

## A genetic algorithm for scheduling multimode resource-constrained project problem in the presence of preemptive resources

Aidin Delgoshaei<sup>a\*</sup>, Sepehr Esmaeili Hanjani<sup>b</sup> and Amir Hossein Nasiri<sup>c</sup>

<sup>a</sup>Department of Mechanical and Manufacturing Engineering, University Putra Malaysia, 43300 UPM, Serdang, Kuala Lumpur, Malaysia

<sup>b</sup>Department of Industrial Engineering, Azad University, Hashtgerd Branch, Alborz, Iran

<sup>c</sup>Department of Industrial Engineering, Azad University, Central Tehran Branch, Tehran, Iran

### CHRONICLE

### ABSTRACT

#### Article history:

Received: January 8 2019  
Received in revised format: January 27 2019  
Accepted: March 19 2019  
Available online:  
March 19 2019

#### Keywords:

Multimode Project Scheduling  
Genetic Algorithm  
Pre-emptive Constrained Resources  
Discounted Cash Flows

In this paper, a backward approach is proposed for maximizing net present value (NPV) in multi-mode resource constrained project scheduling problem while assuming discounted positive cash flows (MRCPSP-DCF). The progress payment method is used and all resources are considered as pre-emptible. The proposed approach maximizes NPV using unscheduled resources through resource calendar in backward mode. For this purpose, a Genetic Algorithm is applied to solve experimental cases with 50 variables and the results are compared with forward serial programming method. The remarkable results reveal that the backward approach is an effective way to maximize NPV in MRCPSP-DC while activity splitting is allowed. The algorithm is flexible enough to be used in real project.

© 2019 by the authors; licensee Growing Science, Canada.

## 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 the projects which are failed or 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 wasted. Therefore, having a plan to predict, schedule and monitor the resources during project implementation is a vital. Project scheduling as one of the areas of project management that is a 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 big 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.

\* Corresponding author.

E-mail address: [delgoshaei.aidin@gmail.com](mailto:delgoshaei.aidin@gmail.com) (A. Delgoshaei)

- 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 countries there are lots of 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. This will reduce the harms of resource uncertainty.

## 2. Literature Review

This Section presents a review of mathematical programming models and techniques to solve the models. For this purpose some novel researches are reviewed. The advantages and techniques are explained. Such approach helps us to choose the best method in next Sections. In Continue, the researchers are divided into 3 sub categories which are completion time, cost, profit. Afterward, some review papers are introduced for further studies. Well-known objective functions and constraint are explained.

### 2.1. Minimizing Completion Time of a Project ( $C_{max}$ )

Minimizing completion time of a project is an attempt to finish a project as soon as possible. This objective is the most popular objective that is considered in many researches. Kim et al. (2005) developed a fuzzy programming based algorithm that called fuzzy logic controller with genetic algorithm which worked based on serial programming method to schedule resource-constrained multiple project scheduling problems. Ke et al. (2010) also used fuzzy operator in genetic algorithm for minimizing completion time in MRCPSP. Vanhoucke et al. (2008) argued that activity duration should considered variable. Then they developed a model to minimize project completion time by using activity preemption and rapid execution of activities. Van Peteghem et al. (2010) focused on the impact of preemptive resources in minimizing completion time of the MRCPSP problems. To solve the proposed problem a genetic algorithm is proposed. Kreter et al. (2016) focused on RCPSP with general temporal constraints and break-calendars. For this purpose 3 linear models are developed and solved with CPLEX 16.0. Pérez et al. (2016) used multi-modal genetic algorithm in RCPSP for generating solutions with good quality. Their algorithm was then chose the best solution in terms of makespan and average percent delay. Vaez (2017) considered penalties for earliness and tardiness in scheduling process.

### 2.2 Optimizing Robustness of Schedule

The aim of this section is finding schedules that are flexible enough to be used in practice. Normally the schedules that are outcomes of mathematical algorithms are not flexible to be used in real cases. In this sections using some indexes or objective functions scientists are trying to provide more robust solutions in terms of quality of the schedules (Van de Vonder et al., 2005). Afterward, Van de Vonder et al. (2006) focused on resource constraint impacts in determining trade-off values between quality-robustness and solution-robustness in RCPSP. Abbasi et al. (2006) developed a model for minimizing completion time and increasing the float time. Then Chtourou et al. (2008) developed a RCPSP model for minimizing completion time as a criteria for generating robust solutions in the next stage where twelve indicators was used to choose best solutions. Hendricks et al. (2002) used degree of specialization for each of the human resources. Such approach can help evaluating the solution that are provided by their method. Ward et al. (2003) argued that project risks should not been ignored in scheduling projects. Such approach helps the authors to be ready for confronting with project risks before they are happening. Delgoshaei et al. (2017) dealt with scheduling systems while using outsource services. Castejón-Limas et al. (2011) focused on some other managerial factors to be considered for scheduling projects.

In MRCPSP problems activities can be scheduled in more than one way and therefore the activities might have different durations and resources and consequently cash flows. Węglarz et al. (2011) published a review paper in regard with the literature of the MRCPSPs. Laslo (2010) proposed an

new method for minimizing negative dependent cash flows in scheduling process. Financial aspect of scheduling projects should not be ignored as they may cause providing infeasible or useless schedules. It should be mentioned that in MRCPSP studies there are 2 main cash flows can be considered. Positive cash flows that are referred to earned money of the project. Negative cash flows in contrast are those costs that should be considered for completing projects such as expert salary, worker's wage, and maintenance and services costs. Cash flows whether they are positive or negative can affect activity due date, completion time, resource availability, material purchasing etc. (Yu et al., 2012) employed genetic algorithm for selecting multi-criteria project portfolio problem. Their goals were project interactions and preference information. Delgoshaei et al. (2016b) reviewed different resource transferring methods in projects. Rabbani et al. (2017) used metaheuristics for providing robust solutions in a passive optical network planning. ZIAEE (2017) addressed flexible job-shop scheduling for production lines. This idea seems interesting in scheduling projects as well. Yan et al. (2009) focused on finding a solution to prepare a fast response to maritime disasters using heuristics. Metaheuristics are also used in many cases for solving MRCPSP problems. Among all metaheuristics genetic algorithm is used more frequently than others Lin et al. (2008). Kim et al. (2005) used fuzzy based genetic algorithm for minimizing completion time and tardiness penalty. Naber et al. (2014) proposed 4 RCPSP based profiles discrete-time mixed integer programming models with flexible resource. To solve the model they offered preprocessing and priority-based heuristic methods. Delgoshaei et al. (2016a) explained the uncertain conditions that may cause imbalance in a system. (Ke et al., 2010) also used a combination of fuzzy and genetic algorithm for optimizing project cost respecting to completion time (see also Chen et al. (2009); Hartmann et al. (2010)). Particle swarm optimization method is also used by Jarboui et al. (2008) for solving MRCPSPs. Sharon et al. (2015) proposed an applicable method for work breakdown structure in order to provide more robust solutions. Papke-Shields et al. (2017) focused on strategic characteristics in planning of a project.

### *2.3. Maximizing Profit of the project (Profit/Net Present Value (NPV))*

In many researches in contrast, the aim were to increase the benefit of scheduling activities. Each of the activities has positive and negative cash flows that can be changed by executing mode of activity, scheduling time and different types of resource. Maximizing NPV is logic way to choose a project or reject it. If NPV of a project is calculated and the result is a negative value then it can be concluded that the project does not provides any benefit. Laslo (2010) considered negative cash flows to minimize the project completion cost as objective function of their model.

Mika et al. (2005) developed a mathematical model renewable and non-renewable resource constraint scheduling method. Their objective was maximizing NPV which was solved by a hybrid simulated annealing and Tabu search algorithms.

Recently many researchers considered preemptive resource for showing the priority of activities in using resources. Buddhakulsomsiri et al. (2006) argued that preemptive resources can affect to makespan of the project and hence it must be considered during scheduling problems. Damay et al. (2007) used LP method where the RCPSP has preemptive resources. In continue Ballestín et al. (2008) developed a heuristic algorithm in similar problem. Seifi et al. (2008) solved the maximizing NPV problem in 4 different payment methods. Van Peteghem et al. (2010) developed a new method for minimizing completion time in an activity split allowed multi-mode resource constraint method. Reviewing the literature of project scheduling problems show that maximizing NPV of the MRCPSP's while durations of activities are uncertain is not developed. Moreover the negative cash flows are mainly not considered in maximizing NPV. Hence this research continues Naber et al. (2014) research as base-paper and continues their research by considering uncertain durations of activities, considering contracted time as a constraint for providing schedules, considering contracted cost as a constraint for providing schedules.

### 3. Research Methodology

In this section, we present the proposed mathematical scheduling model with the aim of NPV in the condition of resource constraint. The model considers multi-mode of execution for each activity. Our aim is to survey how allocation of pre-emptive resources can change the activity scheduling and what is their impact to the net present value. As summary we can mention the advantages of the proposed model as follows: Considering pre-emptive resources in maximizing net present value of project, considering multi-mode execution activities, considering activity splitting ability with respect of the predecessors, using useless amount of remained resources. Main constraints are the resource capacity, fixed time of the planning, only positive cash flow is considered, exact occurrence of all activities, and exact duration of each activity mode and exact cash flow for each activity mode.

#### 3.1. Problem Definitions

In order to classify models easily, in this section we define problems with a unique code as below:

$$n|m|k|th|it|g|p \quad (1)$$

In this classification,  $n$  presents number of Activities,  $m$  is number of activity mode,  $k$  is number of resource types,  $th$  is time horizon of the project,  $it$  is number of time iterations of the program,  $g$  is number of generations is each time iterations during time horizon,  $p$  indicates population size.

#### 3.2. Assumptions

Following is the properties of the model:

1. Model is presented in AON (Activity on Node) structure.
2. PP (Progress Payment) is selected as the payment model.
3. Resources are considered preemptive.
4. The preemptive resources have limited capacities.
5. In this study, positive cash flows are considered as weight factor of each activity.
6. Activities can be executed in different modes.
7. Activities are allowed to move only in their free-float time.
8. All improving movements will carry out in backward mode.

#### 3.3. Subscripts

Subscripts used in the model are considered as follows:

$i$ = number of activities which is a  $1 \times n$  matrix ( $[1 \dots N]_{1 \times n}$ )

$k$ =number of resource types which is a  $1 \times m$  matrix ( $[1 \dots K]_{1 \times k}$ )

$t$ =available time horizon for production ( $t=1, 2 \dots T$ )

$m$ =number of modes of performance ( $[1 \dots M]_{1 \times m}$ )

#### 3.4. Parameters

The list of parameters and notations is as follows:

*Resource\_Capacity*: illustrates available resource in sub periods:

$$[RC_1 \dots RC_k]_{1 \times k}$$

As result, number of in-process activities that queued in a waiting list to be served by a preemptive resource can be expressed using below formula:

$$RC_k = R_k - \sum_{i=1}^n r_{(i,k)} * y_{(i,m,t)} \quad ; \forall (k \in K) \& (t \in TH) \quad (2)$$

*Activity\_time*: shows duration of each activity considering different execution modes.

$$\begin{bmatrix} 11 & \dots & 1n \\ \vdots & \ddots & \vdots \\ m1 & \dots & mn \end{bmatrix}_{m \times n}$$

*Activity\_sequence* matrix is used in mathematical programming to show precedence relations between activities.

$$\begin{bmatrix} 11 & \dots & 1n \\ \vdots & \ddots & \vdots \\ n1 & \dots & in \end{bmatrix}_{n \times n}$$

$CF(i, m)$  = positive cash flow of activity  $i$  while it performs in mode  $m$

$r(i, k)$  = usage amount of resource type  $k$  for activity  $i$

$R(k)$  = available level of resource type  $k$

$D(i, m)$  = duration of Activity  $i$  while it performs in mode  $m$

$TH$  = time horizon of the project

$\alpha$  = discounted rate

### 3.5 Decision Variables

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

$S(i)$  = start time of activity  $i$

### 3.6 Mathematical Model

As mentioned in the previous parts, studying an MRCPSPP problem is the major objective of this paper. We supposed to have  $N$  activities on an AON network. Hence, Mathematical model is now written as follows:

$$\text{Max: } \sum_{t=1}^{TH} \sum_{i=1}^n \sum_{t=S_i}^{\min(S_i+d_{i,m}, t)} y_{(i,m,t)} \cdot (CF_{i,m} \cdot d_{i,m}) \cdot e^{\alpha/(S_i+d_{i,m})} \tag{3}$$

$S.T:$

$$S_i: \left\{ \sum_{t=ES_i}^{LF_i} y_{(i,m,t)} - \sum_{t=ES_i}^{LF_i} y_{(i,m,t+1)} = 1 \mid y_{(i,m,t-1)} = 0, y_{(i,m,t=ES_i)} \right\} ; \forall (i, j) \in P_i \tag{4}$$

$$S_i + d_{i,m} < S_j ; \forall (i, j) \in P_i \tag{5}$$

$$S_1 = 0 \tag{6}$$

$$S_n \leq TH - d_{i,m} \tag{7}$$

$$\sum_{t=S_i}^{LF_i} y_{i,m,t} = d_{i,m} \cdot y_{(i,m,t=S_i)} ; \forall i = 1, \dots, n \ \& \ m = 1, \dots, M \tag{8}$$

$$\sum_{m=1}^M y_{i,m,t} = 1 ; \forall i = 1, \dots, n \ \& \ t = 1, \dots, TH \tag{9}$$

$$\sum_{i=1}^n \sum_{m=1}^M r_{i,k} \leq R_k ; t = 1, \dots, T \ \& \ k = 1, \dots, K \tag{10}$$

$$S_i = \text{integer} \geq 0 \tag{11}$$

$$y_{i,m,t} = \text{bin} \tag{12}$$

The objective function of this model illustrates maximizing the net present value of an AON network. The first constraint in model is defined as determination of rate of online starts in each activity. Second Constraint ensures the feasibility of the activities precedence relations. Matrix Model "P" indicates all predecessors' relations among activities. The third constraint is a unique constraint to ensure inception of project from the beginning. The fourth constraint indicates make span of model should not exceed beyond the allowed time horizon. Fifth and Sixth constraints guarantee that each activity should select only one executive method in a way that this method should remain

constant in the long run of activity of project. The seventh constraint in the above mentioned model indicates this subject that amount of using limited renewable resources in whole time of project should be less than the total rate of availability of its resource. On the other hand, no more existing renewable resources' balance can be used in this project.

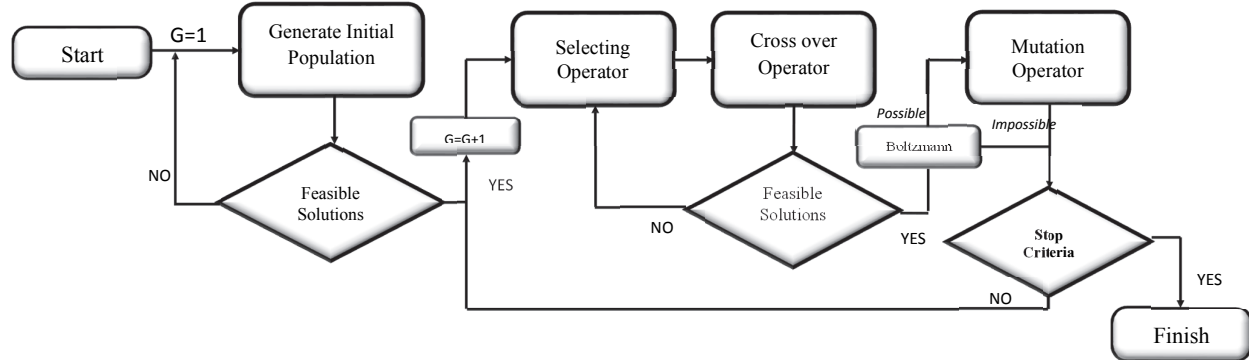
#### 4. The Proposed Genetic Algorithm Procedure

Elmaghraby et al. (1990) have dealt with the demonstration of NP-Complete in its RCPSP models. Zhou et al. (1998) also reported that Resource-constrained project scheduling problems with cash flows (RCPSPCF) are complex and combinatorial optimization problems and should be solved by heuristics. As mentioned in above, if MRCPS issues enjoy more than one resource, they will be considered strongly as part of NP-complete issues. Since nonlinear with exponential status is considered as target function of our desired model and with due observance to this fact that some of constraints enjoy nonlinear status like constraints of the first group, we can come to this conclusion that NP-complete is our desired model in this paper. There is also another reason for considering the mentioned model as NP-hard that is due to the number of the basic solutions that increases extremely while we increase the number of the variables. For example consider a simple model with 10 variables and 3 resources with 75 constraints that includes  $C = 85! / (10! 75!) = 3,129,162,672,636$  solutions as basic feasible and basic infeasible solutions together. Therefore, if the number of the variables increases extremely, optimal solution algorithms obviously cannot able to find the Optimum Basic solution. Consequently since MRCPS are dynamic in their natures, it seems necessary to use self-improve algorithms such as Genetic Algorithms (GA), Simulated Annealing (SA) and Neural Networks as the problem cannot be solved by optimal solution algorithms. As mentioned during last two decades, genetic algorithm has been widely used to solve MRCPS. Therefore, the research group decided to develop an efficient GA in this article to determine net present value of the project while resources are considered as pre-emptive .

In general, the main steps of our GA procedure are:

- Step 1) Create the initial population.
- Step 2) Compute the fitness value of each individual in the population.
- Step 3) Select a set of individuals to undergo genetic operators.
- Step 4) Evaluate the individuals created by the genetic operators.
- Step 5) Apply a replacement strategy to form the new generation.
- Step 6) If the stopping criteria are met then stop, otherwise go to Step 3.

Genetic algorithms (GA) are iterative search procedures, based on the biological process of natural selection and genetic inheritance, which maintain a population of a number of candidate members over many simulated generations. Hopefully, the good characteristics of the members will be retained over the generations, maximizing a determined fitness function (Fig. 1).



**Fig. 1.** Structure of the proposed GA to solve CMS model

The procedure starts by finding an initial feasible solution to the problem from an upper bound for each activity that meet the feasible priorities to each activity but not necessarily the maximum objective function value (or a set of activities that can make full scheduling). In this step we do not

pay attention to the time horizon of the project. The upper bound for the cycle can be found from the below equation:

$$UB = \sum_{i=1}^n \max d_{i,m}; \text{ for all } (i, m) \text{ belongs to Product\_sequence matrix} \tag{13}$$

#### 4.1. Population Size and Number of Generations

Generally metaheuristic algorithms quickly respond to small size or relaxed resource RCPSPs but while large scale problems are taken into account choosing appropriate population size for such algorithms plays essential rule to solve experiments. For this purpose a GA coding operator is developed which suggests the suitable, but not necessarily the best, population size according to the equation below:

$$\frac{\text{Max}(\text{renewable resource demand}_i)}{\text{Number of Generations}} ; \forall i \tag{14}$$

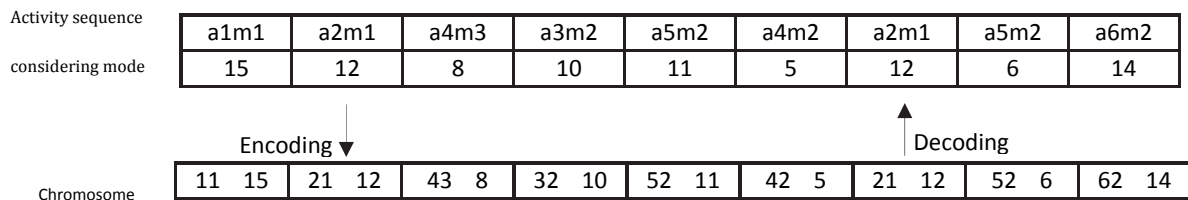
Eq. (10) consists on the largest frequency of the resource demands. The genetic algorithm maintains a collection (population) of solutions in each generation until the end of the searching process. Considering the complexity of experiments, number of generations is considered 20, 50 and 100 generations for small, medium and large scale experiments respectively.

#### 4.2. String Representation

The technique of GA requires a string representation scheme (chromosomes). The encoding of solutions in the proposed procedure is of type ‘one-to-one’ which means that each solution is represented exactly by one chromosome and the decoding of each chromosome results in exactly one solution for the problem. The chromosome is a string of length N where each element represents a Genetic operator of paired data of an activity priority based on activity priority list and machine position to which the corresponding task is assigned.

Fig. 2 shows the solution string which is based on product sequencing:

$$\text{Activity sequence considering executive mode} \begin{bmatrix} a_1m_1 & a_2m_2 & \dots & a_Mm_M \\ r_1 & r_2 & \dots & r_m \end{bmatrix} \tag{15}$$



**Fig. 2.** An example of a chromosome and the corresponding balancing solution

#### 4.3. Selection Operator and Fitness Functions

The selection operator is applied to select parent chromosomes from the population. A Monte Carlo selection technique is applied. Individual's selection procedure operates as follows:

- *Possible feasible function operator:* The GA procedure works to find a feasible solution, that is, a solution with S operators. The procedure is restarted with an upper cycle time to bind the operator movement over feasible solutions.
- *Possible length-string function operator:* Since it was included a constraint that excludes solutions with more than one operator, all solutions in the search space will have the same number of operators.

The fitness Function operator is considered as the objective function of the proposed mathematical model which is represented in Eq. (3).

#### 4.4. Crossover Operator

The genetic algorithm maintains a collection or population of solutions for each activity set throughout the search.

The main genetic operator is the crossover, which has the role to combine pieces of information from different individuals in the population. Two parents (P1 and P2) are chosen from the tournament list and a crossover point (cp), from Priority matrix is selected. The selection method is based on two rules respectively:

- *Weight Rule*: the gen will choose according to maximum weighted factor, here is cash flow, among parents' genes.
- *Remained path*: In this step if resource becomes over allocated, operator will find the much less important scheduled paths to make a split in the activities.
- *Resource availability*: if resource becomes over allocated, algorithm will find the next good gene (next activity) for allocation.

Note that split usually happens in more than one way network in a network diagram or when activity relations are start to start. If none of above happens, the mentioned place will leave blank. The proposed procedure, respecting to activity priorities, consists on scheduling more valuable activities sooner which cause gaining maximum net present value of the project, and filling the remained resources by other activities or even by replacing more weighted activities with current activities. GA will choose according to *Weight Rule*, *Remained path* and *Resource availability* sequentially, which determines the best activity string scheduling among the set of available tasks. In the other words, through child's chromosome string creation each of genes in string would be selected based on the maximum weighted factor among its parent's gens in their string. In this method, GA will support the idea of maximizing the net present value. In addition, if the place don't have enough resource to allocate, GA will find search through the before scheduled paths to find out whether there is any worth less path to make a delay on this path. In this manner GA utilizes the past information by simultaneously operating on a population of solutions. Fig. 3 typically shows how algorithm chooses next machine to minimize the total cost of the project. This heuristic also checks if the task to be assigned is over allocated to the machine capacity. In this manner the task will wait on the queue of the allocated machine or will allocate to another same type machine. In this way, the suggested GA can quickly locate high performance regions in each step in extremely large and complex search spaces of product sequences in order to maximize total NPV of the project.

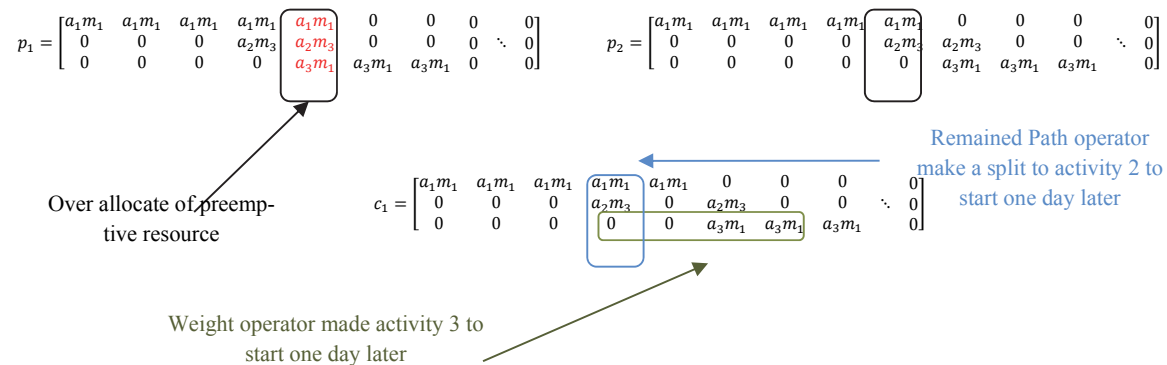


Fig. 3. Sample of Crossover Operator's Function

#### 4.5. Mutation Operator

The mutation operator is used to rearrange the structure of a chromosome which helps escaping from local optimum traps. In this Article, the swap mutation is used, which is selecting two chromosomes randomly and swapping their contents. The probability of mutation of a gene is based on statistical function and is a low probability in its nature as below:



$$Prob_i = \frac{CF(P_i)}{CF(P_i) + CF(P')} ; \forall (P_1, P_2) \quad (16)$$

Where  $P_1, P_2$  are chromosomes of random parent 1 and 2 and  $P'$  is new solution. The mutation rate is considered 0.1 as found in many researches in literature. This equation evaluated the objective function of the new population member. If the objective function of the new population is worse than its parents, there is still a small chance to consider it for further processing. Such idea helps escaping from local optimum traps.

#### 4.6. Stopping Criterion

The program is terminated when at least one of these conditions happen:

1. The maximum number of generations is reached.
2. Activities are scheduled in a way that there are no remain resources during time horizon which means there is no improvement in current solutions.
3. Time Horizon of the project finishes.

It is important to consider the steady conditions of designed algorithm as it is dynamic in its nature. For example, if two activities, which scheduled simultaneously and over allocated through their scheduled period, were bonded by a common successor, the program would never meet steady condition since it got stock into a loop:

$$|ES(A) - ES(B)| < |D(A) - D(B)| \quad (17)$$

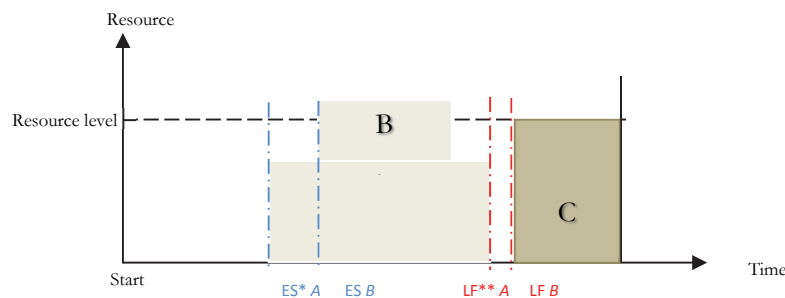


Fig. 4. A graphical sample of unsteady condition of MRCSP

\*ES=early start    \*\*LF=late finish    \*\*\*D=Duration

Fig. 4 shows that under mentioned condition, activity A and B will over allocate during the scheduling. This means that MRCSP system will stay in unsteady state or may come into transient state but it may never pass it.

## 5. Results and Discussion

To examine and verify the impact of pre-emptive resources to net present value of a resource-constraint scheduling problem, 3 problems (in small, medium and large scales) will be illustrated in detail at first. Then a number of large scale experiments will be solved to examine the effectiveness of the proposed method. The problems will be solved using MATLAB R2009® which is installed on a Core i7 laptop that is supported by 4 Mb RAM. Each problem is allowed the maximum time based on upper bound introduced in Equation 13. Noted that the proposed model is a nonlinear and non-concave that cannot be solved within reasonable time (say, one year) optimally. Thus, we consider a feasible interval for the optimal objective function value (OFV). At such a point, the user may choose to interrupt the solver and go with the current best solution in the interest of saving on additional computation time.

5.1 Small Size Experiment

A comparative evaluation of the proposed approach is made using bench-mark numerical examples. The first example input data set is related to the simple problem with 8 activity, 2 executive modes for each activities, 1 types of pre-emptive resource with 8 capacity level, 4 generation in each period, time-horizon of each period shall not exceed 42 days (Table 1). As mentioned before, the population-size is based on equation 29 for small case studies and 50 for large ones which will honor to its initial size during the time-horizon. Rests of the information are as follow ( $\alpha = 0.3$ ):

8|2|1|42|1|4|8

**Table 1**  
Input Data for Numerical Example 1

	Task Name								Sequence Information							
	A	B	C	D	E	F	G	H	A	B	C	D	E	F	G	
Activity Mode's Duration	4	3	8	3	6	7	6	4	A	0	0	0	0	0	0	0
Resource_usage	5	2	9	4	4	6	7	4	B	A	0	0	0	0	0	0
Cash_flow	3	6	2	3	2	6	5	1	C	0	B	0	0	0	0	0
	90	140	200	110	50	144	148	80	D	0	B	0	0	0	0	0
	100	130	210	100	48	120	122	70	E	0	0	0	D	0	0	0
									F	0	0	C	0	0	0	0
									G	0	0	0	0	E	0	0
									H	0	0	0	0	0	F	G

In the first step, GA has found an upper bond which guarantees the sequence relations. The upper bond also respect to pre-emptive resource capacity in its nature. It is observed that time horizon (upper bond) for this problem is found as 42 days. Afterward, GA seeks to find out the fragment available resources (gaps) which reveal the availability for improvements. Then, it finds the activities that can fill such gaps in a way that net present value of the project maximizes. In this approach, backward movements of activities will guaranty reducing make span of the project. In addition, respect to the activity priorities, if there is possibility of gaining higher NPV by making a split to an activity, GA will uses such occasions. Tables 2 indicates that using GA for the proposed model reduces the make span of the project by using free available resources in a backward movement of activities. For instance, activity D was moved back 8 days to fill the gap between days 9 and 12 which caused backward movement for all its successor activities (E, F, G and H respectively). Assuming higher cash flow for activity number G (rather than number F) causes a split within activity F as soon as activity E (its predecessor) finished. Consequently, Applying GA for the typical MRCSP with pre-emptive resources increased the NPV to around 5154 while reduced make span of the project for 10 days in this example. The Table 2 shows the amount of the net present value of the project during GA iterations:

**Table 2**  
Observed Result of Numerical Example 1

Solution String (Activity scheduling)								Start Date								NPV	MAKE SPAN	r1*
A	B	C-D	C-E	E-F(1)	G	F(2)	H	A	B	C	D	E	F	G	H			
A	B	C-D	C-E	E-F(1)	G	F(2)	H	1	6	9	9	13	17-35	19	29	5154.431	32	113

\* r1 is the amount of unused resource (considering solution string).

In addition, the experimental results show that after 3 generations, the amount usage of 'remains available resource of gaps' is increased and consequently net present value of the project change to maximum point (Fig. 5). In the presented model,  $\alpha$  is considered as an attractive rate encouraging activities to set backward movements in order to get maximum utilization of using remained resource in the previous gaps. Therefore, it is essential to survey the impact of  $\alpha$  in proposed MRCSP model with using different rates (Fig. 6). Figure above clearly shows that MRCSP model obeys rather integration at high rates of  $\alpha$  while other other problem's parameters are not changes. it means that the more  $\alpha$  rate considered, the more backward polarity in results can be achieved. It is noticeable that  $\alpha$  can consider as minimum attractive rate (MARR) for preemptive supported activities in

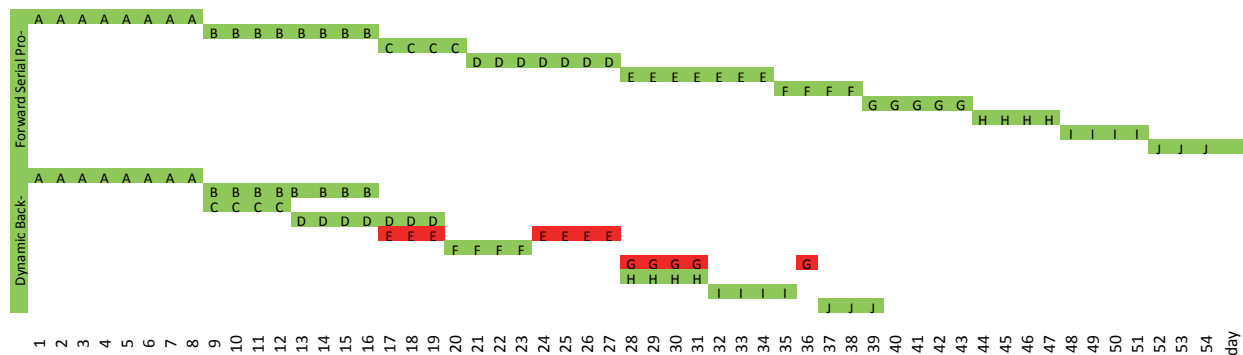


During the problem solving, in step 33, GA makes split in activity E (in day 20), where GA find out for the first time the possibility of splitting the activity, in order to start activity f. Since, such splits would increase the profit of NPV, all the coming members follow this gene as a part of their string solution in the next generations. Results show that using proposed backward procedure causes significantly increasing of the NPV by moving back the activities (Table 4). As seen the proposed method improves the amount of resource usage while it effectively reduce the make span of the project.

**Table 4**  
Observed Results of Numerical Example 2

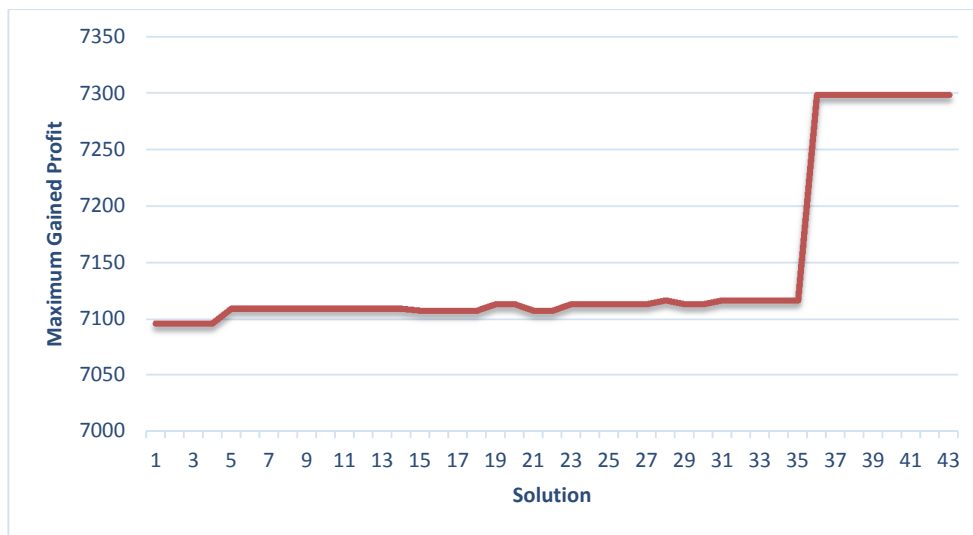
Solution String (Activity scheduling)										Start Date										NPV	MAKE SPAN	r1*	r2
										A	B	C	D	E	F	G	H	I	J				
A	B-C	B-D	D-E(1)	F	E(2)	G(1)-H	I	G(2)	J	1	9	9	13	17-31	27	28-43	28	32	37	7298.221	39	352	232

The Gantt of modified activities is shown in Fig. 7, which is transformed from linguistic variables; reveals that re-scheduling project with applied GA will cause 15 days saving while increasing NPV and honors to resource level. The dedicating pre-emptive resource to activities allows GA to split the activity E in order to maximize the gained cash flow as there is opportunity to do it due to remained resources between day 12 and 16.



**Fig. 7.** Comparing outcomes of backward approach and upper bond for numerical example 2

The Fig. 8 shows that GA stood at significant high level of integration for creating solutions due to using crossover operators through generation. Such ability will allow GA to perform more effectively in the case of complex problems.



**Fig. 8.** Solution Integrity for proposed MRCSP model (numerical example 2)

### 5.3 Large Scale Experiment

In this section a more complicated example which contains 17 task and 3 pre-emptive resources. It is noticeable that even increasing a few tasks in proposed model would dramatically rise up the complexity of the model. The amount allowed for pre-emptive resources are 27, 20 and 14 per iteration, respectively. Rest of the information is given in Table 5 including activity durations, expected profit, activity relations and the amount of resource usage for each activity (17\2\3\71\1\10\50).

**Table 5**  
Input Data for Numerical Example 3

		Task																	Sequence Information																		
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q		
Activity Mode's Duration	1	4	2	10	4	2	5	13	3	3	5	4	3	2	1	3	1	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	2	5	3	11	3	3	6	17	4	2	3	3	4	3	2	3	2	B	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Resource Usage	20	13	12	18	22	6	11	7	2	10	10	8	2	11	9	10	18	C	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	10	8	4	9	12	2	6	3	2	8	5	5	2	4	8	7	12	D	0	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Cash flow	100	30	80	10	100	40	40	100	30	170	70	150	40	80	200	60	20	E	0	0	C	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	110	32	40	9	91	35	30	60	40	158	50	124	30	75	180	74	12	F	0	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
																			G	0	0	0	0	0	F	0	0	0	0	0	0	0	0	0	0	0	0
																			H	0	0	0	0	0	F	0	0	0	0	0	0	0	0	0	0	0	0
																			I	0	0	0	0	0	0	G	0	0	0	0	0	0	0	0	0	0	0
																			J	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
																			K	0	0	0	0	0	0	0	0	0	J	0	0	0	0	0	0	0	0
																			L	0	0	0	0	0	0	0	H	0	0	0	0	0	0	0	0	0	0
																			M	0	0	0	0	0	0	0	0	0	0	K	0	0	0	0	0	0	0
																			N	0	0	0	0	0	0	0	0	0	0	0	0	0	M	0	0	0	0
																			O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	N	0	0	0
																			P	0	0	0	D	0	0	0	0	0	0	0	0	0	M	0	0	0	0
																			Q	0	0	0	0	E	0	0	0	0	0	0	0	L	M	O	O	P	0

The upper bond solution set is replaced and updated for backward movements if a better solution is obtained (Table 6). The GA consists searching until any stop conditions occurs.

**Table 6**  
Most observed Solution String for example 3

Solution String (Activity scheduling)		Start Date																	NPV	MAKE SPAN	r1*	r2	r3													
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q																		
A <sup>B(1)-</sup> C	E <sup>D(1)-</sup> F	G <sup>H(1)</sup>	D(2)-H(1)-I	J	D(3)-H(2)-K	H(2)-M(1)	D(4)-L	M(2)	N-P(1)	O	P(2)	Q	-	1	2-7	2	9-16-22-27	4	9	11	11-22	16	19	22	27	25-30	31	33	33-34	35	4306.28	2	35	294	168	135

Table 6 presents the best solution strings which created by GA through solving example 3. It reveals that activity splitting will affects the project NPV significantly. Usage for pre-emptible resource R1 (27) occurs from day 4 to 6, while the backward movement of activity C makes splitting activity B which cause increasing NPV from 4267 to 4306. Preemption rule for resource R2 (20) occurs for activity J and L from day 19 to 21 and day 27 to 29 since they have priority rather than H and M, respectively. Constraint in availability R (3) causes splitting activity P in order to gain more NPV (by making backward movement for activity O). In addition, it should be noticed that, respecting to activity relations, low cash-flow for activity D will cause several times of activity splitting in order to back moving of other activities. The Gantt chart shows that proposed backward approach will effectively fills resources gaps in modified plan. Such declining also caused significant reduction in make span of project (26 days). The preemption rule also allows GA to maximize NPV by splitting activities B, D, H, K and P (Fig. 9). Furthermore, the results of setting tight resource constraints will affect the gained profit of the project by reducing daily resource usage in some iteration. Therefore, the proposed approach successfully integrates both time and resource constraints and generates an adequate schedule plan to maximize NPV of the project.

Validating the proposed Algorithm:

In continue performance of the proposed method for maximizing NPV in large scale problems are examined using 12 series of experiments with 18, 20, 30 and 50 variables. For evaluating the efficiency of proposed model each example is solved under two conditions where all the criteria are considered the same but resource availability. The results are then compared with results of forward serial programming method (Table 7).

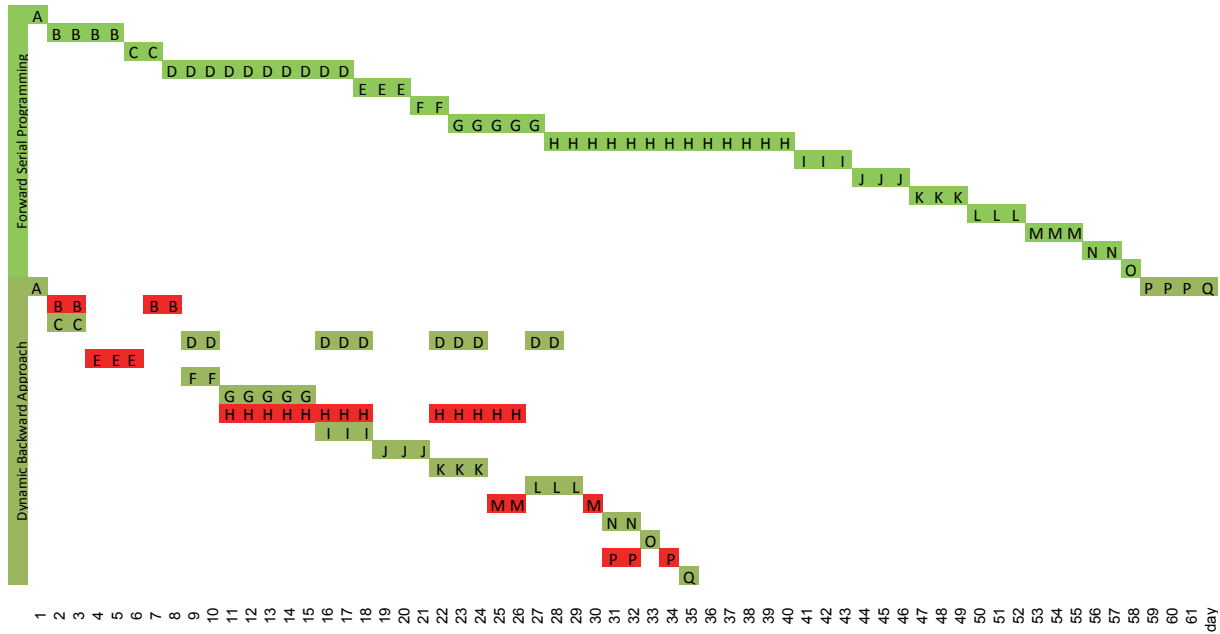


Fig. 9. Comparing Outcomes of Backward Approach and Upper Bond for Numerical Example 3

Table 7  
Numerical examples for the proposed backward method

No.	Problem	$\alpha$	Available Pre-emptive Resources	Pop	Gen	U.B	Precedence Matrix	FSP*	GA	$\Delta NPV$	RMS**
1	18/3/3	0.3	40/45/45	50	50	99	A B C D E F G H I	2791.6	2774.3	17.3	59
2	18/3/3	0.4	40/45/45	50	50	99	A B C D E F G H I	2811.7	2789.2	22.5	59
3	18/3/3	0.5	40/45/45	50	50	99	A B C D E	2833.8	2804.3	<b>29.5</b>	59
4	20/4/3	0.3	25/30/25	50	50	105	A B C D E F G H I J	2935	2916.7	18.3	38
5	20/4/3	0.4	25/30/25	50	50	105	A B C D E F G H I J	2958.3	2932.6	25.7	58
6	20/4/3	0.5	25/30/25	50	50	105	A B C D E	2984	2950.7	<b>33.3</b>	42
7	30/4/2	0.3	40/30	100	100	150	A B C D E F G H I J	4194.9	4166.9	28	61
8	30/4/2	0.4	40/30	100	100	150	A B C D E F G H I J	4227.8	4188.5	39.3	26
9	30/4/2	0.5	40/30	100	100	150	A B C D E	4264.4	4211.4	<b>53</b>	26
10	50/4/5	0.3	80/90/70/80/70	100	100	255	A B C D E	7671.4	7602.7	68.7	204
11	50/4/5	0.4	80/90/70/80/70	100	100	255	F G H I J K L M N O	7735.4	7640.7	94.7	204
12	50/4/5	0.5	80/90/70/80/70	100	100	255	L M N O	7797.9	7679.1	<b>118.8</b>	203

Results indicate the Proposed GA can effectively maximize the NPV of the MRCPSP-DSF problem by using remained resources in backward rescheduling approach. In Table 8 the schedules of the 12 experiments are shown. Noted that those activities that are shown in parenthesis are decided being taken apart in order to achieve more NPV through the calendar.

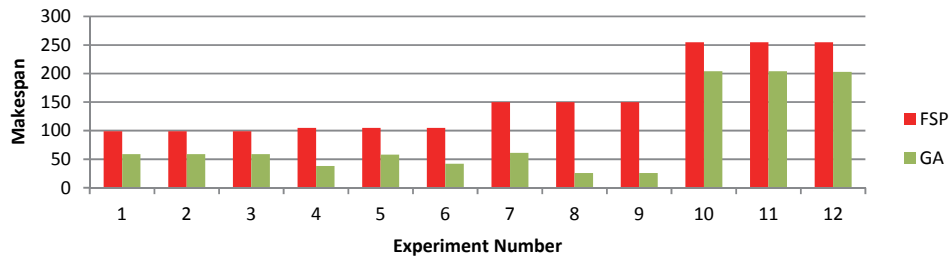


Fig. 10. Comparing the Makespan between FSP and the proposed GA

Fig. 10 shows that, in the solved experiments, the backward method can always result less makespan than FSP. In order to evaluate the performance of the proposed method in minimizing makespan of the projects, we used the performance measure called Makespan improvement that was proposed by (Buddhakulsomsiri et al., 2006):

$$Makespan\ Improvement = \frac{Makespan\ without\ splitting - makespan\ with\ splitting}{makespan\ without\ splitting} \tag{18}$$

Using makespan improvement ratio, a supreme improvement can be seen while activity splitting is allowed which means that GA can effectively use unfilled resource capacities respecting to activity priorities. Note that for the first problem, since both states of makespan with and without activity splitting reported the same structure, the makespan improvement value reported 0.

**Table 9**  
Results of Comparing the proposed GA with FSP Method

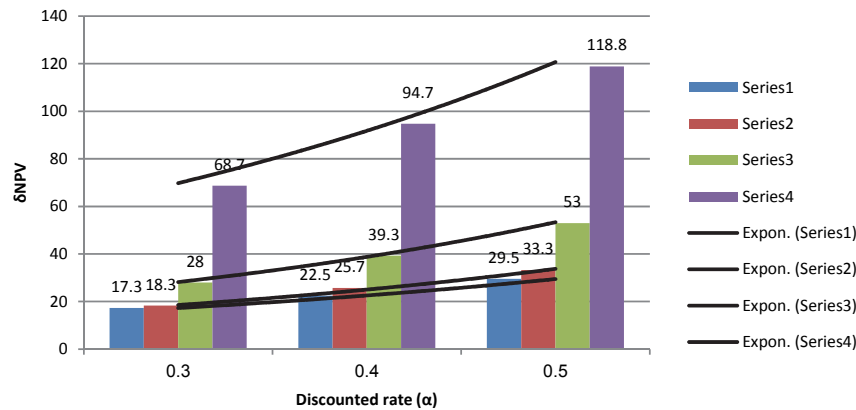
No.	A/R	Resource capacity	M.W.S	M.W(O).S	Makespan Improvement
1	6/3	8/10/8	14	14	0.00
3	10/2	10/10	21	32	0.34
5	15/2	20/11	29	54	0.46
7	18/3	15/20/10	57	79	0.278
9	20/3	15/20/10	53	91	0.417
11	30/2	20/25	55	144	0.618
13	50/5	20/30/20/30/40	79	128	0.383

A: Activity R: Resource  
M.W.S: Makespan with activity splitting  
M.W(O).S: Makespan without activity splitting

**Table 8**  
Solution String Presentation for Solved Case studies

No.	Exam-	Solution String
1	18/3/3	A-B-C-D-E-F-G-I-H-K-J-N-L-M-P-Q-O-R
2	18/3/3	A-B-C-D-E-F-G-I-H-K-J-N-L-M-P-Q-O-R
3	18/3/3	A-B-C-D-E-F-G-I-H-K-J-N-L-M-P-Q-O-R
4	20/4/3	A-B-C-D-G-E-F-H-M-I(1)-J-K-L-N-I(2)-O-P-Q
5	20/4/3	A-B-C-D-E-G-H-I-F(1)-J-M-N-O-P-F(2)-S-R-Q-K-L-T
6	20/4/3	A-B-C-D-E-G(1)-H-I-G(2)-F(1)-J-M-N-O-P-F(2)-S-R-Q-K-L-T
7	30/4/2	A-B-C-D-E-G-E-F-I(1)-J-O-L-M-N-P-I(2)-T-V-U-Q-R-S-Z-X-Y-W-AA(1)AB-AA(2)-AC-AD
8	30/4/2	A-B-C-D-E-H(1)-F-G-K-H(2)-I-J-M-N-P-T-S-O-Q-R-W-V-U-Z-AA-AC-T-Y-R-X-AB-AD
9	30/4/2	A-B-C-D-E-H(1)-F-G-K-H(2)-I-J-M-N-P-T-S-O-Q-R-W-V-U-Z-AA-AC-T-Y-R-X-AB-AD
10	50/4/5	A-B-C-D-E-F-G-H-I-J-L-M-K-N-O-R-S-T-U-V-P-Q-W-AC-X-Y-AB-AG-AD-AE-AF-Z-AA-AH-AK-AL-AI-AJ-AM-AP-AO-AN-AQ-AR-AT-AU-AS-AV-AW-AX
11	50/4/5	A-B-C-D-E-F-G-H-I-J-L-M-K-N-O-R-S-T-U-V-P-Q-W-AC-X-Y-AB-AG-AD-AE-AF-Z-AA-AH-AK-AL-AI-AJ-AM-AP-AO-AN-AQ-AR-AT-AU-AS-AV-AW-AX
12	50/4/5	A-B-C-D-I(1)-F-G-E-H-I(2)-L-M-I(3)-J-P(1)-Q-V-K-R-S(1)-N-T-A-B-O-U(1)-X-Y(1)-AE(1)-P(2)-Q(2)-Z-S(2)-AA-U(2)-AC-V(2)-A-I(1)-W-AC(2)-AF-AI(2)-Y(2)-AG-AC(2)-AD-AE(2)-AL-AM-AP-AQ-AR(1)-AI-AN(1)-AR(2)-AJ-AU-AT(2)-AN(2)-AK-AO(1)-AS(1)-AO(2)-AR-AS(2)-AV-AT(4)-AU(2)-AW(1)-AV(2)-AW(2)-AX

Results also indicate that by increasing the discounted rate and the number of activities (dimension of the problem) the angle of slope of the NPV is increased (Fig. 11).



**Fig. 11.** Comparing the effects of alpha rate and number of activities in increasing NPV

## 6. Conclusion

As far as the issue of maximizing net present value of project is concerned, there exists inclination to use available resources remaining and also vacant remaining times in previous. By using these opportunities, more efforts have been taken with the aim of adding present value of activities coupled with present value of project while Pre-emptive resources, regarding to their identity in establishing cease on activities, will cause longevity of time span of project implementation and finally, reducing present value of project. By presenting a dynamic backward approach, an appropriate and logical solution will be boosted. Using the proposed procedure has caused noticeable rise in remaining resources usage in the long run of project implementation period.

## Acknowledgements

The authors would like to thank Dr. Mohammad Gholami (Post-doctoral fellow; University of Calgary-Abb.CA) for his positive comments during the writing of this manuscript.

## References

- Abbasi, B., Shadrokh, S., & Arkat, J. (2006). Bi-objective resource-constrained project scheduling with robustness and makespan criteria. *Applied Mathematics and Computation*, 180(1), 146-152. doi:<https://doi.org/10.1016/j.amc.2005.11.160>
- Ballestín, F., Valls, V., & Quintanilla, S. (2008). Pre-emption in resource-constrained project scheduling. *European Journal of Operational Research*, 189(3), 1136-1152. doi:<https://doi.org/10.1016/j.ejor.2006.07.052>
- Buddhakulsomsiri, J., & Kim, D. S. (2006). Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting. *European Journal of Operational Research*, 175(1), 279-295. doi:<https://doi.org/10.1016/j.ejor.2005.04.030>
- Castejón-Limas, M., Ordieres-Meré, J., González-Marcos, A., & González-Castro, V. (2011). Effort estimates through project complexity. *Annals of Operations research*, 186(1), 395-406. doi:<https://doi.org/10.1007/s10479-010-0776-0>
- Chen, J., & Askin, R. G. (2009). Project selection, scheduling and resource allocation with time dependent returns. *European Journal of Operational Research*, 193(1), 23-34. doi:<https://doi.org/10.1016/j.ejor.2007.10.040>
- Chetourou, H., & Haouari, M. (2008). A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. *Computers & Industrial Engineering*, 55(1), 183-194. doi:<https://doi.org/10.1016/j.cie.2007.11.017>



- Damay, J., Quilliot, A., & Sanlaville, E. (2007). Linear programming based algorithms for preemptive and non-preemptive RCPSP. *European Journal of Operational Research*, 182(3), 1012-1022. doi:<https://doi.org/10.1016/j.ejor.2006.09.052>
- Delgoshaei, A., Ariffin, M., Baharudin, B., & Leman, Z. (2016a). A new method for decreasing cell-load variation in dynamic cellular manufacturing systems. *International Journal of Industrial Engineering Computations*, 7(1), 83-110.
- Delgoshaei, A., Ariffin, M. K. A., & Ali, A. (2017). A multi-period scheduling method for trading-off between skilled-workers allocation and outsource service usage in dynamic CMS. *International Journal of Production Research*, 55(4), 997-1039.
- Delgoshaei, A., Ariffin, M. K. A. M., Leman, Z., Baharudin, B. H. T. B., & Gomes, C. (2016b). Review of evolution of cellular manufacturing system's approaches: Material transferring models. *International Journal of Precision Engineering and Manufacturing*, 17(1), 131-149.
- Elmaghraby, S. E., & Herroelen, W. S. (1990). The scheduling of activities to maximize the net present value of projects. *European Journal of Operational Research*, 49(1), 35-49.
- Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1), 1-14. doi:<https://doi.org/10.1016/j.ejor.2009.11.005>
- Hendricks, M. H., Voeten, B., & Kroep, L. H. (2002). Human resource allocation in a multiproject research and development environment. *Managing Multiple Projects. Planning, Scheduling, and Allocating Resources for Competitive Advantage*.
- Jarboui, B., Damak, N., Siarry, P., & Rebai, A. (2008). A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation*, 195(1), 299-308.
- Ke, H., & Liu, B. (2010). Fuzzy project scheduling problem and its hybrid intelligent algorithm. *Applied Mathematical Modelling*, 34(2), 301-308.
- Kim, K., Yun, Y., Yoon, J., Gen, M., & Yamazaki, G. (2005). Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling. *Computers in Industry*, 56(2), 143-160. doi:<https://doi.org/10.1016/j.compind.2004.06.006>
- Kreter, S., Rieck, J., & 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*, 251(2), 387-403. doi:<https://doi.org/10.1016/j.ejor.2015.11.021>
- 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*, 28(6), 609-618. doi:<https://doi.org/10.1016/j.ijproman.2009.10.001>
- Lin, C.-M., & Gen, M. (2008). Multi-criteria human resource allocation for solving multistage combinatorial optimization problems using multiobjective hybrid genetic algorithm. *Expert Systems with Applications*, 34(4), 2480-2490. doi:<https://doi.org/10.1016/j.eswa.2007.04.016>
- Mika, M., Waligóra, G., & Węglarz, J. (2005). Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models. *European Journal of Operational Research*, 164(3), 639-668. doi:<https://doi.org/10.1016/j.ejor.2003.10.053>
- Naber, A., & Kolisch, R. (2014). MIP models for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, 239(2), 335-348. doi:<https://doi.org/10.1016/j.ejor.2014.05.036>
- Papke-Shields, K. E., & Boyer-Wright, K. M. (2017). Strategic planning characteristics applied to project management. *International Journal of Project Management*, 35(2), 169-179.
- Pérez, E., Posada, M., & Lorenzana, A. (2016). Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms. *Soft Computing*, 20(5), 1879-1896.
- Rabbani, M., Ravanbakhsh, M., Farrokhi-Asl, H., & Taheri, M. (2017). Using metaheuristic algorithms for solving a hub location problem: application in passive optical network planning. *International Journal of Supply and Operations Management*, 4(1), 0-0.

- Seifi, M., & 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*, 21(4), 347-360.
- Sharon, A., & Dori, D. (2015). A Project-product model-based approach to planning work breakdown structures of complex system projects. *IEEE Systems Journal*, 9(2), 366-376.
- Vaez, P. (2017). A New Mathematical Model for Simultaneous Lot-sizing and Production Scheduling Problems Considering Earliness/Tardiness Penalties and Setup Costs. *International Journal of Supply and Operations Management*, 4(2), 167-179.
- Van de Vonder, S., Demeulemeester, E., Herroelen, W., & Leus, R. (2005). The use of buffers in project management: The trade-off between stability and makespan. *International Journal of Production Economics*, 97(2), 227-240.
- Van de Vonder, S., Demeulemeester, E., Herroelen, W., & Leus, R. (2006). The trade-off between stability and makespan in resource-constrained project scheduling. *International Journal of Production Research*, 44(2), 215-236.
- Van Peteghem, V., & Vanhoucke, M. (2010). A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, 201(2), 409-418.
- Vanhoucke, M., & Debels, D. (2008). The impact of various activity assumptions on the lead time and resource utilization of resource-constrained projects. *Computers & Industrial Engineering*, 54(1), 140-154. doi:<https://doi.org/10.1016/j.cie.2007.07.001>
- Ward, S., & Chapman, C. (2003). Transforming project risk management into project uncertainty management. *International Journal of Project Management*, 21(2), 97-105. doi:[https://doi.org/10.1016/S0263-7863\(01\)00080-1](https://doi.org/10.1016/S0263-7863(01)00080-1)
- Węglarz, J., Józefowska, J., Mika, M., & Waligóra, G. (2011). Project scheduling with finite or infinite number of activity processing modes—A survey. *European Journal of Operational Research*, 208(3), 177-205. doi:<https://doi.org/10.1016/j.ejor.2010.03.037>
- Yan, L., Jinsong, B., Xiaofeng, H., & Ye, J. (2009). A heuristic project scheduling approach for quick response to maritime disaster rescue. *International Journal of Project Management*, 27(6), 620-628. doi:[doi.org/10.1016/j.ijproman.2008.10.001](https://doi.org/10.1016/j.ijproman.2008.10.001)
- Yu, Wang, S., Wen, F., & Lai, K. K. (2012). Genetic algorithm-based multi-criteria project portfolio selection. *Annals of Operations Research*, 197(1), 71-86.
- Zhou, M., & Askin, R. G. (1998). Formation of general GT cells: an operation-based approach. *Computers & industrial engineering*, 34(1), 147-157.
- ZIAEE, M. (2017). Modeling and solving the distributed and flexible job shop scheduling problem with WIPs supply planning and bounded processing times.

