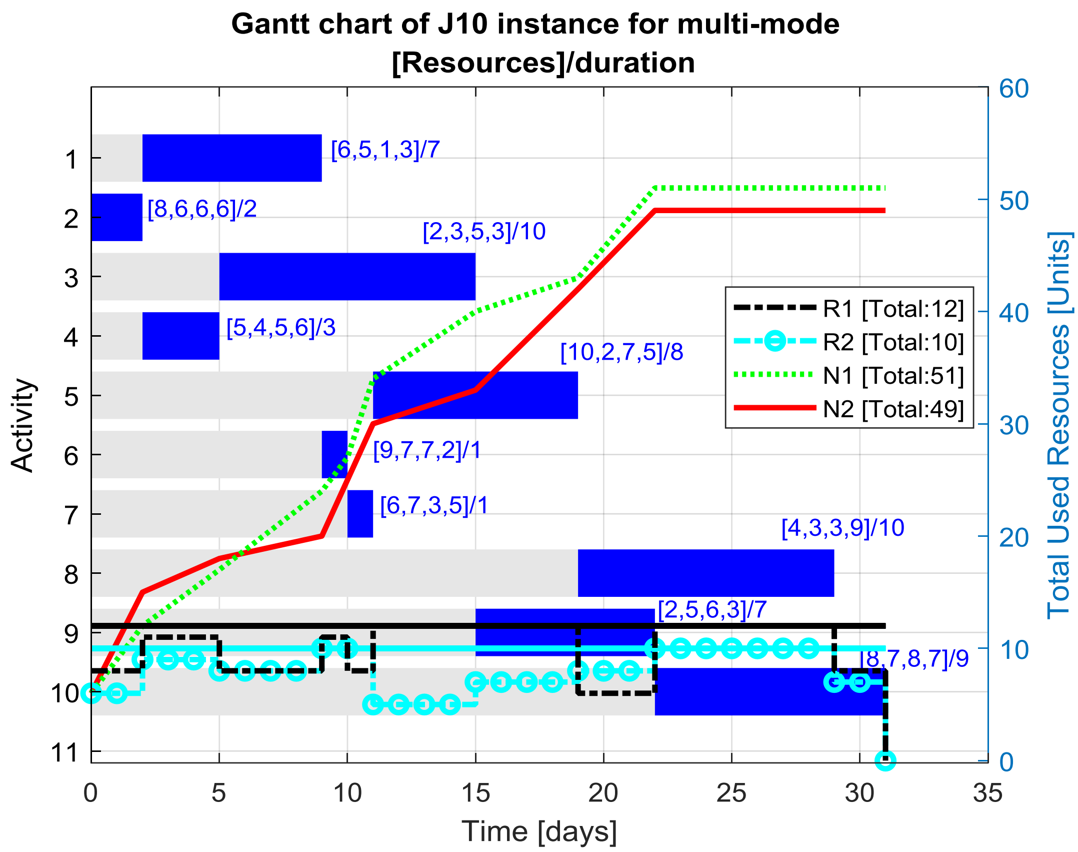

Fig. 1: A SAMPLE GANTT CHART FOR $J10_{37,9}$ PROBLEM

5. CONCLUSION AND RECOMMENDATION FOR FUTURE WORKS

In this study, H-MOEA was designed to solve MM-RCPSP, which has been proven to be a strong NP-hard problem. One of the main difficulties of solving this problem using EAs, is to obtain feasible modes over the solution process. This paper proposed a linear-programming-approach to find the best quality feasible modes in the quickest time period. To obtain the best quality schedule of the activities of a project, two MM -forward- and -backward-SSGS were described. Both SSGS were not only used to satisfy the temporal constraints between the activities, they changed the activities' modes to reduce duration, while satisfying the available renewable and non-renewable resources. In the algorithm, to avoid excessive parameter tunning, we used two self-adaptive MOEAs to determine the best algorithm for solving a particular problem. In it, two algorithms performed simultaneously but different numbers of individuals, based on their prior performance on generating successful offspring.

The proposed H-MOEA was tested by solving the well-known MM-RCPSP benchmark sets of J10, J12, J14, J16, J18 and J20. It was found that H-MOEA obtained the best quality solutions, comparing to the recent state-of-the-art algorithms. In fact, it obtained 100% optimal solutions for J10 and J12 benchmark sets.

In future, H-MOEA can be tested by solving some large sets of benchmarks. Considering resource uncertainties and disruptions in MM-RCPSP and solved them using H-MOEA could be another possible research direction.

6. ACKNOWLEDGMENT

This research has been funded by Australian Research Council Discovery Project DP170102416.

7. REFERENCES

[1] **M. F. Zaman, S. M. Elsayed, T. Ray, and R. A. Sarker. 2018 of Conference**. Scenario-based Solution Approach for Uncertain Resource Constrained Scheduling Problems, 2018 IEEE World Congress on Computational Intelligence (IEEE WCCI),July 2018,pp 1-8.

[2] **S. Elsayed, R. Sarker, T. Ray, and C. C. Coello. 2017**. Consolidated optimization algorithm for resource-constrained project scheduling problems, Information Sciences, 418-419, pp 346-362.

[3] **M. H. Sebt, M. R. Afshar, and Y. Alipouri. 2017**. Hybridization of genetic algorithm and fully informed particle swarm for solving the multi-mode resource-constrained project scheduling problem, Engineering Optimization, 49(3), pp 513-530.

[4] **R. K. Chakrabortty, R. A. Sarker, and D. L. Essam. 2017**. Resource constrained project scheduling with uncertain activity durations, Computers & Industrial Engineering, 112, pp 537-550.

[5] **V. V. Peteghem, and M. Vanhoucke. 2010**. A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem, European Journal of Operational Research, 201(2), pp 409-418.

[6] **J. Alcaraz, and C. Maroto. 2001**. A Robust Genetic Algorithm for Resource Allocation in Project Scheduling, Annals of Operations Research, 102(1-4), pp 83-109.

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

[8] **H. Zhang, X. Li, H. Li, and F. Huang. 2005**. Particle swarm optimization-based schemes for resource-constrained project scheduling, Automation in Construction, 14(3), pp 393-404.

[9] **S. Hartmann. 2001**. Project Scheduling with Multiple Modes: A Genetic Algorithm, Annals of Operations Research, 102(1), pp 111-135.

[10] **J. Alcaraz, C. Maroto, and R. Ruiz. 2003**. Solving the Multi-Mode Resource-Constrained Project Scheduling Problem with genetic algorithms, Journal of the Operational Research Society, 54(6), pp 614-626.

[11] **J. Coelho, and M. Vanhoucke. 2011**. Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers, European Journal of Operational Research, 213(1), pp 73-82.

[12] **L. Wang, and C. Fang. 2011**. An effective shuffled frog-leaping algorithm for multi-mode resource-constrained project scheduling problem, Information Sciences, 181(20), pp 4804-4822.

[13] **R. Kolisch, and A. Sprecher. 1997**. PSPLIB - A project scheduling problem library, European Journal of Operational Research, 96(1), pp 205-216.

[14] 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)*, 2014, pp. 680-690.

[15] **J. Józefowska, M. Mika, R. Rózycki, G. Waligóra, and J. Węglarz. 2001**. Simulated Annealing for Multi-Mode Resource-Constrained Project Scheduling, Annals of Operations Research, 102(1-4), pp 137-155.

[16] **A. H. L. Chen, Y. C. Liang, and J. D. Padilla. 2014**. An entropy-based upper bound methodology for robust predictive multi-mode RCPSP schedules, Entropy, 16(9), pp 5032-5067.

[17] **C. W. Chiang, and Y. Q. Huang. 2012 of Conference**. 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.

[18] **L. Zhang, Y. Luo, and Y. Zhang. 2015**. Hybrid Particle Swarm and Differential Evolution Algorithm for Solving Multimode Resource-Constrained Project Scheduling Problem, Journal of Control Science and Engineering, 2015, pp.

[19] **L. Y. Tseng, and S. C. Chen. 2009**. Two-phase genetic local search algorithm for the multimode resource-constrained project scheduling problem, IEEE Transactions on Evolutionary Computation, 13(4), pp 848-857.

[20] **M. Ranjbar, B. De Reyck, and F. Kianfar. 2009**. A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling, European Journal of Operational Research, 193(1), pp 35-48.

[21] **S. Elloumi, and P. Fortemps. 2010**. A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem, European Journal of Operational Research, 205(1), pp 31-41.

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

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

[24] **O. S. Soliman, and E. A. R. Elgendi. 2014**. 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, 8, pp 57-64.

[25] **C. Fang, and L. Wang. 2012**. An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem, Computers and Operations Research, 39(5), pp 890-901.

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

[27] **H. Shen, and X. Li. 2013 of Conference**. 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.

[28] **A. Lova, P. Tormos, M. Cervantes, and F. Barber. 2009**. An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes, International Journal of Production Economics, 117(2), pp 302-316.

[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*, 2012, pp. 231-242.

[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)*, 2013, pp. 545-555.

[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)*, 2009, pp. 85-96.

[32] **M. H. Sebt, M. R. Afshar, and Y. Alipouri. 2015**. 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, 2(3), pp 905-924.