 DOI: 10.9744/duts.10.2.232-251

# MULTI-RESOURCE ALLOCATION AND LEVELING IN MULTI-PROJECT SCHEDULING PROBLEM WITH HYBRID-CHROMOSOME NON-DOMINATED SORTING GENETIC ALGORITHM II

Nathaniel Alvin[1] and I-Tung Yang[2]
[1] Graduate Student of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taiwan and Master Program in Civil Engineering, Petra Christian University, Indonesia
[2] Professor of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taiwan
[1] b21210003@john.petra.ac.id, [2] ityang@mail.ntust.edu.tw

**ABSTRACT:** *Multi-resource allocation and leveling in multi-project (MR-AL-MP) scheduling refers to the attempt of producing a project schedule with minimum project duration and maximum resource utilization while complying with all precedence and resource availability constraints in a multi-project environment involving multiple resources. This study proposes a model that integrates both resource allocation and leveling models into a unified framework. This study develops a modified version of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II), called Hybrid-Chromosome NSGA-II, as the optimization algorithm. For validation purposes, the performance of Hybrid-Chromosome NSGA-II is compared with two benchmark metaheuristic algorithms which are Multi-Objective Particle Swarm Optimization (MOPSO) and Multi-Objective Symbiotic Organisms Search (MOSOS) in optimizing a case study. It is shown that the proposed model and algorithm are able to produce a set of non-dominated solutions that represent the feasible trade-off relationships between the objectives. Furthermore, the Hybrid-Chromosome NSGA-II is superior to MOPSO and MOSOS in terms of the quality, spread, and diversity of the solutions.*

Keywords: *resource allocation, resource leveling, multi-project scheduling problem, optimization, metaheuristic, hybrid-chromosome NSGA-II*

## 1. INTRODUCTION

Project management is the application of skills, methods, or techniques in order to achieve certain project objectives, and one of them is a specific completion date that has to be satisfied based on the relevant contract document (Son and Matilla, 2004). Several well-known techniques, such as basic Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT), have been extensively used as major tools for construction project scheduling. However, those techniques usually assume unlimited resource availability, which is not relevant to practical circumstances (Zhang et al., 2006), and also possibly cause undesirable resource fluctuations and peak demands (Jun and El-Rayes, 2011). Resource fluctuations and peak demands are inefficient and costly to implement due to the possibility of disrupting the learning curve effects, needing to release and rehire resources on a short-term basis, and even causing some resources to be idle on the site, especially during low demand periods (El-Rayes and Jun, 2009). Therefore, it is important to schedule the project activities subject to the constraints of resource availability and resource usage variation.

The resource allocation model, or also known as the resource-constrained model, attempts to overcome the resource conflicts that may occur between activities while maintaining the shortest possible project duration. On the other hand, the resource leveling model is used to minimize the resource fluctuations within a fixed project duration to achieve maximum resource utilization efficiency. Despite being two individual subproblems (Hegazy, 1999), these two models are both essential to produce a project schedule with the shortest possible duration and maximum resource utilization efficiency while complying with all the resource availability constraints. The present study aims to integrate both resource allocation model and resource leveling model in a single unified model.

## 2. LITERATURE REVIEW

### 2.1. Resource Allocation Model

The resource allocation model is a model used to produce a project schedule with minimum project duration while complying with all the resource availability constraints, or also known as resource-constrained project scheduling problem (RCPSP). Basically, the RCPSP can be formulated as follows (Zhang et al., 2005):

$$\min\{maxF_i | i = 1,2, \dots, N\} \tag{1}$$

Subject to

$$F_j \leq F_i - D_i, \forall j \in P_i; i = 1,2, \dots, N \tag{2}$$

$$\sum_{A_t} r_{ik} \leq R_k, k = 1,2, \dots, K; t = S_1, S_2, \dots, S_N \tag{3}$$

Where:
$N$ = Number of activities
$F_i$ = Finish time of activity $a_i$
$D_i$ = Duration of activity $a_i$
$P_i$ = Predecessors of activity $a_i$
$R_k$ = Amount of resource $k$ available
$K$ = Number of resource types
$r_{ik}$ = Amount of resource $k$ needed for activity $a_i$
$A_t$ = Ongoing activities at $t$
$S_i$ = Start time of activity $a_i$

### 2.2. Resource Leveling Model

The resource leveling model can be used to minimize the resource usage fluctuations and avoid peak demands that might occur in a construction project (Zhao et al., 2006). Basically, the resource leveling model works by shifting or moving the non-critical activities within their available total floats in order to produce a project schedule with the best resource utilization (Harris, 1978; Popescu and Charoenngam, 1995).

To handle resource leveling, researchers have been developing various metrics, with one of the first and most common metrics being the sum of squares method (Burgess and Killebrew, 1962). Other metrics, such as the deviation between the required and available resources

(Chan et al., 1996), resource intensity (Guo et al., 2009), and release and rehire the resource idle days (El-Rayes and Jun, 2009), have also been developed by various researchers. Most of the more conventional metrics may not be as practical as they usually aim to match a predetermined resource profile, especially uniform shape (Ponz-Tienda et al., 2017), and thereby overlooking possible alternatives shapes of resource profiles that are efficient.

Instead of just matching a predetermined resource profile, the two metrics developed by El-Rayes and Jun (2009), release and rehire ($RRH$) and resource idle days ($RID$), have the ability to directly measure undesirable fluctuations, and thus they are adopted in this study. Basically, $RRH$ measures the resource release and rehire that occurs in a project if such practice is allowed, while $RID$ measures the idle days of each resource if the idle resources are required to stay on site instead of being released and rehired at a later time. Since these two issues are different and do not occur concurrently, only one of the two metrics will be considered at a time. Furthermore, since $RRH$ and $RID$ only attempt to minimize the occurrence of the valley shapes, it completely overlooks the need of minimizing the mountain shapes in the resource profile. Therefore, a third metric developed by Guo et al. (2009) called resource intensity ($RI$) is also employed as one of the objectives in order to reduce resource usage fluctuations in general. The formulas of the three objective functions are expressed in equation (4), (5), (6), (7), (8), (9), and (10) below.

$$RRH = H - MRD = \frac{1}{2} \times HR - MRD \tag{4}$$

$$HR = \left[ r_1 + \sum_{t=1}^{T-1} |r_t - r_{t+1}| + r_T \right] \tag{5}$$

$$MRD = Max(r_1, r_2, \dots, r_T) \tag{6}$$

Where:

$RRH$ = Release and rehire
$H$ = Total increases in the daily resource demand
$MRD$ = Maximum resource demand
$HR$ = Total daily resource fluctuations
$T$ = Total project duration
$r_t$ = Resource demand on day $t$
$r_{t+1}$ = Resource demand on day $(t+1)$

$$RID = \sum_{t=1}^{T} [Min\{Max(r_1, r_2, \dots, r_t), Max(r_t, r_{t+1}, \dots, r_T)\} - r_t] \tag{7}$$

Where:

$RID$ = Resource idle days
$T$ = Total project duration
$r_t$ = Resource demand on day $t$
$r_{t+1}$ = Resource demand on day $(t+1)$

$$\boldsymbol{RI} = \frac{\boldsymbol{1}}{\boldsymbol{T}} \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}} \sum_{\boldsymbol{m=1}}^{\boldsymbol{p}} \left[ \boldsymbol{w_m} (\boldsymbol{SR_m}(\boldsymbol{t}) - \overline{\boldsymbol{SR_m}})^2 \right] \tag{8}$$

$$\overline{SR_m} = \frac{1}{T} \sum_{t=1}^{T} SR_m(t) \tag{9}$$

$$SR_m(t) = \lambda R_m(t)/Rmax_m \tag{10}$$

Where:
$RI$ = Resource intensity
$T$ = Overall project duration
$w_m$ = Weight of resource $m$
$SR_m(t)$ = Relative demand of resource $m$ in all projects on day $t$
$\overline{SR_m}$ = Average of relative demand
$\lambda$ = Amplifying coefficient [1,100]
$R_m(t)$ = Total demand of resource $m$ in all projects on day $t$

### 2.3. Multi-Resource Allocation and Leveling in Multi-Project Scheduling Problem

The main objectives in resource allocation and leveling problem are minimizing the project duration and resource fluctuations that occur. The basic concept is to apply the resource allocation model first in order to find a new project schedule whose resource profile satisfies the resource availability constraints. Then, apply the resource leveling model by shifting the non-critical activities based on the total floats (the amount of time an activity can be delayed without delaying the overall project duration) from the newly obtained schedule. In the past few decades, there have been various researchers that attempt in combining these two models together described as follows.

a. Hegazy (1999) developed a new model called double moments as an improvement of a previously developed heuristic resource leveling metrics called minimum moment. This new metric was used to solve a single project problem that consists of 20 activities and six types of resources using Genetic Algorithms (GAs) as the optimization algorithm.
b. Wu et al. (2008) attempted to solve a multi-resource allocation and leveling optimization using a self-adaptive ant colony algorithm and further conducted a comparison study with genetic algorithms. This study used minimum moment around the vertical axis of the resource histogram as the resource leveling metric.
c. Jun and El-Rayes (2011) developed a three-module novel multi-objective optimization model to minimize undesirable resource fluctuations and project duration while complying with all available constraints. Their study used release and rehire and resource idle days as the resource leveling metrics and multi-objective genetic algorithm as the optimization algorithm to solve a single project scheduling problem with multiple resources.
d. Koulinas and Anagnostopoulos (2013) used a tabu search-based hyper-heuristic algorithm to solve resource leveling problems with resource availability constraints and predetermined maximum project duration. In this study, the resource leveling objective is to minimize the sum of the squared deviations of the resource requirements to achieve the ideal resource histogram, which is the rectangle shape.
e. Khanzadi et al. (2016) used two new metaheuristic algorithms, colliding body optimization and charged system search, to solve a single-project resource allocation and leveling problem. This study minimizes the resource moment deviation on the horizontal axis as the resource leveling objective.

In the real-life situation, construction companies usually work on multiple projects at the same time (Chen and Shahandashti, 2009). In order to schedule multiple projects simultaneously, there are two approaches that can be used, namely multi-project approach and mono-project approach. In the multi-project approach, each project has its own 'start' and 'end' dummy activities, whereas in mono-project approach, all the projects are combined into a single project by adding 'start' and 'end' dummy activities to connect all the projects (Lova and Tormos, 2001).

Furthermore, construction projects always involve multiple resources, which make the optimization process more complex. Guo et al. (2009) stated that in a multi-resource problem, since each resource has different demand, it is necessary to transform absolute demand into relative demand by dividing it by the corresponding maximum resource demand so that the resources are comparable in quantity. The use of weight factors is also essential to represent the degree of importance of each resource. A bigger value of weight factor indicates a higher priority of the corresponding resource.

### 2.4. Multi-Objective Optimization

Multi-objective optimization is used to solve optimization problems with two or more objectives that are usually conflicting with each other (Cui et al., 2017). The goal is to obtain a set of acceptable trade-off solutions from which a decision maker can select, which is usually referred to as the Pareto front. In the attempt of comparing the possible solutions, Pareto optimality concepts are often used (Ngatchou et al., 2005).

In the world of optimization, two methods are mainly used, namely analytical method and numerical method. The analytical method solves a problem by finding the exact solution, which is not suitable to be used in a rather complex problem. Conversely, the numerical method, such as the heuristic algorithm, works through a series of iterations to get an acceptable near optimal solution (Cui et al., 2017). As the latest generation of the heuristic algorithm (Alcaraz and Maroto, 2001), metaheuristic algorithms (strategies that guide the search to efficiently find the near–optimal solutions) are proven to be more viable and superior than the other traditional methods (Sörensen and Glover, 2013), and thereby more often used in the present time, especially for multi-objective optimization problems. Some of the most popular multi-objective metaheuristic algorithms are Non-Dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002) and Multi-Objective Particle Swarm Optimization (MOPSO) (Coello and Lechuga, 2002). Recently, a rather new and interesting multi-objective metaheuristic algorithm called Multi-Objective Symbiotic Organisms Search (MOSOS) (Tran et al., 2016) has also been developed. For the optimization process, this study proposes a slightly modified version of NSGA-II, called hybrid-chromosome NSGA-II, whose performance will be compared with both MOPSO and MOSOS algorithm.

## 3. RESEARCH METHODOLOGY

The proposed model consists of two phases that work sequentially. The first phase, which focuses on the resource allocation, attempts to create a new project schedule that satisfies all resource availability constraints using priority-based representation as the solution representation. The priority-based representation works by shifting the activities with conflicting resources based on their Priority Value ($P_n1$). Activities with greater $P_n1$ will be prioritized to stay, whereas activities with smaller $P_n1$ will be shifted first. The new schedule obtained from the first phase will then be further adjusted in the second phase.

The second phase focuses on implementing the resource leveling model to shift the non-critical activities of the newly obtained schedule based on their available total float. The number of days an activity will be shifted is determined by its Shift Value ($S_n$). The $S_n$ of each activity is represented by a certain number that plays a part in determining the start day of an activity. Furthermore, this phase also adopts priority-based representation, denoted by $P_n2$, to determine which activities are eligible in getting that portion of total float first. Activities with greater $P_n2$ will be prioritized in getting the total float.

This study proposes the use of a modified NSGA-II called Hybrid-Chromosome NSGA-II. The name comes from the different characteristics that the decision variables have. In the proposed model, the decision variables consist of two different types of numbers, specifically permutation sequence to represent the $P_n1$ and $P_n2$ and continuous real value to represent the $S_n$. Therefore, this model requires different types of crossover and mutation operators. For the crossover operator performed on $P_n1$ and $P_n2$, the single point partially-mapped crossover is used. Conversely, for the $S_n$, this model adopts a commonly used single point crossover operator that works by swapping a certain part of a chromosome with another chromosome. For the mutation operators, this model works by swapping two different genes for the $P_n1$ and $P_n2$ and assigning a new random number for the $S_n$. The schedule generation process will be explained using an illustration of a chromosome illustrated in Figure 1.

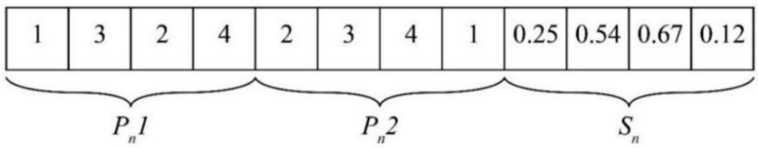


Figure 1. Chromosome illustration

As mentioned before, the schedule generation begins with the implementation of the resource allocation model based on the decision variable $P_n1$. From the chromosome presented in Figure 1, each activity 1, 2, 3, and 4 has a $P_n1$ of 1, 3, 2, and 4, respectively. Therefore, activity 4 has the highest priority, while activity one has the lowest priority. Based on $P_n1$, the activities will be shifted until all the resource availability constraints are satisfied. From this process, a new schedule is obtained.

The new schedule will be modified again using the resource leveling model based on the new total float of each activity, $P_n2$, and $S_n$. According to Figure 1, each activity 1, 2, 3, and 4 has a $P_n2$ of 2, 3, 4, and 1, respectively. It means that activity 3, with a $P_n2$ of 4, has the highest priority in getting the total float first, while activity 4 has the lowest. The amount of total float for an activity depends on the corresponding $S_n$. Firstly, in order to guarantee that this process does not produce a schedule that violates the resource availability constraints, the shifting of each activity involves a checking process that is adopted from the study carried out by Jun and El-Rayes (2011). The activity shifting is performed on the non-critical activities, whose total floats are not 0, by performing the following steps.

A numerical example is also provided below, assuming activity 3 is not critical, has a duration of 1 and a total float of 5, and starts on day 4, based on the new schedule. An illustration of the activity shifting is given in Figure 2.

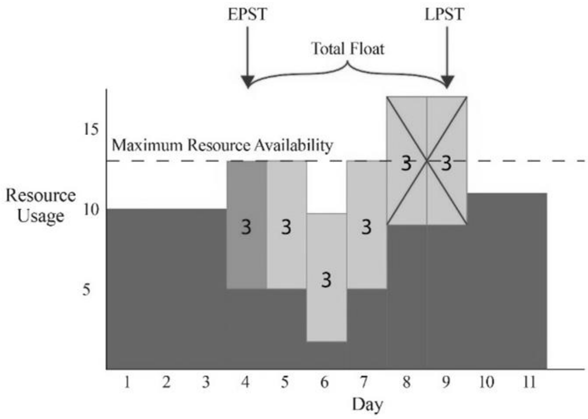


Figure 2. Activity shifting illustration

1. Based on the information provided, define the start time of activity 3, which is 4, as the earliest possible start time ($EPST$). The latest possible start time ($LPST$) is obtained by adding the total float to the start time, which is calculated as $LPST = 4 + 5 = 9$.
2. From the $EPST$ and $LPST$, it can be concluded that activity 3 can possibly start on between day 4 and 9.
3. Check if any of those days will produce a schedule that violates the resource availability constraints. Based on Figure 2, it can be seen that the constraints are violated if activity 3 starts on day 8 and 9.
4. By eliminating day 8 and 9 as the start time of activity 3, the possible start time ($PST$) of activity 3 will be $PST_3 = [4, 5, 6, 7]$.
5. From the previously obtained $PST$, the total number of the possible start time ($TPST$) of activity 3 is defined as 4, as there are four possible start times.
6. From there, the matrix index that represents the new start time of each activity, based on its corresponding $PST$, is obtained by following equation (11).

$$SSD_n = \lceil S_n \times TPST_n \rceil \tag{11}$$

   Where:
   $SSD_n$ = Selected start day of activity $n$
   $S_n$ = Shift value of activity $n$
   $TPST_n$ = Total possible start time of activity $n$
   Then, the matrix index of $PST_3$ can be calculated as: $SSD_3 = \lceil 0.67 \times 4 \rceil = \lceil 2.68 \rceil = 3$. Therefore, from $PST_3 = [4, 5, 6, 7]$, the chosen start day of activity 3 is 6.
7. These steps are repeated for all the non-critical activities.

The proposed framework of the Hybrid-Chromosome NSGA-II for handling the multi-resource allocation and leveling in multi-project (MR-AL-MP) scheduling problem is illustrated through a flowchart presented in Figure 3.

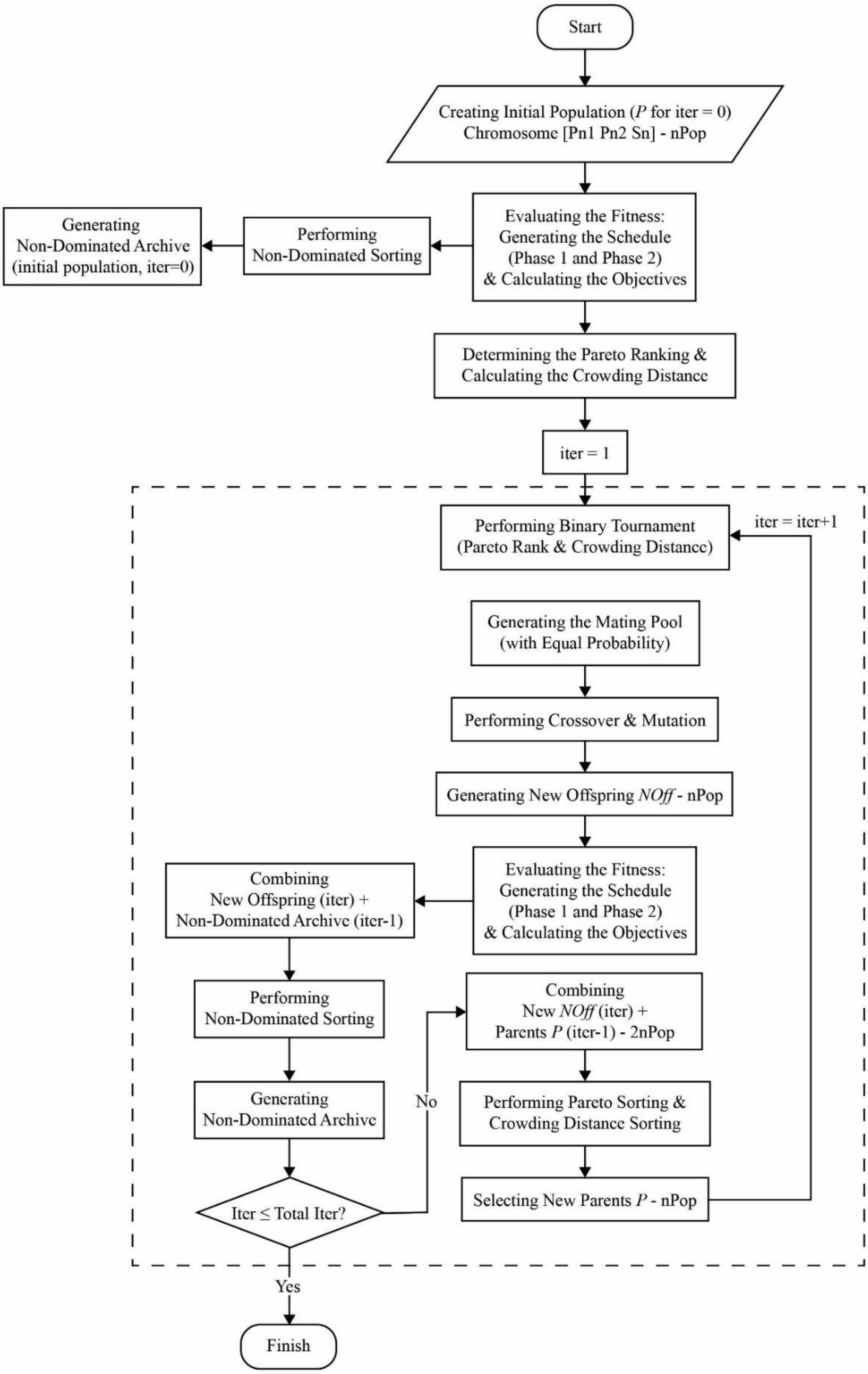


Figure 3. Flowchart of the proposed hybrid-chromosome NSGA-II algorithm

## 4. CASE STUDY

This case study is used to evaluate the performance of the Hybrid-Chromosome NSGA-II by comparing it with two other metaheuristic algorithms (MOPSO and MOSOS) in terms of the hypervolume and the computational time obtained from five runs. The approximate hypervolume is obtained by computing the number of solutions dominated by the pareto fronts out of one million solutions that are spreading uniformly. For the case study, two experiments are conducted. The first experiment attempts to minimize the project duration, $RRH$, and $RI$, whereas the second experiment attempts to minimize the project duration, $RID$, and $RI$.

### 4.1. Project Information

The case study is retrieved from Dalfard and Ranjbar (2012). The case study consists of five projects with each of them having six activities and involving four different types of resources (R1, R2, R3, and R4). The daily resource usage is limited to 35 for each type of resource. This study assumes that each resource has the same weight factor. The complete data and the precedence networks of all the projects are presented in Table 1 and Figure 4.

Table 1. Project data

| **Activity** | **Duration (days)** | **Predecessors** | **Daily Resource Requirements** | | | |
|---|---|---|---|---|---|---|
| | | | **R1** | **R2** | **R3** | **R4** |
| 1 | 1 | - | 1 | 9 | 6 | 10 |
| 2 | 3 | - | 3 | 7 | 0 | 7 |
| 3 | 4 | 1, 2 | 6 | 8 | 2 | 1 |
| 4 | 5 | 1, 2 | 7 | 5 | 8 | 2 |
| 5 | 2 | 3, 4 | 8 | 4 | 0 | 2 |
| 6 | 2 | 3, 4 | 6 | 0 | 8 | 9 |
| 7 | 5 | - | 10 | 3 | 7 | 10 |
| 8 | 2 | - | 8 | 6 | 8 | 2 |
| 9 | 1 | - | 6 | 0 | 3 | 0 |
| 10 | 5 | 7, 8 | 8 | 0 | 0 | 10 |
| 11 | 2 | 9, 10 | 10 | 9 | 9 | 10 |
| 12 | 4 | 7, 8, 9 | 4 | 2 | 6 | 2 |
| 13 | 4 | - | 10 | 5 | 3 | 0 |
| 14 | 4 | - | 0 | 3 | 0 | 6 |
| 15 | 2 | - | 0 | 8 | 3 | 6 |
| 16 | 3 | 13, 14 | 9 | 5 | 2 | 10 |
| 17 | 4 | 13, 14, 15 | 5 | 0 | 0 | 4 |
| 18 | 5 | 15, 16 | 2 | 6 | 0 | 0 |
| 19 | 2 | - | 4 | 8 | 9 | 0 |
| 20 | 3 | - | 0 | 4 | 1 | 9 |
| 21 | 2 | - | 9 | 2 | 7 | 0 |
| 22 | 3 | 19, 21 | 0 | 7 | 3 | 0 |
| 23 | 1 | 20, 22 | 8 | 9 | 6 | 6 |
| 24 | 3 | 19, 20, 21 | 0 | 10 | 0 | 3 |
| 25 | 2 | - | 0 | 8 | 0 | 0 |
| 26 | 1 | 25 | 0 | 2 | 8 | 0 |
| 27 | 1 | 25 | 0 | 0 | 8 | 3 |
| 28 | 5 | 25 | 2 | 8 | 0 | 0 |
| 29 | 5 | 26, 27, 28 | 0 | 1 | 0 | 0 |
| 30 | 1 | 26, 27, 28 | 0 | 4 | 0 | 7 |
| **Daily Resource Limits** | | | 35 | 35 | 35 | 35 |

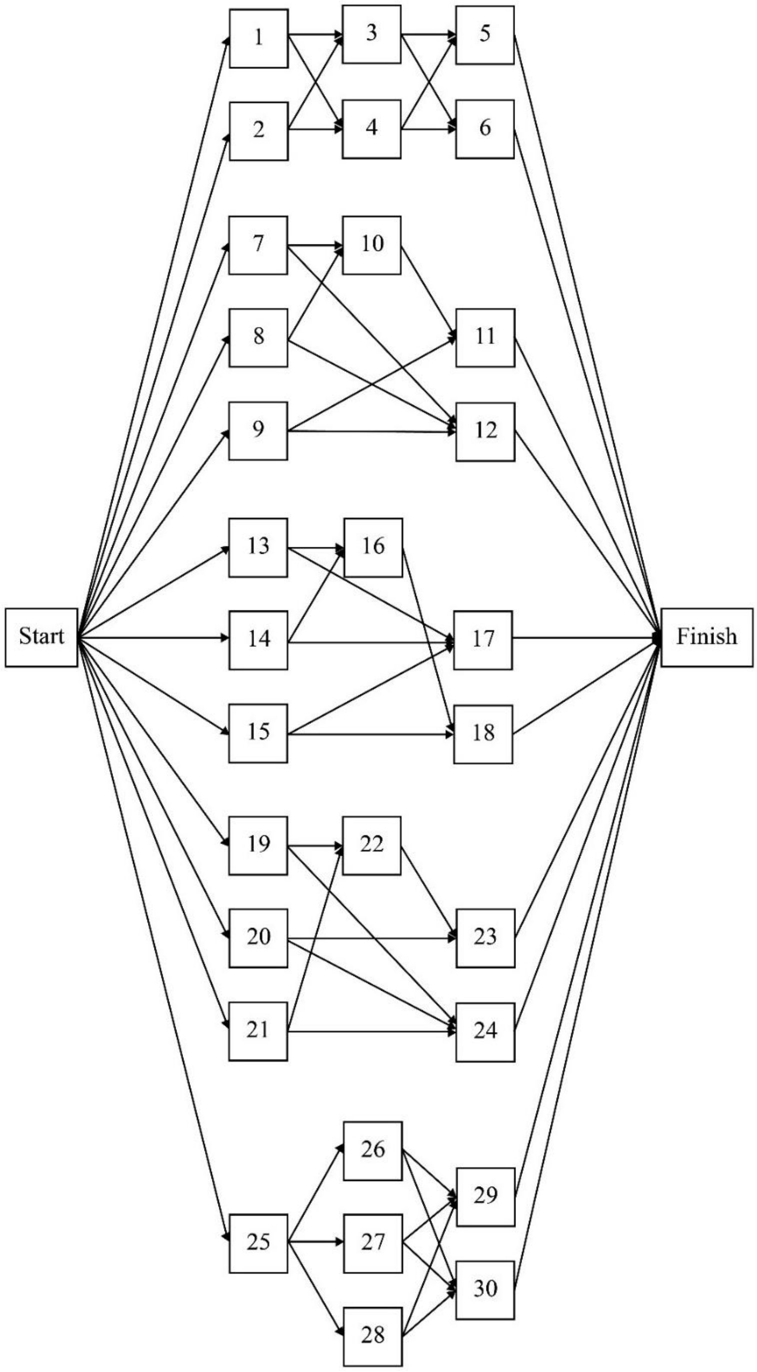


Figure 4. Project precedence network

### 4.2. Parameter Selection

In order to provide a fair comparison, the number of function evaluations of each algorithm must be the same. In each iteration, the Hybrid-Chromosome NSGA-II and MOPSO do one function evaluation, whereas the MOSOS does four. Therefore, the number of iterations of the Hybrid-Chromosome NSGA-II and MOPSO must be four times as many as that of the MOSOS. Same as the previous study, the value of $\lambda$ is assumed to be 30. The parameter selection of each algorithm is presented in Table 2.

Table 2. Parameter selection

| Algorithm | Parameters | Value |
|---|---|---|
| Hybrid-Chromosome NSGA-II | Population size | 100 |
| | Number of iterations | 1000 |
| | Crossover rate $CR$ | 0.9 |
| | Mutation rate $MR$ | 0.2 |
| | Mating pool size | 30 |
| MOPSO | Population size | 100 |
| | Number of iterations | 1000 |
| | $w$ | 0.5 |
| | $c_1$ | 2 |
| | $c_2$ | 2 |
| MOSOS | Population size | 100 |
| | Number of iterations | 250 |

### 4.3. Results and Comparison

The hypervolume calculation results are presented in Table 3 and 4, with the green highlight indicating the best hypervolume.

Table 3. Hypervolume calculation results of experiment 1

| Algorithm | Run | Number of Solutions | Hypervolume (%) | Average | Standard Deviation |
|---|---|---|---|---|---|
| Hybrid-Chromosome NSGA-II | 1 | 12 | 60.85 | 65.12 | 7.20 |
| | 2 | 12 | 59.52 | | |
| | 3 | 9 | 65.18 | | |
| | 4 | 11 | 62.61 | | |
| | 5 | 9 | 77.44 | | |
| MOPSO | 1 | 4 | 42.93 | 38.82 | 9.22 |
| | 2 | 7 | 37.91 | | |
| | 3 | 4 | 43.13 | | |
| | 4 | 8 | 23.32 | | |
| | 5 | 3 | 46.80 | | |
| MOSOS | 1 | 5 | 55.52 | 51.40 | 4.40 |
| | 2 | 4 | 56.70 | | |
| | 3 | 6 | 47.03 | | |
| | 4 | 3 | 48.31 | | |
| | 5 | 4 | 49.48 | | |

Table 4. Hypervolume calculation results of experiment 2

| Algorithm | Run | Number of Solutions | Hypervolume (%) | Average | Standard Deviation |
|---|---|---|---|---|---|
| Hybrid-Chromosome NSGA-II | 1 | 21 | 59.09 | 59.72 | 11.89 |
| | 2 | 15 | 52.30 | | |
| | 3 | 29 | 79.81 | | |
| | 4 | 31 | 57.87 | | |
| | 5 | 26 | 49.55 | | |
| MOPSO | 1 | 3 | 42.42 | 43.46 | 3.34 |

| | | | | | |
|---|---|---|---|---|---|
| | 2 | 5 | 44.14 | | |
| | 3 | 5 | 46.14 | | |
| | 4 | 7 | 46.39 | | |
| | 5 | 6 | 38.23 | | |
| MOSOS | 1 | 7 | 51.33 | 55.27 | 7.40 |
| | 2 | 10 | 58.00 | | |
| | 3 | 7 | 65.44 | | |
| | 4 | 3 | 55.92 | | |
| | 5 | 4 | 45.67 | | |

Table 3 and 4 show that the Hybrid-Chromosome NSGA-II algorithm outperforms both MOPSO and MOSOS in terms of the average value of the hypervolume. However, MOSOS and MOPSO show better consistency by obtaining the smallest value of standard deviation in both experiment 1 and experiment 2, respectively. Furthermore, it can be seen from Table 3 that the smallest hypervolume obtained by Hybrid-Chromosome NSGA-II (59.5232%) is still greater than the biggest hypervolume obtained by both MOPSO (46.8008%) and MOSOS (56.6968%), and thereby validating the superior performance of the Hybrid-Chromosome NSGA-II algorithm.

Based on Table 4, the smallest hypervolume obtained by Hybrid-Chromosome NSGA-II (52.3002%) only surpasses the biggest hypervolume obtained by MOPSO (46.388%). Therefore, further statistical test (Mann-Whitney Test) is needed to compare the hypervolume obtained by Hybrid-Chromosome NSGA-II and MOSOS in experiment 2, with $H_0$: the two algorithms perform equally and $H_1$: the two algorithms perform differently. The ranks of the hypervolume from all five runs, sorted from the smallest to the largest (where rank 1 is the smallest and rank 10 is the highest), are presented in Table 5.

Table 5. The ranks of the hypervolume

| | | | | | |
|---|---|---|---|---|---|
| Hybrid-Chromosome NSGA-II | 8 | 4 | 10 | 6 | 2 |
| MOSOS | 3 | 7 | 9 | 5 | 1 |

From Table 5, the value of $W_1$ and $W_2$ can be obtained by calculating the sum of the ranks: $W_1 = 8 + 4 + 10 + 6 + 2 = 30$, $W_2 = 3 + 7 + 9 + 5 + 1 = 25$. Then, the value of $U_1$ and $U_2$ can be calculated using these two equations below, where $n_1$ and $n_2$ are the number of participants of each group.

$$\begin{aligned} U_1 &= n_1 n_2 + \frac{n_1(n_1+1)}{2} - W_1 \\ &= 5 \times 5 + \frac{5(5+1)}{2} - 30 \\ &= 10 \end{aligned} \tag{12}$$

$$\begin{aligned} U_2 &= n_1 n_2 + \frac{n_2(n_2+1)}{2} - W_2 \\ &= 5 \times 5 + \frac{5(5+1)}{2} - 25 \\ &= 15 \end{aligned} \tag{13}$$

From $U_1$ and $U_2$, the smallest value of the two is selected: $U = 10$. Based on the reference table for Mann-Whitney Test, the $P$ value is 0.345 for $n = 5$ and $U = 10$. Therefore, it can be concluded that none of these two algorithms performs better statistically than the other ($H_0$ is not rejected) in significance level $\alpha = 0.05$, as the value of $P = 0.345$ is greater than 0.05. Furthermore, the computational time of each algorithm from both experiments are also presented in Table 6 and 7.

Table 6. Computational time of experiment 1

| Algorithm | Run | Number of Solutions | Time (Minutes) | Average | Standard Deviation |
|---|---|---|---|---|---|
| Hybrid-Chromosome NSGA-II | 1 | 12 | 32.83 | 33.46 | 0.78 |
| | 2 | 12 | 32.72 | | |
| | 3 | 9 | 33.23 | | |
| | 4 | 11 | 34.53 | | |
| | 5 | 9 | 33.97 | | |
| MOPSO | 1 | 4 | 47.06 | 48.03 | 1.33 |
| | 2 | 7 | 47.99 | | |
| | 3 | 4 | 46.70 | | |
| | 4 | 8 | 48.30 | | |
| | 5 | 3 | 50.11 | | |
| MOSOS | 1 | 5 | 52.19 | 51.56 | 0.39 |
| | 2 | 4 | 51.63 | | |
| | 3 | 6 | 51.17 | | |
| | 4 | 3 | 51.36 | | |
| | 5 | 4 | 51.47 | | |

Table 7. Computational time of experiment 2

| Algorithm | Run | Number of Solutions | Time (Minutes) | Average | Standard Deviation |
|---|---|---|---|---|---|
| Hybrid-Chromosome NSGA-II | 1 | 21 | 34.20 | 34.70 | 2.86 |
| | 2 | 15 | 32.93 | | |
| | 3 | 29 | 34.96 | | |
| | 4 | 31 | 39.40 | | |
| | 5 | 26 | 32.00 | | |
| MOPSO | 1 | 3 | 51.83 | 52.06 | 1.07 |
| | 2 | 5 | 50.84 | | |
| | 3 | 5 | 53.05 | | |
| | 4 | 7 | 53.25 | | |
| | 5 | 6 | 51.27 | | |
| MOSOS | 1 | 7 | 54.01 | 53.49 | 0.34 |
| | 2 | 10 | 53.11 | | |
| | 3 | 7 | 53.30 | | |
| | 4 | 3 | 53.55 | | |
| | 5 | 4 | 53.49 | | |

It can be seen from Table 6 and 7 that the Hybrid-Chromosome NSGA-II algorithm shows faster computational time compared to both MOPSO and MOSOS, with an average computational time of 33.4573 and 34.69762 minutes for experiment 1 and experiment 2, respectively. In terms of consistency, MOSOS shows better performance compared to both Hybrid-Chromosome NSGA-II and MOPSO.

In order to give a more detailed explanation about the trade-off among the objectives, the solutions from the best hypervolume are presented in Table 8 for experiment 1 and Table 9 for experiment 2. The green highlight indicates where the minimum objective values occur.

Table 8. Solutions from the best hypervolume of experiment 1

| **Algorithm** | **Solution** | **Project Duration** | $RRH$ | $RI$ |
|---|---|---|---|---|
| Hybrid-Chromosome NSGA-II | 1 | 13 | 7.25 | 6.67 |
| | 2 | 14 | 6.75 | 16.96 |
| | 3 | 14 | 7 | 16.51 |
| | 4 | 15 | 5.25 | 18.76 |
| | 5 | 15 | 5.5 | 18.44 |
| | 6 | 15 | 5.75 | 18.38 |
| | 7 | 15 | 6 | 18.37 |
| | 8 | 15 | 6.5 | 16.22 |
| | 9 | 15 | 6.75 | 15.90 |
| MOPSO | 1 | 14 | 7.5 | 12.84 |
| | 2 | 14 | 8 | 12.37 |
| | 3 | 14 | 8.75 | 11.42 |
| MOSOS | 1 | 13 | 14 | 10.02 |
| | 2 | 14 | 6.5 | 14.97 |
| | 3 | 14 | 7.5 | 14.03 |
| | 4 | 14 | 9.25 | 12.75 |

Table 9. Solutions from the best hypervolume of experiment 2

| **Algorithm** | **Solution** | **Project Duration** | $RID$ | $RI$ |
|---|---|---|---|---|
| Hybrid-Chromosome NSGA-II | 1 | 13 | 12 | 17.58 |
| | 2 | 13 | 13.25 | 13.90 |
| | 3 | 13 | 13.5 | 13.44 |
| | 4 | 13 | 14.75 | 13.22 |
| | 5 | 13 | 15.5 | 12.94 |
| | 6 | 13 | 15.75 | 9.73 |
| | 7 | 13 | 17.25 | 8.10 |
| | 8 | 13 | 17.5 | 8.074 |
| | 9 | 13 | 18 | 8.05 |
| | 10 | 13 | 18.25 | 8.02 |
| | 11 | 14 | 6.5 | 19.05 |
| | 12 | 14 | 6.75 | 15.22 |
| | 13 | 14 | 7 | 14.97 |
| | 14 | 14 | 7.25 | 12.92 |
| | 15 | 14 | 7.5 | 12.67 |
| | 16 | 14 | 7.75 | 12.42 |
| | 17 | 14 | 8 | 12.33 |
| | 18 | 14 | 8.5 | 12.16 |
| | 19 | 14 | 8.75 | 11.91 |
| | 20 | 14 | 9.25 | 11.38 |
| | 21 | 14 | 9.5 | 9.42 |

| | | | | |
|---|---|---|---|---|
| | 22 | 14 | 10 | 8.92 |
| | 23 | 14 | 10.25 | 8.85 |
| | 24 | 14 | 11 | 8.63 |
| | 25 | 14 | 11.25 | 8.16 |
| | 26 | 14 | 11.5 | 8.11 |
| | 27 | 14 | 12 | 7.94 |
| | 28 | 14 | 12.75 | 7.78 |
| | 29 | 14 | 14.25 | 7.75 |
| MOPSO | 1 | 13 | 20.75 | 18.40 |
| | 2 | 13 | 25.5 | 14.64 |
| | 3 | 13 | 30.5 | 12.03 |
| | 4 | 14 | 17.25 | 14.71 |
| | 5 | 14 | 17.5 | 14.02 |
| | 6 | 14 | 19.75 | 12.99 |
| | 7 | 15 | 16.25 | 29.81 |
| MOSOS | 1 | 13 | 19.5 | 10.93 |
| | 2 | 14 | 8.5 | 16.93 |
| | 3 | 14 | 14.5 | 11.26 |
| | 4 | 14 | 14.75 | 10.84 |
| | 5 | 14 | 15 | 10.73 |
| | 6 | 14 | 16.75 | 10.56 |
| | 7 | 14 | 19 | 8.95 |

Table 8 shows the solutions from the best hypervolume of experiment 1. It can be seen that the Hybrid-Chromosome NSGA-II algorithm is able to produce the minimum value of all the objectives, which are 13 days for the project duration, 5.25 for the $RRH$, and 6.67 for the $RI$. Moreover, both the minimum project duration and $RI$ come from the same solution (solution 1 obtained by Hybrid-Chromosome NSGA-II). Additionally, MOSOS is also able to find a solution with a minimum project duration of 13 days. Table 8 clearly shows the superiority of Hybrid-Chromosome NSGA-II compared to the other two benchmark algorithms.

Table 9 shows the solutions from the best hypervolume of experiment 2. Table 9 also clearly shows that the Hybrid-Chromosome NSGA-II algorithm manages to find the minimum value of all the objectives, which are 13 days for the project duration, 6.5 for the $RID$, and 7.75 for the $RI$. Additionally, the minimum project duration of 13 days is also found by MOPSO and MOSOS. It can also be seen that the number of non-dominated solutions found by Hybrid-Chromosome NSGA-II is also greater than both MOPSO and MOSOS. Therefore, it can be concluded that Hybrid-Chromosome NSGA-II also demonstrates better spread of the solutions based on Table 9.

The plots of all the solutions are illustrated in Figure 5 for experiment 1 and Figure 6 for experiment 2. Figure 5 and 6 show that the solutions produced by the Hybrid-Chromosome NSGA-II algorithm have better spread and diversity compared to the ones obtained by the other two benchmark algorithms, MOPSO and MOSOS.

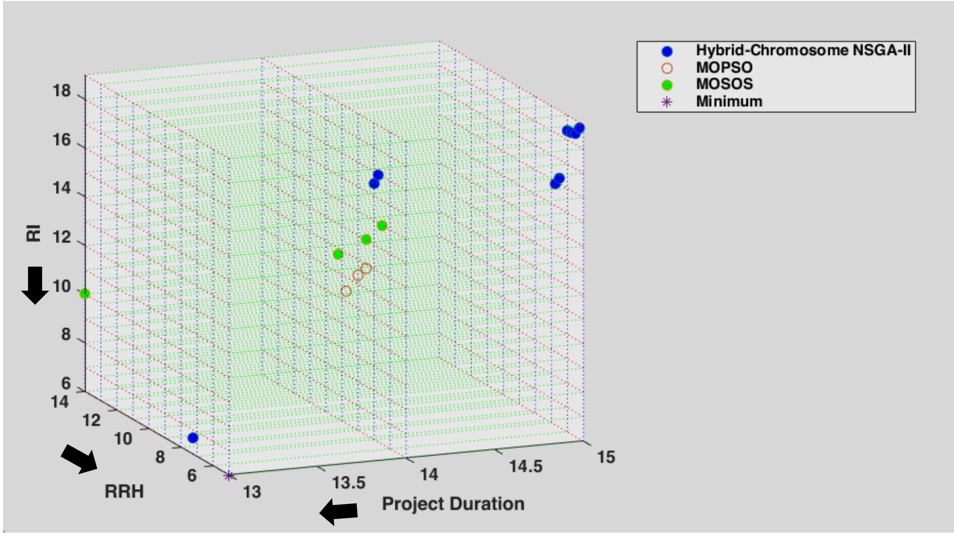


Figure 5. Plotted solutions of the best hypervolume of experiment 1

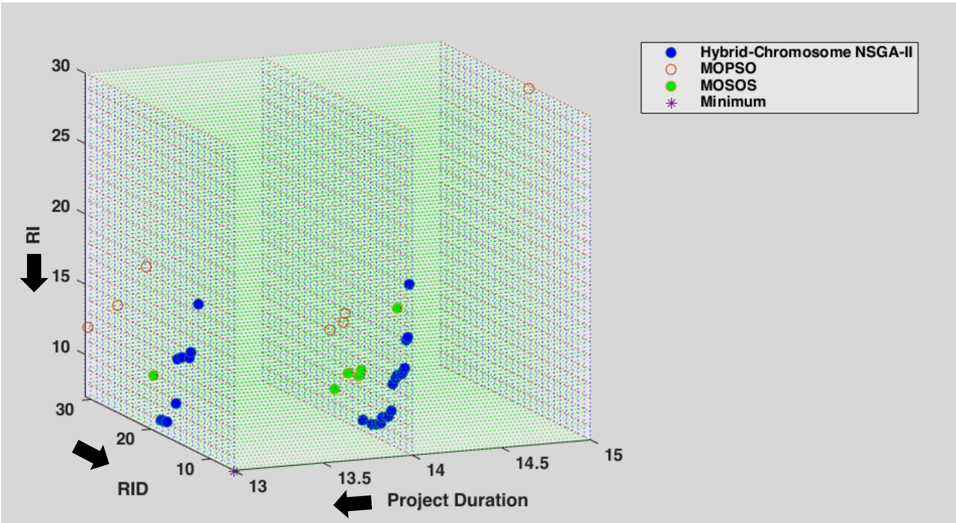


Figure 6. Plotted solutions of the best hypervolume of experiment 2

In order to illustrate the solutions better, the Gantt-Chart of solution 1 obtained by the Hybrid-Chromosome NSGA-II algorithm from experiment 1 is illustrated in Figure 7, while the distribution of R1 until R4 are illustrated in Figure 8 until Figure 11 with the dashed line representing the daily resource limits. This solution is selected as an example since it produces two minimum values out of the three objectives.

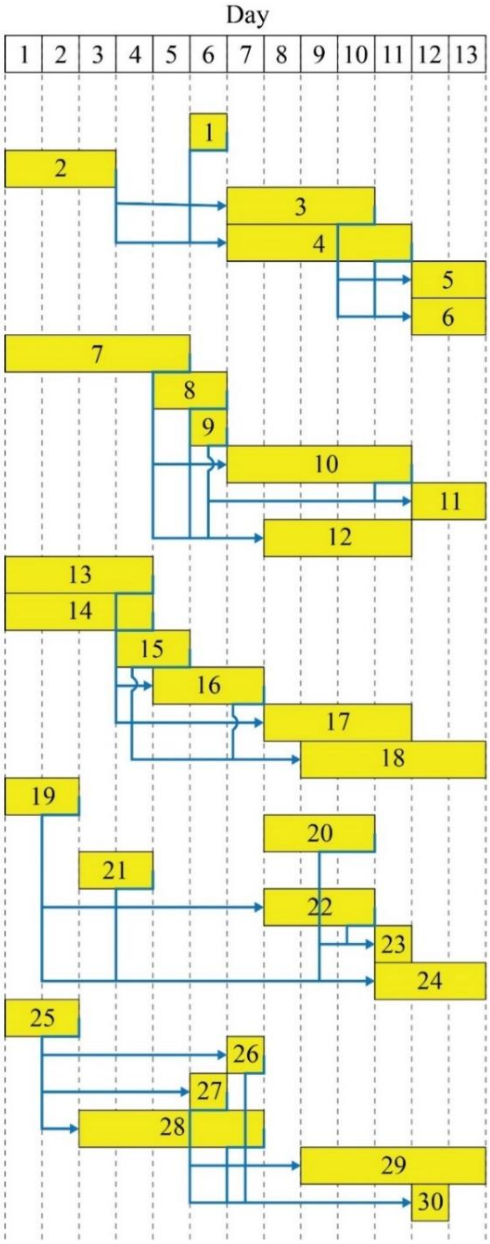


Figure 7. The gantt-chart of solution 1 obtained by hybrid-chromosome NSGA-II from experiment 1

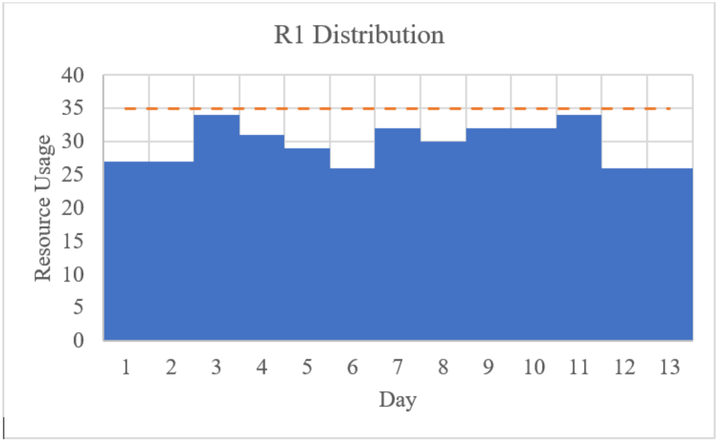


Figure 8. R1 distribution of solution 1 obtained by hybrid-chromosome NSGA-II from experiment 1

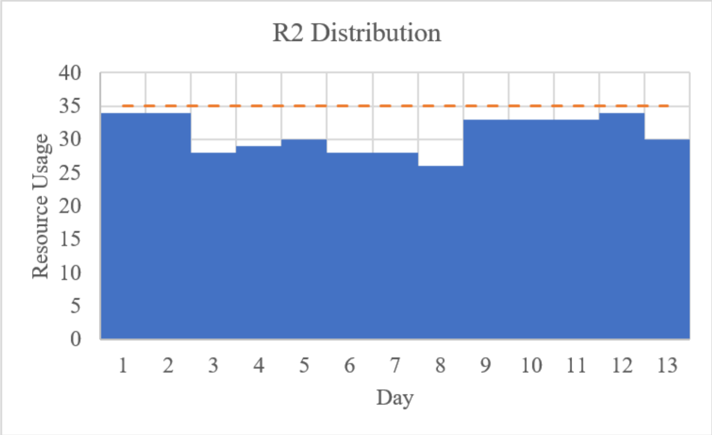


Figure 9. R2 distribution of solution 1 obtained by hybrid-chromosome NSGA-II from experiment 1

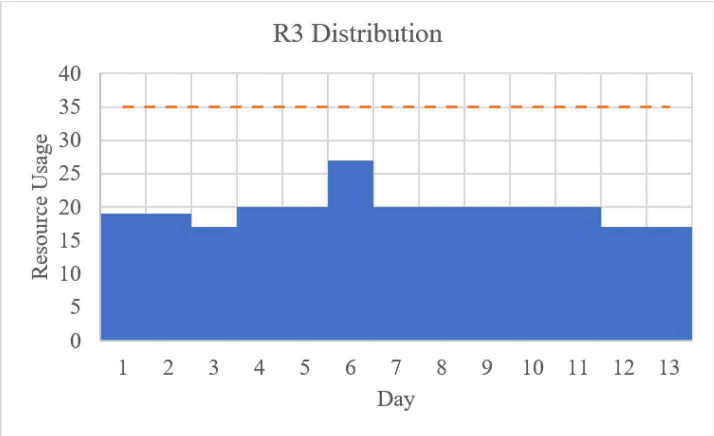


Figure 10. R3 distribution of solution 1 obtained by hybrid-chromosome NSGA-II from experiment 1

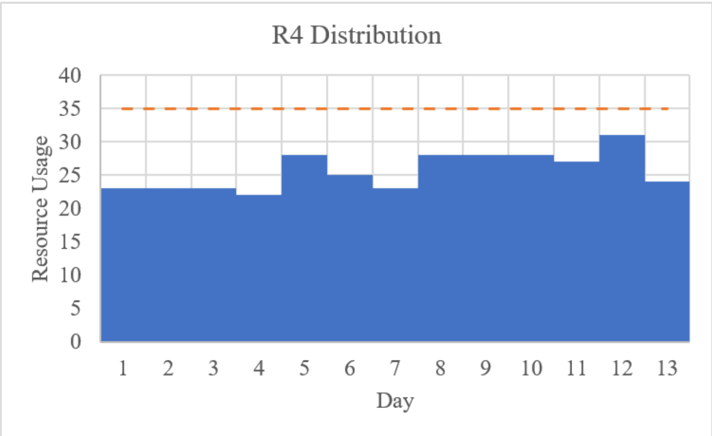


Figure 11. R4 distribution of solution 1 obtained by hybrid-chromosome NSGA-II from experiment 1

From the Gantt-Chart shown in Figure 7 and resource distribution shown in Figure 8 until Figure 11, it is proven that the proposed model and algorithm are able to solve the MR-AL-MP scheduling problem by producing alternative project schedules that satisfy both the precedence and resource availability constraints.

## 5. CONCLUSIONS

This study focuses in creating a unified framework to handle the MR-AL-MP scheduling problem by producing a set of alternative project schedules from which a project manager can select the most preferred schedule based on practical management needs. This study formulates the MR-AL-MP scheduling model to find the trade-off between minimum project duration and maximum resource utilization, while complying with all precedence and resource availability constraints. To be more practical, the proposed model is framed in a multi-project environment that involves multiple types of resources. The considered optimization objective functions include project duration, $RRH$ or $RID$, and $RI$.

The MR-AL-MP scheduling model is solved by the Hybrid-Chromosome NSGA-II algorithm. For the purpose of validation, two case studies are used to validate and evaluate the performance of the proposed model and algorithm, while also providing a comparison with two other benchmark algorithms (MOPSO and MOSOS). From the results obtained, it is proven that the proposed model, along with the Hybrid-Chromosome NSGA-II, is able to provide a set of feasible solutions for the MR-AL-MP scheduling problem. Furthermore, the Hybrid-Chromosome NSGA-II outperforms the MOPSO and MOSOS in terms of solution quality, spread, and diversity.

For future studies, the following suggestions are proposed:

a. This study attempts to handle scheduling problems in a multi-project environment, meaning that the resources are shared among the projects. Therefore, resource mobilization is an important matter to be discussed as it involves some critical resources, including money. It will certainly be beneficial if further studies can consider resource mobilization, such as transportation cost and time, as one of the determining factors.
b. The proposed model currently is only able to handle scheduling problems with single-mode activities. The use of multi-mode can provide more flexibility as each activity will have different alternatives on how it is executed, especially in terms of the duration and the number of resources involved. Future work can be devoted to handle multi-mode activities.
c. The proposed model employs the $RRH$ or $RID$ as one of the objectives, and thus only considers the utilization of renewable resources. However, in real life situation, the use of non-renewable resources, such as money and material, in construction projects is inevitable. Therefore, it will be beneficial if further studies can extend the proposed model to consider non-renewable resources.

## 6. REFERENCES

Alcaraz, J., & Maroto, C. (2001). “A Robust Genetic Algorithm for Resource Allocation in Project Scheduling”. *Annals of Operations Research*, *102*(1), 83-109.

Chan, W. T., Chua, D. K., & Kannan, G. (1996). “Construction Resource Scheduling with Genetic Algorithms”. *Journal of Construction Engineering and Management*, *122*(2), 125-132.

Chen, P. H., & Shahandashti, S. M. (2009). “Hybrid of Genetic Algorithm and Simulated Annealing for Multiple Project Scheduling with Multiple Resource Constraints”. *Automation in Construction*, *18*(4), 434-443.

Coello, C. C., & Lechuga, M. S. (2002). “Mopso: a Proposal for Multiple Objective Particle Swarm Optimization”. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)* (Vol. 2, pp. 1051-1056). IEEE.

Cui, Y., Geng, Z., Zhu, Q., & Han, Y. (2017). "Multi-objective Optimization Methods and Application in Energy Saving". *Energy*, *125*, 681-704.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). "A Fast and Elitist Multiobjective Genetic Algorithm NSGA-II". *IEEE Transactions on Evolutionary Computation*, *6*(2), 182-197.

El-Rayes, K., & Jun, D. H. (2009). "Optimizing Resource Leveling in Construction Projects". *Journal of Construction Engineering and Management*, *135*(11), 1172-1180.

Guo, Y., Li, N., & Ye, T. (2009). "Multiple Resources Leveling in Multiple Projects Scheduling Problem Using Particle Swarm Optimization". In *2009 Fifth International Conference on Natural Computation* (Vol. 3, pp. 260-264). IEEE.

Hegazy, T. (1999). "Optimization of Resource Allocation and Leveling Using Genetic Algorithms". *Journal of Construction Engineering and Management*, *125*(3), 167-175.

Heon Jun, D., & El-Rayes, K. (2011). "Multiobjective Optimization of Resource Leveling and Allocation during Construction Scheduling". *Journal of Construction Engineering and Management*, *137*(12), 1080-1088.

Khanzadi, M., Kaveh, A., Alipour, M., & Karimi Aghmiuni, H. (2016). "Application of CBO and CSS for Resource Allocation and Resource Leveling Problem". *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, *40*(1), 1-10.

Koulinas, G. K., & Anagnostopoulos, K. P. (2013). "A New Tabu Search-Based Hyper-Heuristic Algorithm for Solving Construction Leveling Problems with Limited Resource Availabilities". *Automation in Construction*, *31*, 169-175.

Lova, A., & Tormos, P. (2001). "Analysis of Scheduling Schemes and Heuristic Rules Performance in Resource-Constrained Multiproject Scheduling". *Annals of Operations Research*, *102*(1), 263-286.

Majazi Dalfard, V., & Ranjbar, V. (2012). "Multi-Projects Scheduling with Resource Constraints & Priority Rules by the Use of Simulated Annealing Algorithm". *Tehnički Vjesnik*, *19*(3), 493-499.

Ngatchou, P., Zarei, A., & El-Sharkawi, A. (2005). "Pareto Multi Objective Optimization". In *Proceedings of The 13th International Conference on, Intelligent Systems Application to Power Systems* (pp. 84-91). IEEE.

Ponz-Tienda, J. L., Salcedo-Bernal, A., Pellicer, E., & Benlloch-Marco, J. (2017). "Improved Adaptive Harmony Search Algorithm for the Resource Leveling Problem with Minimal Lags". *Automation in Construction*, *77*, 82-92.

Popescu, C. M., & Charoenngam, C. (1995). *Project Planning, Scheduling, and Control in Construction: an Encyclopedia of Terms and Applications*. John Wiley & Sons.

Son, J., & Mattila, K. G. (2004). "Binary Resource Leveling Model: Activity Splitting Allowed". *Journal of Construction Engineering and Management*, *130*(6), 887-894.

Sörensen, K., & Glover, F. (2013). "Metaheuristics". *Encyclopedia of Operations Research and Management Science*, *62*, 960-970.

Tran, D. H., Cheng, M. Y., & Prayogo, D. (2016). "A Novel Multiple Objective Symbiotic Organisms Search (MOSOS) for Time–Cost–Labor Utilization Tradeoff Problem". *Knowledge-Based Systems*, *94*, 132-145.

Wu, Z., Zhang, L., Wang, Y., & Wang, K. (2008). "Optimization for Multi-Resource Allocation and Leveling based on a Self-Adaptive Ant Colony Algorithm". In *2008 International Conference on Computational Intelligence and Security* (Vol. 1, pp. 47-51). IEEE.

Zhang, H., Li, H., & Tam, C. M. (2006). "Permutation-Based Particle Swarm Optimization for Resource-Constrained Project Scheduling". *Journal of Computing in Civil Engineering*, *20*(2), 141-149.

Zhang, H., Li, X., Li, H., & Huang, F. (2005). "Particle Swarm Optimization-Based Schemes for Resource-Constrained Project Scheduling". *Automation in Construction*, *14*(3), 393-404.

Zhao, S. L., Liu, Y., Zhao, H. M., & Zhou, R. L. (2006). "GA-Based Resource Leveling Optimization for Construction Project". In *2006 International Conference on Machine Learning and Cybernetics* (pp. 2363-2367). IEEE.