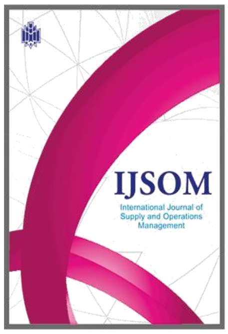
International Journal of Supply and Operations Management

IJSOM



August 2019, Volume 6, Issue 3, pp. 213-230
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com

A Multi-Objectives Weighting Genetic Algorithm for Scheduling Resource-Constraint Project Problem in the Presence of Resource Uncertainty

Aidin Delgoshaei [a,*], Aisa Khoshniat Aram[a], Vahid Z Mantegh[a], Sepehr Hanjani[a], Amir Hossein Nasiri[a] and Fatemeh Shirmohamadi[a]

[a] *Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, University of Putra, Serdang, Malaysia*

Abstract
Scarce resources may cause delay in completion of a project. In this research, a multi-objective decision making model is developed for scheduling multi-mode resource constraint scheduling problem in the presence of uncertain resources. The objectives are profit, execution cost and completion time. To develop this idea, a multi-objective non-linear mixed integer programming model is developed where resource availability is not deterministic and expressed by triangular probability function. Then, a multi-objective weighting genetic algorithm is proposed (MOWGA) which is flexible enough to be used in real projects. To verify the performance of the proposed method, a number of experiments are solved and results are analyzed. The outcomes indicated that while resource uncertainty increases, higher complexity in schedules is observed. It is also found that optimizing one objective function does not necessarily resulted in optimizing the others. The MOWGA is then successfully applied for a project with real data.

Keywords: Project Planning; Multi-Mode Scheduling; Multi-Objective Weighting Genetic Algorithm.

1. Introduction

Lack of sufficient resources or having low quality resources is considered a serious risk for executing project activities on time. Each year there are thousands of projects which fail or are stopped due to lack or insufficient resources. It is estimated that a failed project can cause wasting money of organization in a way that for each 1 billion USD of investment in a fail project, 135 million USD is waved. Therefore, having a plan to predict, schedule and monitor the resources during project implementation is vital. Project scheduling is one of the areas of project management that is very important and can play a key role in preventing project failures due to lack of resources. Results of surveying the literature review shows that a large number of researches are carried out where their focus were on resources scheduling. This shows the importance of resource scheduling in project management. There are many reasons one resource cannot be predicted exactly in advance.

1- The suppliers of the resource may not be able to deliver all the requirements as scheduled.
2- Some parts of the resource may breakdown during the transportation.
3- Some materials may be breakdown due to high humidity or temperature in the warehouses.

In every country, there are many tough rules for making extreme fines for delaying in delivering project. Therefore, developing a new model to take quick response to changes of the resources during the life cycle of the project is necessary. This will reduce the harms of resource uncertainty. An initial investigation on the current methods on scheduling projects in terms of resource availability shows that the issue of uncertainty in resource availability is less developed. In this research a useful method is offered for scheduling resource constraint projects while resource availabilities are uncertain during -

Corresponding author email address: delgoshaei.aidin@gmail.com

project calendar. Thus, this study focuses on scheduling projects while constraint uncertain resources dominate the project and all activities are affected (and total project as a consequent) by such resources uncertainty.

The findings of this research can help project engineers during scheduling projects. After explaining the problem statement of the project, the question of the research is made as follows: “How do we reduce the harms of resource uncertainty during scheduling process of multi-mode resource constraint scheduling problems?” To verify findings, the proposed method in virtual systems must be examined using simulation. Then in the next step, the proposed method shall be validated in real project to verify the results. Malaysia is chosen for project environment.

2. Literature Review

This section reviews the models and solving algorithms in project scheduling problem. For this purpose, 51 papers have been cited and reviewed in detail. In each section novelties, advantages or lack of the methods will be explained in details. Then, solving algorithms that are normally heuristics and metaheuristics will be considered. Multi-mode resource constraint project scheduling problems (MRCPSP) are distinctive resource-constraint problems where each activity can be carried out via different modes (regarding technologies or materials). As a consequence, the execution period (activity duration), resource requirement level and even the cash flow may vary from a mode to another. The MPRCPSP problem was initially developed for minimizing the project makespan and was proved to be a NP-hard problem.

2.1. Makespan minimization

Perhaps reducing the completion time of a project is a wish for both contractor and owner of a project. Therefore, many researchers did their best to minimize the makespan of projects by considering various conditions. Afterwards, Ke and Liu (2010) used the same logic of fuzzy and genetic algorithm to minimize makespan. Vanhoucke and Debels (2008) focused on impacts of variable activity durations under a fixed work content, possibility of allowing activity pre-emption and use of fast tracking in decreasing project makespan. Van Peteghem and Vanhoucke (2010) investigated the impact of pre-emptive resources in the process of minimizing completion time in scheduling MRCPSPs. Delgoshaei et al. (2015) dealt with minimizing completion time of a RCPSP using empty spaces of resources through calendar of projects. For this purpose, they used greedy and genetic algorithms. Chtourou and Haouari (2008) proposed an algorithm which worked based on increasing float time. In the first part of their model a threshold value is used to minimize completion time and in the next part the solution robustness is increased using mathematical indicators. Sprecher (2012) focused on minimizing makespan of a resource constraint project using binary programming. Browning and Yassine (2010) focused on the problem allocate resources to minimize the average delay per project or the time to complete. W. Chen et al. (2010) proposed a hybrid ant colony optimization (ACO) and a scatter search for solving RCPSP in real times. W.-x. Wang et al. (2014) addressed a multi-project scheduling in critical chain problem using a multi-objective optimization mode where overall duration, financing costs and whole robustness are considered objectives. They solved their model using GA. Naber and Kolisch (2014) developed 4 mathematical models. Their models that used mixed integer programming are solving different conditions in resource scheduling. Afterwards, they proposed heuristics to solve the models. Vaez et al. (2018) addressed a new model for simultaneous scheduling and lot-sizing while earliness and tardiness penalties are taken into consideration. Kreter et al. (2016) developed 3 LP models for scheduling RCPSPs where breaks in activity calendars are taken into account. The models are then solved by CPLEX 16.0. Pérez et al. (2016) applied genetic algorithm for scheduling RCPSPs while minimizing completion time and average percent delay were the main two objectives. Kadri and Boctor (2018) proposed an efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times. Tao et al. (2018) focused on the problem Space-time repetitive project scheduling considering location and congestion. Davari and Demeulemeester (2019) dealt with the proactive and reactive resource-constrained project scheduling problem with stochastic activity durations.

2.2. Maximizing Profit of the project

Earning financial benefits from a project is among the most important goals of scheduling a project. In some researches, Net present value of project is considered a goal to be maximized. Others may consider payment progress of activities. Beyond positive cash flows that deal with the profit of a project, negative cash flows are also important to taken into consideration. Negative cash flows refer to expenses but they are necessary to earn positive cash flows. Salary, machinery, logistics, water, gas, electricity bills are among negative cash flows. In MRCPSP models, the negative and positive cash flows depend to the duration, technology and time of implementing activities. During the last decade, considering pre-emptive resource in scheduling problems have been more developed due to their impacts on making major delays through project lifecycle as well. Li et al. (2016) proposed a mixed integer programming method to choose projects in a portfolio selection problem in terms of divisibility and interdependency. Delgoshaei et al. (2014) used SA for maximizing NPV of the MRCPSP-DCF. Damay et al. (2007) applied linear programming algorithms for pre-emptive RCPSP studies while Ballestín et al. (2008) proposed heuristic for solving pre-emptive RCPSP. Seifi and Tavakkoli-Moghaddam (2008) evaluated four payment methods during maximizing NPV and minimizing holding cost of completed activities in a MRCPSP. Van Peteghem and Vanhoucke (2010) used GA to minimize makespan of MRCPSP while they considered pre-emptive resources which allow activity splitting through their research. Delgoshaei et al. (2017)

developed a heuristic method for maximizing NPV in MRCPSP models while the model considers all types of precedence relations and activity split is allowed.

2.3. Heuristics and Metaheuristics in Scheduling Problems

Resource constraint scheduling problems are complicated and cannot be optimized easily especially when the size of problem is large. RCPSP problem has more than one cause, and it is NP-hard and should use heuristics and metaheuristics. Beyond the size of a project, there are some other factors that can increase the complexity of a RCPSP which are mode of executing and number of resources and some managerial factors for implementing a project (Castejón-Limas et al., 2011). Genetic Algorithm (GA) is applied for project planning problems frequently. The problem project portfolio selection is investigated by Yu et al. (2012). They found that the proposed method can effectively select best project portfolios. The MRCPSPs are also complicated and regular method cannot solve them effectively. Therefore, researchers developed many heuristics and metaheuristics methods to overcome such complexity. Narayanan et al. (2009) dealt with the problems of fast reaction structure for maritime disasters. Laslo (2010) proposed a method for minimizing scheduling dependent expenses during scheduling activities. It is important to know that simulated annealing and genetic algorithm are used much more than other types of meta-heuristics in solving MRCPSP problems. Ke and Liu (2010) is also similar structure of fuzzy and GA to minimize total cost with completion time limits. Particle Swarm Optimization (PSO) that is a successful method is also used for solving MRCPSPS but less than other types of metaheuristics (J. Chen & Askin, 2009). L. Wang and Zheng (2018) addressed a knowledge-guided multi-objective fruit fly optimization algorithm for minimizing the makespan and the total cost in MRCPSPs simultaneously. As far as found in the literature, the idea of minimizing the cost of the activities in in MRCPSPs while resources availabilities are uncertain has not been taken into account yet. Therefore, this gap can be filled by developing a MRCPSP model while resources are considered stochastic. The model will develop in a way that each of the resources follows triangular distribution function. Then, impact of resource uncertainty on preventing resource over-allocation will also be evaluated.

3. Research Methodology

3.1. Flowchart of the Methodology

In this research, a forward serial scheduling is used while activities are scheduled in a forward mode in respect to their precedence as shown by Figure 1. Choosing each activity is based on its cash flow and also resource availability.

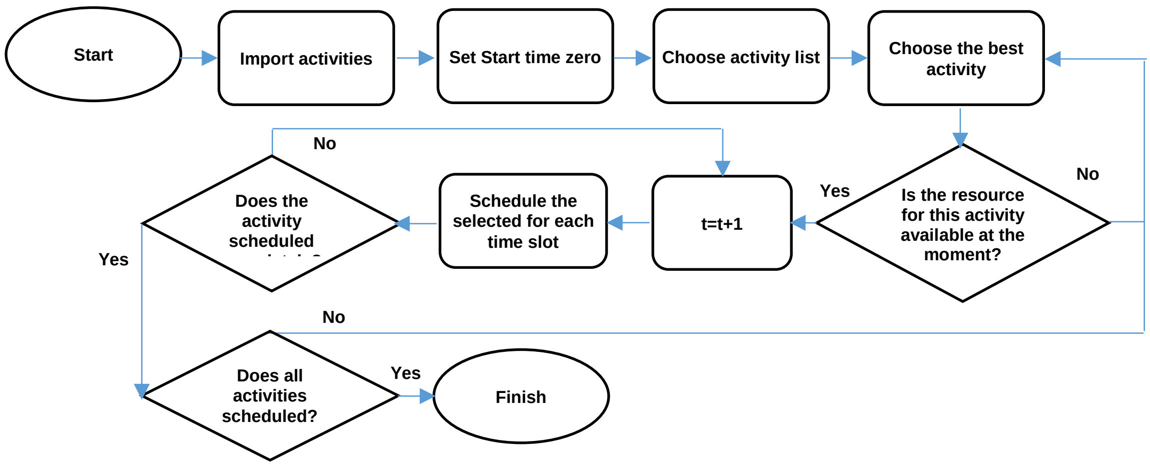


Figure 1. Flowchart of the proposed model

3.2. Develop Mathematical Model

As seen in the literature, many of the researches used multi-objective mixed integer programming method which seems suitable for this research as well. The model is MRCPSP. The objectives of this model are:

1- Maximizing Profit
2- Minimizing Total Cost (including fixed costs and execution costs of activities)
3- Minimizing Makespan
It should be noted that the resource availabilities are not deterministic and so expressed by a probability function.

As summary, the advantages of the model are:

- Maximizing Profit
- Minimizing scheduling costs including fixed and varied costs.
- Minimizing the completion time
- Uncertain resources
- Multi-mode execution modes
- All types of activity relations (Finish to Start, Start to Start, Start to Finish and Finish to Finish)

There are also a number of assumptions in this research:

1. The model is considered in multi-mode, so each of the activities can be executed in more than one way.
2. Activity split is allowed. This assumption is frequently used in practice and can be seen in all scheduling software like MSP and Primavera.
3. While a mode is selected for an activity, it is necessary to keep that mode until the activity is finished.
4. Forward programming is considered for transferring the activities.
5. All resources are renewable.
6. The range of resources is known and follows the triangular probability function.
7. Resource availability is not known in advance and should be predicted using triangular probability function.

3.3. Subscript

The indexes of the variables and parameters are defined as below:
i: number of activities
k: number of resource types
t: timeslots
m: number of execution modes

3.4. Input Parameters

The list of parameters and notations is as follows:

$D(i,m) =$ duration of Activity i while it performs in mode m
$FC(i,m) = Fixed\ Cost\ of\ Implementing$ activity i while it performs in mode m
Fixed cost can be considered secretary salary, water and electricity bills, copy of documents, etc.
$VC(i,m,t) = Variable\ Cost\ of\ Implementing$ activity i in each day while it performs in mode m
Variable cost is the cost of implementing one activity and since it depends on the number of days that an activity is being implemented and number of available resources on that day, it is not fixed.
$B_{(i,m,t)}$: Benefit of implementing the activity i using mode m during period t
$AR(k) =$ available level of resource type k

$$R_k \sim Triangular(\delta, \gamma) \quad \forall\ k = 1, .., K \tag{1}$$

$\mathrm{UR(i,k)} =$ usage amount of resource type k for activity i
Therefore the amount of remained resource in any time can be calculated by the following formula:

$$RC_k = AR_k - \sum_{i=1}^{n} UR_{(i,k)} \quad ; \forall\ (k \in K) \& (t \in TH) \tag{2}$$

$TH = time\ horizon\ of\ the\ project$
In this model, all relations between activities are considered which are finish to start, finish to finish, start to finish and start to start. Therefore ESj and EFj are calculated by equations 3-6.

$ES_j = ES_i + D_i + LAG_{ij}$ (If the precedence type between activitiy i and j is defined FS) (3)
$ES_j = ES_i + LAG_{ij}$ (If the precedence type between activitiy i and j is defined SS) (4)
$EF_j = EF_i + LAG_{\ ij} + D_j$ (If the precedence type between activitiy i and j is defined FF) (5)
$EF_j = ES_i + D_i + LAG_{ij} + D_j$ (If the precedence type between activitiy i and j is defined SF) (6)
$\alpha = interest\ rate$ (7)

3.5. Variables

It should be mentioned that in this model, 2 types of variables are used. The first variable (Y) is defined to enable model in choosing a mode for each of the activities. Afterward, the model uses variable X to schedule the activity.

$$Y(i,m) = \begin{cases} 1 & \text{if mode m is decided for activity } i \\ 0 & \text{otherwise} \end{cases}$$

$$X(i,m,t) = \begin{cases} 1 & \text{if activity } i \text{ implemented using mode } m \text{ during sub period } t \\ 0 & \text{otherwise} \end{cases}$$

3.6. Mathematical Model

As found in the literature review, mixed integer programming models are the most frequently used for scheduling MRCPSPs. The main idea of this mathematical model is inspired from Delgoshaei et al., 2016 by developing their idea in some new areas to fill some new gaps. After developing the model the new contribution of this model will be explained. The model is Binary programming (BP). It is assumed the model has n activities and m modes:

$$Max\ Z_1: \sum_{t=1}^{TH} \sum_{i=1}^{n} \sum_{m=1}^{M} X_{(i,m,t)} \, . B_{(i,m,t)} . e^{-\alpha} \tag{8}$$

$$Min\ Z_2: \sum_{t=1}^{TH} \sum_{i=1}^{n} \sum_{m=1}^{M} Y_{(i,m)} \, . FC_{(i,m)} + \sum_{t=1}^{TH} \sum_{i=1}^{n} \sum_{m=1}^{M} X_{(i,m,t)} \, . VC_{(i,m,t)} \tag{9}$$

$$Min\ Z_3: \sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} X_{(i,m,t)} \tag{10}$$

$S.T$:

$$ES_1 = 1 \ \ for\ the\ first\ activity \tag{11}$$

$$\sum_{t=1}^{TH} X_{i,m,t} = d_{i,m};\ \forall\ i = 1,..,n\ \&\ m = 1,..,M \tag{12}$$

$$\sum_{m=1}^{M} Y_{i,m} = 1;\ \forall\ i = 1,..,n \tag{13}$$

$$ES_i > \max_{t=1:TH}\{t.X_{(j,m,t)} | X_{(j,m,t)} = 1\};\ if\ j\ is\ predecessor\ for\ i \tag{14}$$

$$ES_i = \min_{t=1:TH}(\{t.(X_{(i,m,t)} - X_{(i,m,t-1)}) | X_{(i,m,t-1)} = 0\})\quad \forall\ i = 1,\dots,n \tag{15}$$

$$ES_j \geq ES_i + D_i + LAG_{ij};\ \ if\ the\ relation\ between\ i\ and\ j\ is\ FS \tag{16}$$

$$ES_j \geq ES_i + LAG_{ij};\ \ if\ the\ relation\ between\ i\ and\ j\ is\ SS \tag{17}$$

$$EF_j \geq EF_i + LAG_{ij} + D_j;\ if\ the\ relation\ between\ i\ and\ j\ is\ FF \tag{18}$$

$$EF_j \geq ES_i + D_i + LAG_{ij} + D_j;\ \ if\ the\ relation\ between\ i\ and\ j\ is\ SF \tag{19}$$

$$\max_{t=1:TH}\{t.X_{(n,m,t)} | X_{(n,m,t)} = 1\} \leq TH\ \ \text{for the last activity} \tag{20}$$

$$\textstyle\sum_{i=1}^{n}\sum_{m=1}^{M} UR_{i,k} \leq AR_k \quad \forall\ t = 1,..,T\ \&\ k = 1,..,K \tag{21}$$

$$R_k \sim Triangular(\delta,\gamma)\quad \forall\ k = 1,..,K \tag{22}$$

$$X_{i,m,t}\ \ is\ binary \tag{23}$$

$$Y_{i,m}\ \ is\ binary \tag{24}$$

3.7. Explaining the proposed Mathematical model

In this section, the proposed model will be explained in details. The objective function of the model is divided into three different objective functions. The first part represents maximizing profit of executing activities. For this part maximizing the profit using the net present value is $Max\ Z_1: \sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} X_{(i,m,t)}\,.\,B_{(i,m,t)}.\,e^{-\alpha}$ (eq. 8). In this formula, TH, N and M are upper limit (the largest value) of t, i and m respectively. B is benefit of executing an activity. Z is a symbol to show the objective function number. This formula is a well-known engineering economy formula for calculating NPV and means if an activity is scheduled and executed sooner, more profit will be gained. Note that alpha is interest rate and is a rate more than bank interest. The second term in objective function shows minimizing the total project cost. In this model, the cost is defined as fixed and varied cost. The fixed cost refers to those costs that are not dependent on the activity duration or activity mode and must be paid in a project such as electricity bill, water bill and office expenses, etc. For this section, we defined a variable which can take value 1 (independent to the activity duration) if the activity is decided to use one mode and else it will take 0. Thus, this term will be used to calculate the project fixed costs:

$$\textstyle\sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} Y_{(i,m)}\,.\,FC_{(i,m)} \tag{25}$$

Varied costs in contrast refer to those costs that completely depend on activity duration or activity modes. For example, the cost of renewable resources (like human resources or machines) completely depends on the activity duration and if the activity lasts for more days, more money must be paid for human resources that are working on that activity. For these activities, we used $X_{(i,m,t)}$ which is a variable that depends on the mode and number of the days the activity lasts. So the terms below can successfully calculate the varied cost of the activity depending on the number of the days an activity takes to be completed:

$$\sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} X_{(i,m,t)}\,.\,VC_{(i,m,t)} \tag{26}$$

Finally, the last part of objective function is set to minimize the execution time of activities or completion time of the project. The first constraint of the model shows that the first activity in the activity list must be started in day 1.

$$ES_1 = 1 \tag{27}$$

The 2nd constraint is used to guarantee that the activity will not be scheduled more than its predefined duration.

$$\sum_{t=1}^{TH} X_{i,m,t} = d_{i,m} \tag{28}$$

The 3rd constraint is to show that only one execution mode must be chosen for each activity:

$$\sum_{m=1}^{M} Y_{i,m} = 1 \tag{29}$$

The 4th constraint indicates that none of the activities can be started unless all its predecessors are scheduled.

$$ES_i > \max_{t=1:TH} \{t.X_{(j,m,t)} | X_{(j,m,t)} = 1\};\ if\ j\ is\ predecessor\ for\ i \tag{30}$$

The 5th constraint is to show the beginning day of each activity in the calendar.

$$ES_i = \min_{t=1:TH} \left(\{t.\left(X_{(i,m,t)} - X_{(i,m,t-1)}\right) | X_{(i,m,t-1)} = 0\}\right) \quad \forall\ i = 1, \dots, n \tag{31}$$

The constraint number 6, 7, 8 and 9 are for showing the finish to start, start to start, finish to finish and start to finish relations between activities. It should be mentioned that these formulas are general formulas that can be found in any project planning book. The 10th constraint is to show that the maximum day of the last activity should not exceed more than what is expressed in the contract of the project (Time Horizon). Note that this limitation can be ignored for those projects that are not pre-defined completion day by the owner of the project.

$$\max_{t=1:TH} \{t.X_{(n,m,t)} | X_{(n,m,t)} = 1\} \leq TH \tag{32}$$

The 11th limitation shows that in each day the number of allocated resources for the activities should not exceed more than the available resources levels. Note that each of the resource availabilities are not known in advance and must be predicted by the normal probability function.

$$\sum_{i=1}^{n} \sum_{m=1}^{M} UR_{i,k} \leq AR_k \tag{33}$$

Equation 19 shows the resources availability using triangular function. The formula is represented below:

$$R(k) = \begin{cases} Optimistic; \quad If\ P > Gamma \\ Most\ Probable; If\ Delta < P \leq Gamma \\ Pestimistic; if\ P \leq Delta \end{cases} \tag{34}$$

4. A Multi-Mode Genetic Algorithm for Solving the Developed Model

The MRCPSPs are complicated problems and as showed in the literature almost in all cases the scientists used heuristics and metaheuristics to solve them. Therefore it seems logical to follow their way. Each of the heuristics has excellent abilities to be used in solving non-linear mathematical models (as what developed in this research). Some of them have the ability to memorize the results through the searching process as Tabu search and artificial neural networks. Despite that, some others have the opportunity to escape local optimum points like Simulated Annealing. Genetic algorithm is among the most popular searching method which can guarantee improving the direction of the searching process in a promoting manner. GA uses the cross over and mutation operators for this purpose which will be explained later. In this research, GA chooses for solving the developed mathematical model for the following reasons:

1- There are many similar models that are solved by GA in the literature so GA is trustable.

2- GA uses a series of chromosomes and genes which perfectly match the activities to be scheduled in the Gantt chart.

The proposed flowchart in figure 2 will now be updated by adding MOWGA method.

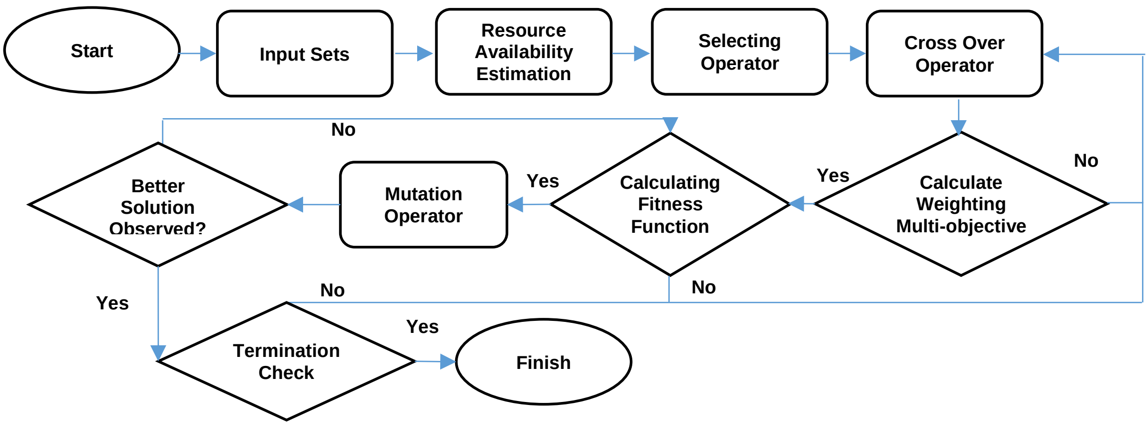


Figure 2. Flow chart of the proposed MOWGA

Step1: This method works based on principles of classic genetic algorithm. Therefore, in the first step, the number of inputs must be set by decision maker which are:

- Number of generations
- Number of Populations
- Mutation Rate

Step 2: There are also number of inputs that shows the example parameters. These parameters are:

- Number of Activities
- Number of activity modes
- Number of resources
- Gamma
- Delta
- Duration of activities
- Resource usage of activities
- Relations between activities
- LAGs between activities

Step 3: the algorithm will estimate the resources availability using triangular function. The formula is represented below:

$$R(k) = \begin{cases} Optimistic; \quad If\ P > Gamma \\ Most\ Probable; If\ Delta < P \leq Gamma \\ Pestimistic; if\ P \leq Delta \end{cases} \qquad (35)$$

Step 4: the algorithm will use selecting operator to choose best algorithm. The mechanism of the algorithm is designed in a way that the activities with more benefit and less cost have priority to be scheduled.

Step 5: While an activity is selected, the cross over operator will be used to find the best date to schedule it. In this section, the activity precedence (FS, SS, FF and SF) will be considered.

Step 6: the fitness function will be employed to check if the created solution is good enough to be considered a suitable solutions for the next generations.

Step 7: Even if the new solution has not improved the best observed objective value, there is still a little chance that can be given to the solution for next generations. This idea which can be expressed by mutation operator is necessary to escape the local optimum solutions in future.

Step 8: At the end of each level, the algorithm will check the stopping criteria (Generation number, Population size and Run time).

4.1. Multi-objective Scheduling Genetic Algorithm

Since the developed mathematical model in this research is multi-objective, the GA should work based on Multi-objective scales. For this purpose, the GA elements are developed to be fitted will MCDM solving methods. In this section, a general Multi-objective decision making model will be considered. Suppose a multi-objective function is shown below:

$$Max\ Z_1: aX + b \qquad (36)$$

$$Max\ Z_2: cY + d \qquad (37)$$

$$Min\ Z_2: eP + f \qquad (38)$$

$$s.t:$$

$$\sum_{i=1}^{n} A_i X_i < B \qquad (39)$$

$$\sum_{i=1}^{n} C_i Y_i < D \qquad (40)$$

$$\sum_{i=1}^{n} EP_i < F \qquad (41)$$

Where X, Y and P, are variables and a, b, c, d, e and f are parameters. Then the objective function in weighting method will be:

$$Max: W_1(aX+b)+W_2(cY+d)+W_3(eP+f) \quad (42)$$

W_1, W_2 *and* W_3 are the weights of the objective functions which will be chosen according to the importance of each of the objective functions. While the importance of the objectives is equal we can assume $W_1 = W_2 = W_3 = 0.33$.

4.2. Number of Generations

Number of generations is the first and most important factor in the genetic algorithm as it represents number of times that series of populations promotes to achieve the optimum or near optimum solutions. The number of generations is different from problem to problem. Moreover, the input values can play a key role in choosing the number of generations. To choose the best value for generation number, a set of examples will be solved using 3 different generation numbers (small, medium and large numbers) and the best result will be chosen accordingly (Appendix 1). Therefore, the generation number for small, medium and large size problem is set 10, 20 and 30 respectively. Any value larger than the mentioned values results in wasting time with no added multi-objective function.

4.3. Population Size

Population size is an important factor in achieving or approaching the optimum solutions. Although choosing large population size helps the algorithm find best solution, it will increase the computation time. Therefore, reasonable values should be considered for the population size. For this purpose, and as used by many researchers in the literature, we considered the following values for the population size. In Appendix 2 for each of the small scale problems 3 different population sizes are considered which are 10, 20 and 50 respectively. Afterwards, a similar problem is solved. The results showed that the best value is obtained while population size is considered 10. Therefore, for small scale problems the population size will be considered 10. Similar logic is used for medium and large scale problems. Similar to generation number, the population size is better to set 10, 20 and 50 for small, medium and large scale examples.

4.4. Selecting Operator

Selecting the best solutions for generating the best solution is vital. In the proposed method the algorithm estimates the values of the duration and resource availability based on triangular functions. This process can help us consider best values for resource availability (Appendix 3):

A) $R_k \sim Tri(\delta_k, \gamma_k)\ \forall\ k \quad (43)$

B) Suppose b is a random value.

C) If b<δ_k $r_k = (1+\alpha).R_k \quad (44)$

D) If $\gamma_k \leq$b<δ_k $r_k = R_k \quad (45)$

E) If b≥ δ_k $r_k = (1+\beta).R_k \quad (46)$

α *and* β are two random parameters that depend on the decision maker opinion. Appendix 3 shows the code of MOWGA using$\alpha = 0.4$ *and* $\beta = 0.6$. As seen, in this figure triangular probability function is used to estimate the resource values. For this purpose, the algorithm will make a random number (let's call b). If the number is smaller than Gamma, the value will be same as predefined value for R. If the random number is greater than Gamma but smaller than Delta, the amount of R increases 40% and if the value is greater than Delta, the value of R will increase 60%. Note that Gamma and delta are 2 input parameters that can be set in algorithm and must be estimated using the conditions of availability of each resource in real projects.

4.5. Fitness Function

Since nature of the developed mathematical model is multi-objective decision making model, the fitness function operator must follow MCDM principles. Weighting method is one of the most important and reliable methods for solving MCDM problems. In this method, each of the objective functions receives a weight (which can be set by decision maker) and then the algorithm will find the best solutions according to these weights (Appendix 4). The mathematical equation for calculating objective function based on weighting method will be presented:

$$Max\ Z_1: \sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} X_{(i,m,t)}\,.\,B_{(i,m,t)} \quad (47)$$

$$Min\ Z_2: \sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} Y_{(i,m)}\,.\,FC_{(i,m)} + \sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} X_{(i,m,t)}\,.\,VC_{(i,m,t)} \quad (48)$$

$$Min\ Z_3: \sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} X_{(i,m,t)} \quad (49)$$

Then, as explained in section 4.2, the weighting method objective function will be:

$$Max: W_1(\sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} X_{(i,m,t)} \cdot B_{(i,m,t)}) - W_2(\sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} Y_{(i,m)} \cdot FC_{(i,m)} + \sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} X_{(i,m,t)} \cdot VC_{(i,m,t)}) - W_3(\sum_{t=1}^{TH}\sum_{i=1}^{n}\sum_{m=1}^{M} X_{(i,m,t)}) \quad (50)$$

Note that since the second and third objective functions must be minimized, W_1 and W_2 are considered negative. As shown by Appendix 4, the algorithm uses weights (W1, W2 and W3) for calculating the objective function.

4.6. Cross-over Operator

The cross-over operator is an operator to choose best genes inside the solution chromosomes. In fact, this operator leads the solving algorithm to better and better stage (Appendix 5). In this research, we used the following mathematical equations for cross-over operates:

A) Create candidate List (note that the candidate list is list of activities that can be scheduled in the time.

If all predecessors of I is nominated in list, then I $\in$ $candidatelist$

B) Calculate the following formula for each members in candidate list (Normalized values):

$$Profit(i).D(i) - Expense(i).D(i) - Makespan(i).D(i) \quad (51)$$

C) Choose maximum value.

4.7. Mutation Operator

In metaheuristic algorithms, in order to prevent observing local optimum points, there must be a mechanism to escape them. In genetic algorithm, using mutation operator is a successful way to jump local optimum points (Appendix 6).

A) Generate a random number (say Y)

B) If Y>mutation rate, Run Cross over operator.

- Choose Maximum value in Candidate list.

C) If Y≤mutation rate, Run Mutation operator.

- Choose another element in Candidate list.

To choose the best value for mutation rate, a large scale sample is run. The results are indicated in Table 1.

Table 1. Finding Best Values for Mutation Rate

Mutation rate	0.1	0.2	0.3	0.4	0.5
Objective function variance	-0.3129	-0.2872	-0.2948	-0.2887	-0.3147

As mentioned before, mutation rate is a small chance for those solutions which are not good enough to be placed in next generation list. Therefore, it must be a small value between [0, 0.2]. The mechanism in MOWGA is set in a way that the more mutation rate is used in a generation, the greater mutation value will be set for next generations.

4.8. Termination the searching algorithm

The algorithm will be terminated if:

- Number of generations and populations are reached.
- All activities are scheduled.

4.9. Contribution of the proposed method

After developing the mathematical model and proposing the solving algorithm, it is time to compare our research with some other related researches in the literature. Table 2 shows the results.

4.10. Solving number of experiments (Validation)

In this section, a number of experiments that are set using different input values will be solved. Note that the number of activities, resources and modes are inspired from Delgoshaei et al. (2016). The remainder of the data are assumed by the authors. The examples are designed from small domain (with 5 activities and 3 resource) up to large domain (with 200 activities and 20 resources). In Tables 3 and 4, the solving time, objective function values and number of split activities are shown as outcomes. Others are input values.

Table 2. Comparing the novelties of this research with similar researches in the literature

Research	Model Type	Model	Idea	Scheduling Mode	Activity Precedence	LAG	Method	Largest Example
This research	Multi-objective	BP	Uncertain Duration Uncertain Resource	Multi-mode resource constraint scheduling	All Types	Yes	Multi-objective Weighting Genetic Algorithm	1000 activity, 100 resources
(Delgoshaei et al., 2016)	Single objective	NL-MIP	Fixed Duration Fixed Resource	Multi-mode resource constraint scheduling	All Types	Yes	Heuristic	1000 activity, 100 resources
(Yu et al., 2012)	Single Objective	NL-MIP	Fixed Duration Known Resource	Single Mode	FS	No	Heuristic	50 activities

Table 3. Outcomes Solving Number of Experiments (5 to 200 activities)

No.	Activity	Resource	Mode	Resources Capacity	Delta	Gamma	Generation	Population Size	Mutation Operator
1	5	3	2	[20 14 42]	0.25	0.75	10	10	0.1
2	6	2	2	[182 140]	0.25	0.75	10	10	0.1
3	1	2	2	[27.2 16.8]	0.25	0.75	10	10	0.1
4	1	3	4	[16.8 25.2 14.0]	0.25	0.75	10	10	0.1
5	1	3	3	[24.0 33.6 39.2]	0.25	0.75	10	10	0.1
6	2	3	4	[43.2 28.0 30.0]	0.25	0.75	10	10	0.1
7	2	2	3	[3360 2800]	0.25	0.75	20	30	0.1
8	3	2	4	[630 560]	0.25	0.75	20	30	0.1
9	100	10	4	[45.0 56.0 63.0 70.0 89.6 63.0 89.6 80.0 91.0 45.0]	0.25	0.75	30	50	0.1
10	200	20	3	[40 60 56 308 264 88 184 120 70 56 196 84 232 168 72 70 184 120 150 216]	0.25	0.75	30	50	0.1

Table 4. Outcomes Solving Number of Experiments (5 to 200 activities)

No	W=[W1,W2,W3]	Profit	Total Cost	Make span	OFV	CPU time	M.S.A	Cumulative Resource Usage
1	[0.33 -0.33 -0.33]	809.9	84.37	30	-0.313	0.54	0	[89 177 224]
2	[0.33 -0.33 -0.33]	883.5	93.71	23	-0.278	0.65	0	[228 182]
3	[0.33 -0.33 -0.33]	2600.	356.3	55	-0.283	3.72	2	[521 597]
4	[0.33 -0.33 -0.33]	2952.	328.1	58	-0.307	4	5	[516 640 580]
5	[0.33 -0.33 -0.33]	3870.	451.2	46	-0.305	58	2	[716 792 717]
6	[0.33 -0.33 -0.33]	2900.	1636	53	-0.322	273	2	[860 880 855]
7	[0.33 -0.33 -0.33]	3014.	1982.	68	-0.310	438	0	[960 889]
8	[0.33 -0.33 -0.33]	4709.	1935.	57	-0.311	567	0	[1428 1446]
9	[0.33 -0.33 -0.33]	14367	9071.4	262	-0.318	1093	4	[6082 7050 6070 7060 6478 6156 6338 6682 6264 6580]
10	[0.33 -0.33 -0.33]	16213	26834	380	-0.324	1133.91	29	[11312 11934 12588 11274 11940 12538 11228 11128 11530 11590 11312 11934 12588

*Per seconds **M.S.A: Maximum split activities

5. Analysing the results

After solving the examples, a number of results are achieved. These results are explained and analysed:

1- As seen, the proposed MOWGA can successfully schedule all examples with various activities and resources. Therefore, the proposed method is verified to be used.

2- The results show that the objective functions are sensitive to the increasing complexity of the examples (by increasing the resource uncertainty). Figure 3 shows the slope of the each of the solo objective functions to the increasing the resource uncertainty. As seen, the total cost has more slope than the other objectives in almost all cases. Figures 4, 5 and 6 show the value of solo objective functions in different case studies.

3- Figure 7 shows the value of the multi-objective function. It is noted that the maximum slope of the multi-objective function in the MOWGA is 0.333 which is achieved while profit objective is in maximum and total cost and makespan are 0. Despite that, the lowest slope value of the multi-objective function is -0.666 while profit value is 0 and total cost and makespan are at maximum. Comparing the results of profit, total cost and makespan with multi-objective function shows that while multi-objective is at the maximum, not all objective functions are necessarily in optimum condition. However, the combination of the objectives is in the best value. For example, in example 6, while the best value of multi-objective function of MOWGA is -0.3219, the values of profit, cost and makespan are 2900.5 $, 1636 $ and 53 days respectively. If we see the following list that is achieved during solving the case study, we find that the optimizing multi-objective does not mean optimizing all objectives one by one (appendix-example number 6).

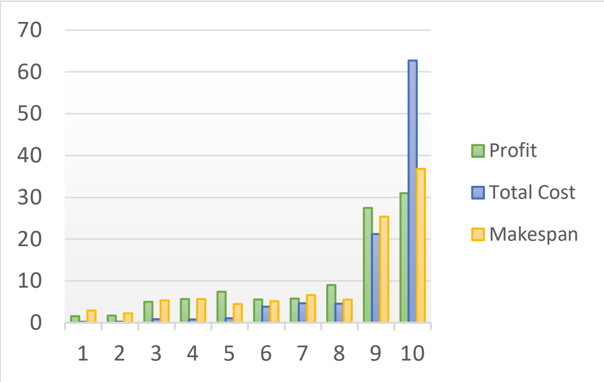
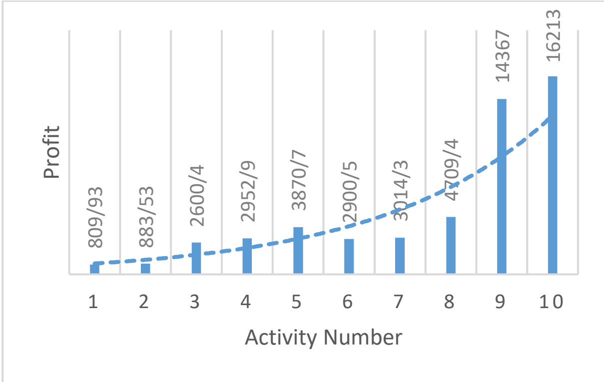


Figure 3. Comparing the slope of the solo objectives



Fiure 4. Values of Profit in the solved examples

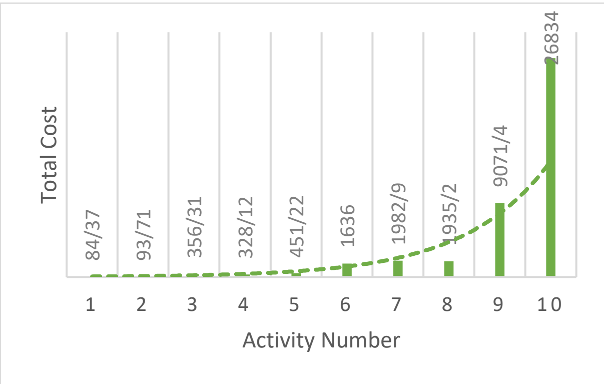


Figure 5. Values of Total cost in the solved examples

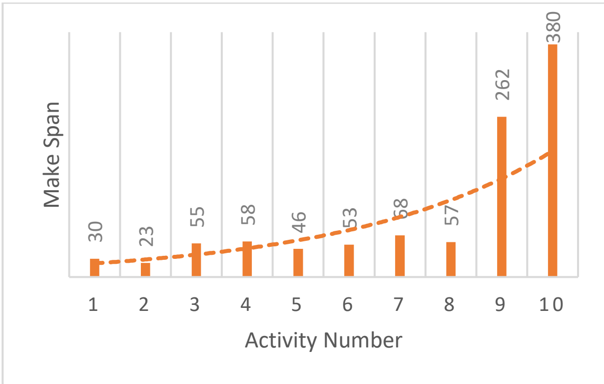


Figure 6. Values of Makespan in the solved examples

```
Profit =
1.0e+03 *
Columns 1 through 16
2.9164    3.0126    2.9801    3.0126    3.0127    2.9340    3.0026    3.0101    2.9003
2.9107    2.9125    2.9725    3.0101    2.9695    2.9413    3.0126
Columns 17 through 30
3.0126    3.0126    3.0101    2.9795    2.9005    2.9486    2.8625    2.9340    3.0126
3.0101    2.8743    2.9795    3.0126    2.9770

Total_Cost =
1.0e+03 *
Columns 1 through 16
1.6349    1.6941    1.6707    1.6941    1.6941    1.6717    1.6840    1.6908    1.6209
1.6462    1.6440    1.6640    1.6908    1.6620    1.6499    1.6941
Columns 17 through 30
1.6941    1.6941    1.6908    1.6720    1.6360    1.6580    1.6040    1.6717    1.6941
1.6908    1.6219    1.6720    1.6941    1.6703

Makespan =
Columns 1 through 26
```

```
55    55    55    55    55    54    55    55    55    53    55    55    55    55    55
55    55    55    55    55    53    55    55    54    55    55
Columns 27 through 30
55    55    55    55
Multi_objective_Function =
Columns 1 through 16
-0.3330   -0.3330   -0.3320   -0.3330   -0.3330   -0.3312   -0.3321   -0.3326   -0.3310
-0.3227   -0.3342   -0.3315   -0.3326   -0.3315   -0.3322   -0.3330

Columns 17 through 30
-0.3330   -0.3330   -0.3326   -0.3323   -0.3219   -0.3330   -0.3319   -0.3312   -0.3330
-0.3326   -0.3341   -0.3323   -0.3330   -0.3323
```

Figure 1. The Multi-objective value for the example number 6

As seen, if we consider profit solely, the best objective function will be 3126 $ which is marked with blue colour. But the best combination of the objective functions is seen in column 21 which can be seen while profit is 2900.5 $, total cost is 1636 $ and makespan is 53 days. This is the exact reason why we should consider multi-objective despite a solo objective (green highlights).

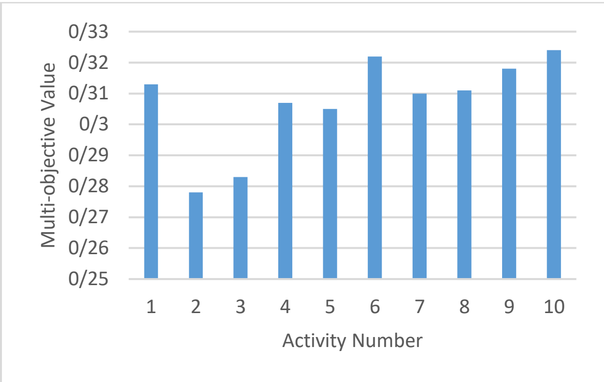


Figure 7. Result of Multi-objective function in the solved examples

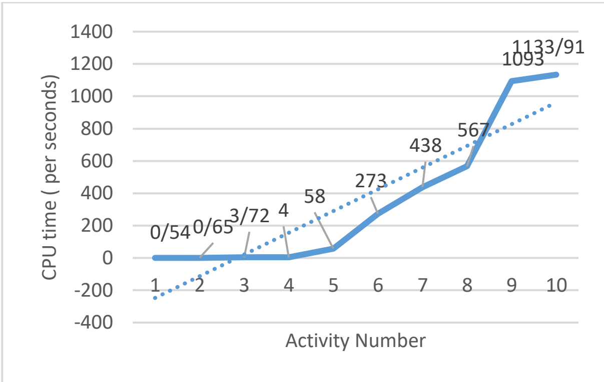


Figure 8. The trend of increasing computation time in the solved examples

4- Figure 8 shows that by increasing the complexity of the examples in terms of number of activities and amount of resource uncertainties, the computation time increases with high slope. The results also indicated that the algorithm is able to schedule the examples with 200 activities and 20 uncertain resources in less than 18.5 minutes which is reasonable. 5- The mechanism of the algorithm is designed in a way that it can take apart the activities where necessary to prevent the resource over-allocation in specific days (Figure 9). Note that there are many researches that used the split ability and that activity split is run in Microsoft Office Project and Primavera.

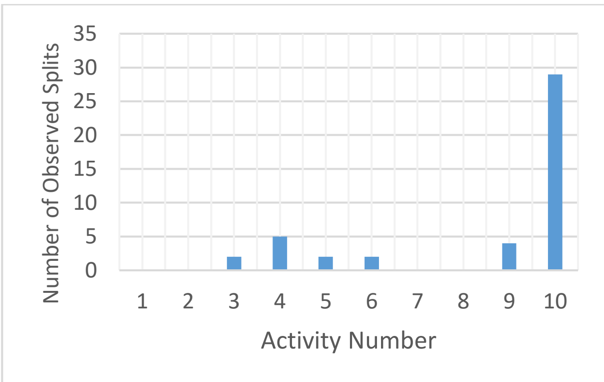


Figure 9. Quantity of split activities in solved examples

5.1. Verifying the method with real data

In this section, the data of a real project will be applied for the proposed method. For this purpose constructing a simple building is taken into account. This example has 22 activities and 5 uncertain resources are considered.

In the first step the problem is scheduled by MSP®2016. After using resource levelling module in MSP® 2016, it can be seen that there are still some resources that remained over-allocated which means the module could not be able to resolve the problem completely (figure 10).

Table 5. Activity precedence for the Experiment in details

ID	Activity	Duration (days)	Precedence
1	Ground levelling	7	-
2	Foundation	14	1FS
3	Structure	45	2FS
4	Roofing	10	3FS
5	Wall erection	15	4FS
6	flooring	7	5SS
7	Water Piping	7	5SS+5 days
8	Ego Piping	5	6SS+5 days
9	Isolating the roof	2	4FS
10	Cabling	15	4FS+2days
11	installing windows	10	5FS
12	Installing doors	10	5FS,11SS+2days
13	Installing Mosaics	7	6FS
14	Installing ceramics	5	13FS
15	Installing Lamps	2	10FS
16	Installing Toilets	5	11FS
17	Installing tiles	5	16SS
18	Installing taps	5	16SS+2days
19	Installing cabinets	3	14FS
20	Gypsum	7	19FS
21	Painting	7	20FS+7days
22	Installing Air condition	1	21FF

Table 6. Resource availability for the Experiment in details

Resource	Maximum Amount
Civil Engineer (R1)	Triangular(5~8)
Mechanical Engineer (R2)	Triangular(3~4)
Electrical Engineer (R3)	Triangular(2~3)
Foreman (R4)	Triangular(4~6)
Simple Worker (R5)	Triangular(20~22)

Table 7. Resource requirements for the experiment in details

Activity	R1	R2	R3	R4	R5
Ground levelling	1	0	0	2	5
Foundation	1	0	0	2	4
Structure	1	0	0	3	10
Roofing	1	0	0	2	5
Wall erection	1	0	0	2	10
flooring	1	0	0	2	6
Water Piping	0	1	0	0	4
Ego Piping	0	1	0	0	4
Isolating the roof	1	0	0	1	3
Cabling	0	0	1	0	4
installing windows	1	0	0	1	3
Installing doors	1	0	0	1	3
Installing Mosaics	1	0	0	2	4
Installing ceramics	1	0	0	2	4
Installing Lamps	0	0	1	0	3
Installing Toilets	1	1	0	1	3
Installing tiles	1	1	0	1	3
Installing taps	0	1	0	0	2
Installing cabinets	0	1	0	1	2
Gypsum	1	0	0	2	5
Painting	1	0	0	2	5
Installing Air condition	0	0	1	0	3

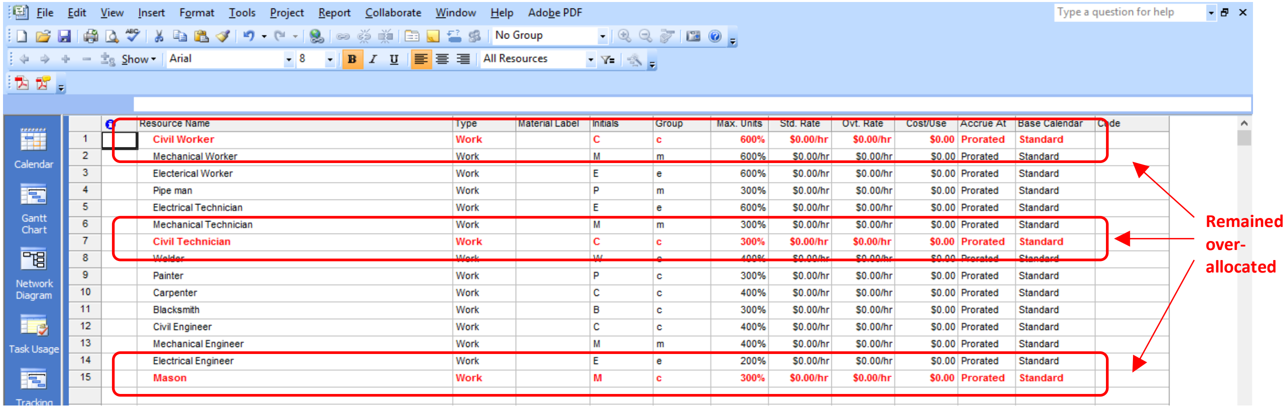


Figure 10. After using resource level module even while effort driven option is turned on, there are still some resources that remained over-allocated

Thus, the case study is solved by the proposed solving algorithm while all resources constraint are estimated by Triangular distribution function as presented by Table 7. This example is a small case study. The number of generations and population size can be set 10 and 10 respectively.

Table 8. Results of solving the construction building

Number of activities	22
Number of uncertain resources	5
Resource probability function	Delta =0.25 Gamma=0.75
Profit	27872
Cost	30789
Makespan	139
Multi-objective function	-0.333
Number of activity split	1
Computation time	12.107

The mutation rate is also 0.1. After solving the case study the following results are achieved (Table 8). Then, the schedule of the building is shown. As seen, only activity number 10 is taken apart due to lack of resources (Figure 11).

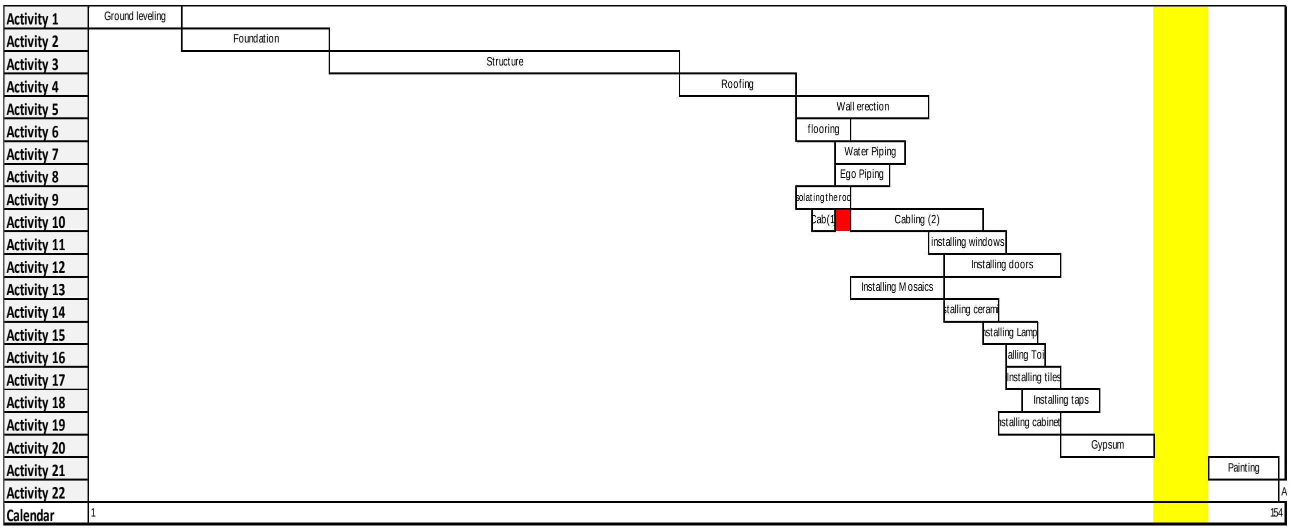


Figure 11. The Gantt chart of the solved case study

The boxes in the gained Gantt chart show the length of the activities. For example in the first row, the 1st activity is started from the day 1 to day 12. Similarly the red lines show those activities that are taken apart (split). For example activity number 10 is stopped for 3 days and then activity 10 is executed. Figure 12 shows a view of the corridor of the hospital before and after installing the false ceiling.

Figure 12. A view of the project before and after installing false celling

6. Conclusions

The idea of managing the limited resources becomes more and more important every day. Today in many countries there are some resources available which are not enough to respond to all needs of a country. Therefore it seems important to manage rare resources. In project management discipline, scheduling projects with constraint resources is a very important field. In this research, a new method is developed for scheduling resource constraint problem while the availability of each resource is not fixed and may be varied from time to time. To show this, a triangular distribution function is used which expressed the resource availability in terms of optimistic, most probable and pessimistic way. To verify the method, a number of examples were used. The results are then evaluated using measuring indexes. Here are the outcomes: Increasing in resource uncertainty raises the scheduling variance in terms of completion time (up to 16%). Moreover, it is found that increasing in resource uncertainty causes achieving less profit and (up to 21%). The proposed MOWGA method can successfully schedule all examples in various conditions. The method is developed in a way that the outcome shows the un-coded Gantt chart of the project and can be used with no changes. More importantly, the proposed MOWGA algorithm can count and shows the split activities preciously while the number of activities is large (200 in this research). It is also observed that the computation time of the solving method depends on the number of activities, number of the resources, and variety of the resource uncertainty respectively. The proposed method is applied for a project with real data. The results confirm that the method can successfully be used in the real conditions and in reasonable time. Therefore, it is concluded that an appropriate scheduling method can help reducing the resource uncertainty harms which can consequently yield to maximize the makespan of the project. Further research can be conducted in the following directions:

1. In many projects, some activities are executed by subcontractors. Therefore, subcontractor scheduling is a critical issue. It is suggested that using subcontractor with limited capacity considered in scheduling MRCPSP problems.
2. In practice, different workers have different skills of performing similar tasks. It is recommended to develop the model by considering the skill level of the workers and sub-contractors.
3. In real projects, there are many temporary workers to complete activates. Such workers will remain to the end of a project and the number of them may vary. It is suggested to consider temporary workers and fixed operators wage in the objective function.
4. During completion of a project, there was much wasted material. There must be a plan to use them as much as possible. It is recommended to consider the material waste management in scheduling the resource constraint problem to minimize the resource wastes.

Acknowledgments: The authors sincerely appreciate the editor and anonymous reviewers for their constructive comments.

References

Ballestín, F., Valls, V., and Quintanilla, S. (2008). Pre-emption in resource-constrained project scheduling. *European Journal of Operational Research*, Vol. 189(3), pp. 1136-1152.

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

Castejón-Limas, M., Ordieres-Meré, J., González-Marcos, A., and González-Castro, V. (2011). Effort estimates through project complexity. *Annals of Operations research*, Vol. 186(1), pp. 395-406.

Chen, J., and Askin, R. G. (2009). Project selection, scheduling and resource allocation with time dependent returns. *European Journal of Operational Research*, Vol. 193(1), pp. 23-34.

Chen, W., Shi, Y.-j., Teng, H.-f., Lan, X.-p., and Hu, L.-c. (2010). An efficient hybrid algorithm for resource-constrained project scheduling. *Information Sciences*, Vol. 180(6), pp. 1031-1039.

Chtourou, H., and Haouari, M. (2008). A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. *Computers & Industrial Engineering*, Vol. 55(1), pp.183-194.

Damay, J., Quilliot, A., and Sanlaville, E. (2007). Linear programming based algorithms for preemptive and non-preemptive RCPSP. *European Journal of Operational Research*, Vol. 182(3), pp. 1012-1022.

Davari, M., and Demeulemeester, E. (2019). The proactive and reactive resource-constrained project scheduling problem. *Journal of Scheduling, 22*(2), pp. 211-237.

Delgoshaei, A., Ariffin, M., Baharudin, B., and Leman, Z. (2015). Minimizing makespan of a resource-constrained scheduling problem: A hybrid greedy and genetic algorithms. *International Journal of Industrial Engineering Computations*, Vol. 6(4), pp. 503-520.

Delgoshaei, A., Ariffin, M. K., Baharudin, B., and Leman, Z. (2014). A backward approach for maximizing net present value of multi-mode pre-emptive resource-constrained project scheduling problem with discounted cash flows using simulated annealing algorithm. *International Journal of Industrial Engineering and Management*, Vol. 5(3), pp. 151-158.

Delgoshaei, A., Rabczuk, T., Ali, A.,and Ariffin, M. K. A. (2017). An applicable method for modifying over-allocated multi-mode resource constraint schedules in the presence of preemptive resources. *Annals of Operations research*, Vol. 259(1-2), pp. 85-117.

Kadri, R. L., and Boctor, F. F. (2018). An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *European Journal of Operational Research*, Vol. 265(2), pp. 454-462.

Ke, H., and Liu, B. (2010). Fuzzy project scheduling problem and its hybrid intelligent algorithm. *Applied Mathematical Modelling*, Vol. 34(2), pp. 301-308.

Kreter, S., Rieck, J., and Zimmermann, J. (2016). Models and solution procedures for the resource-constrained project scheduling problem with general temporal constraints and calendars. *European Journal of Operational Research*, Vol. 251(2), pp. 387-403.

Laslo, Z. (2010). Project portfolio management: An integrated method for resource planning and scheduling to minimize planning/scheduling-dependent expenses. *International Journal of Project Management*, Vol. 28(6), pp. 609-618.

Li, X., Fang, S.-C., Guo, X., Deng, Z., and Qi, J. (2016). An extended model for project portfolio selection with project divisibility and interdependency. *Journal of Systems Science and Systems Engineering*, Vol. 25(1), pp. 119-138.

Naber, A., and Kolisch, R. (2014). MIP models for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, Vol. 239(2), pp. 335-348.

Narayanan, S., Balasubramanian, S., and Swaminathan, J. M. (2009). A matter of balance: Specialization, task variety, and individual learning in a software maintenance environment. *Management Science*, Vol. 55(11), pp. 1861-1876.

Pérez, E., Posada, M., and Lorenzana, A. (2016). Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms. *Soft Computing*, Vol. 20(5), pp. 1879-1896.

Seifi, M., and Tavakkoli-Moghaddam, R. (2008). A new bi-objective model for a multi-mode resource-constrained project scheduling problem with discounted cash flows and four payment models. *Int. J. of Engineering, Transaction A: Basic*, Vol. 21(4), pp. 347-360.

Sprecher, A. (2012). *Resource-constrained project scheduling: Exact methods for the multi-mode case* (Vol. 409): Springer Science & Business Media.

Tao, S., Wu, C., Sheng, Z., and Wang, X. (2018). Space-time repetitive project scheduling considering location and congestion. *Journal of Computing in Civil Engineering*, Vol. 32(3), pp. 04018017

Vaez, P., Bijari, M., and Moslehi, G. (2018). Simultaneous scheduling and lot-sizing with earliness/tardiness penalties. *International Journal of Planning and Scheduling,* Vol. 2(4), pp. 273-291.

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

Vanhoucke, M., and Debels, D. (2008). The impact of various activity assumptions on the lead time and resource utilization of resource-constrained projects. *Computers & Industrial Engineering,* Vol. 54(1), pp. 140-154.

Wang, L., and Zheng, X.-l. (2018). A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem. *Swarm and Evolutionary Computation,* Vol. 38, pp. 54-63.

Wang, W.-x., Wang, X., Ge, X.-l., and Deng, L. (2014). Multi-objective optimization model for multi-project scheduling on critical chain. *Advances in Engineering Software,* Vol. 68, pp. 33-39.

Yu, L., Wang, S., Wen, F., and Lai, K. K. (2012). Genetic algorithm-based multi-criteria project portfolio selection. *Annals of Operations research,* Vol. 197(1), pp. 71-86.

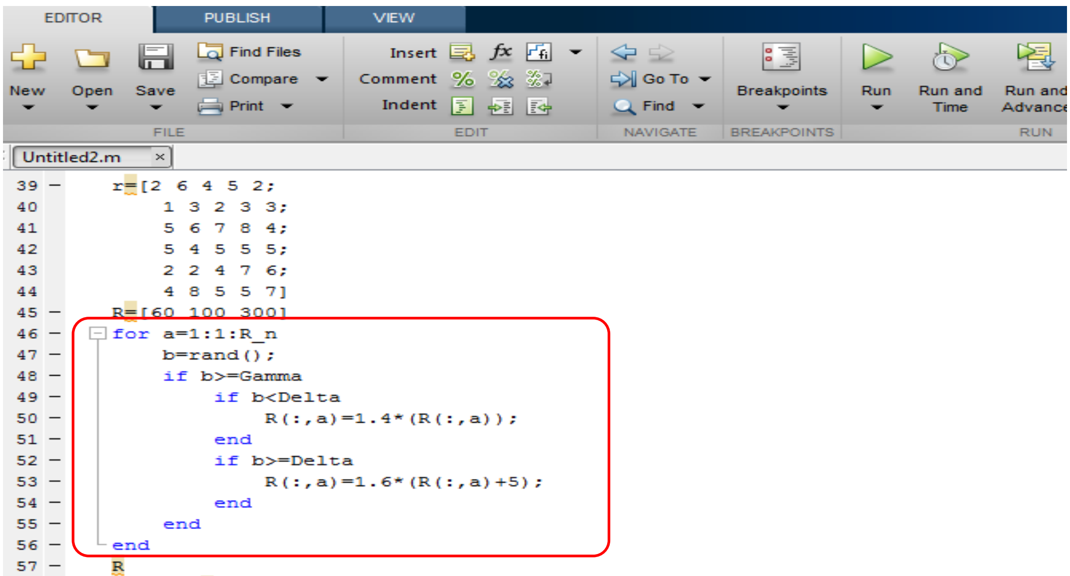
Appendix

Appendix 1. Estimating the Suitable value for Generation Number in MOWGA

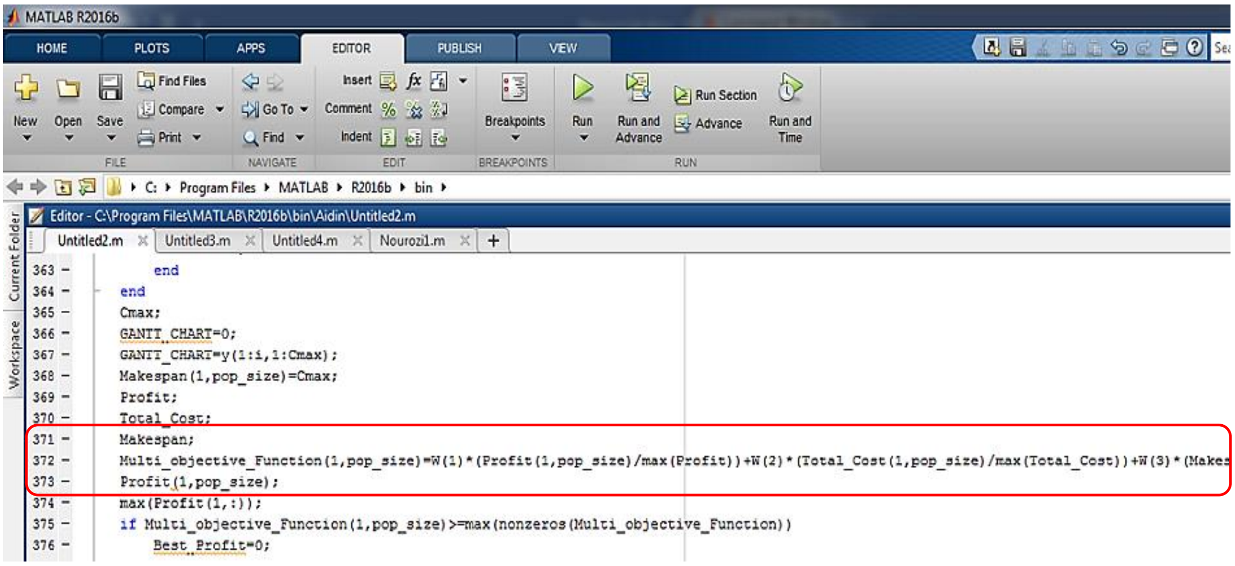
Problem Scale	Generation Number			Best Value
Small	10	20	30	-0.3405
Medium	10	20	50	-0.3326
Large	10	30	50	-0.3421

Appendix 2. Estimating the Suitable value for Population Size Number in MOWGA

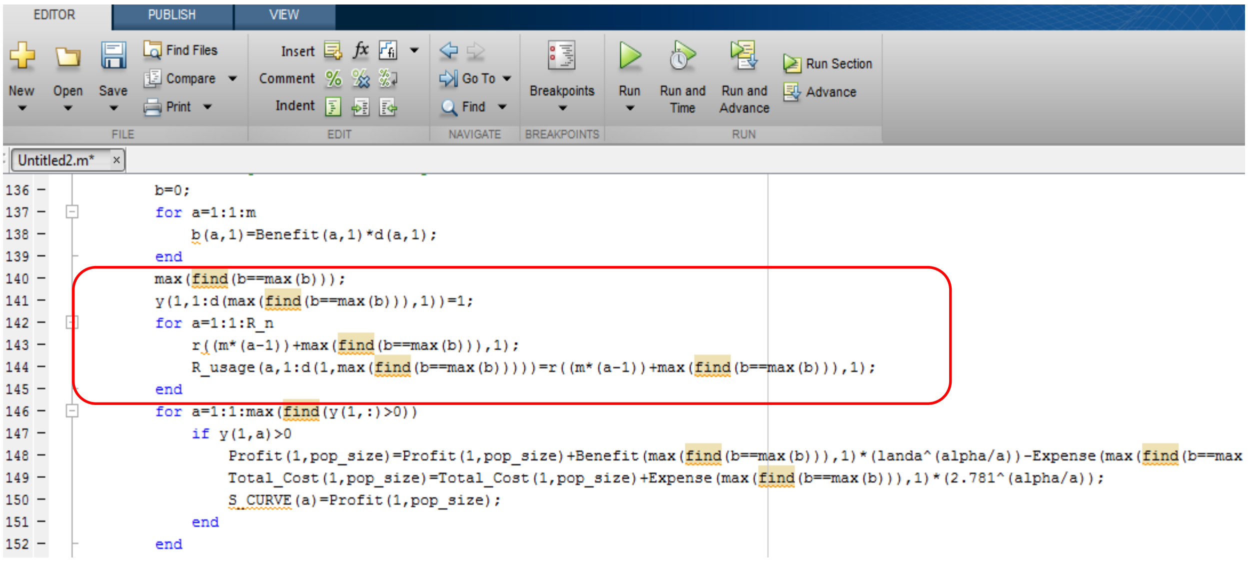
Problem Scale	Population Size			Best Value
Small	10	20	50	-0.3304
Medium	20	50	70	-0.3540
Large	20	50	150	-0.3072



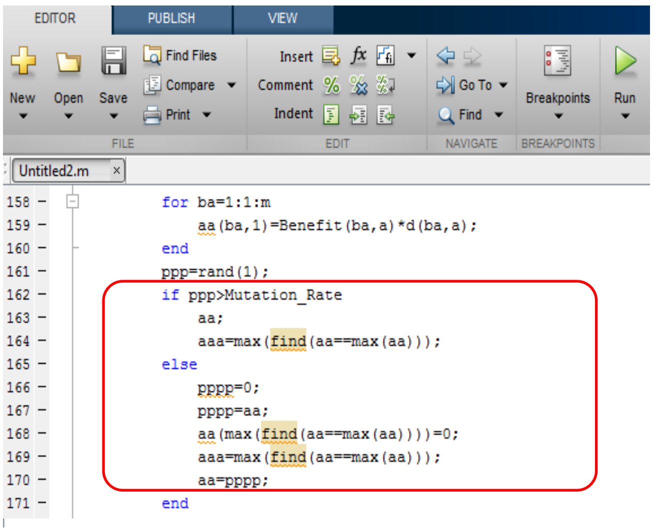
Appendix 3. Using Triangular Function for Uncertain Resources in MOWGA



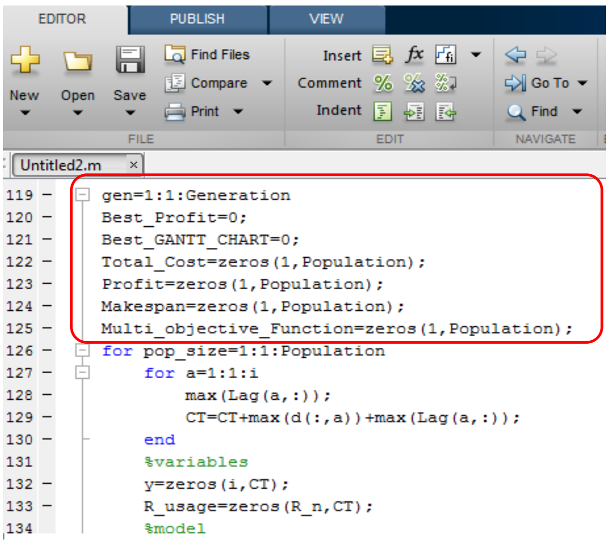
Appendix 4. Using Weighting Method for Multi-objective Function Calculating



Appendix 2. Cross-over operator for choosing best activity in MOWGA



Appendix 3. Performance of Mutation Operator in MOWGA



Appendix 4. Stopping criteria in MOWGA