InternationalJournal of Industrial Engineering and Management (IJIEM),Vol.5 No 3, 2014, pp. 151-158
Available online at www.iim.ftn.uns.ac.rs/ijiem_journal.php
ISSN 2217-2661

UDK:658.5.012.2

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

Aidin Delgoshaei
PhD Candidate, University Putra Malaysia, Department of Mechanical and Manufacturing Engineering, University Putra Malaysia, 43400 UPM, Serdang, Kuala Lumpur, Malaysia, delgoshaei.aidin@gmail.com

Mohd Khairol Ariffin
Associated Professor, University Putra Malaysia, Department of Mechanical and Manufacturing Engineering, University Putra Malaysia, 43400 UPM, Serdang, Kuala Lumpur, Malaysia, khairol@eng.upm.edu.my

B. T. Hang Tuah Bin Baharudin
Associated Professor, University Putra Malaysia, Department of Mechanical and Manufacturing Engineering, University Putra Malaysia, 43400 UPM, Serdang, Kuala Lumpur, Malaysia, tuah@eng.upm.edu.my

Zulkiflle Leman
Associated Professor, University Putra Malaysia, Department of Mechanical and Manufacturing Engineering, University Putra Malaysia, 43400 UPM, Serdang, Kuala Lumpur, Malaysia, zleman@eng.upm.edu.my

Received (25.07.2014); Revised (10.09.2014); Accepted (01.10.2014)

Abstract

In this article, a backward approach for maximizing net present value (NPV) in multi-mode resource constrained project scheduling problem while assuming discounted positive cash flows (MRCPSP-DC) is proposed. 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. A simulated annealing algorithm is developed for solving experimental cases with 50 variables and results are compared with forward serial programming method. The remarkable results demonstrate that the proposed method can effectively improve NPV in MRCPSP-DC while activity splitting is allowed.

Key words: *Project Scheduling, Pre-emptive, Resource-Constrained, Simulated Annealing*

1. INTRODUCTION

Resource-constrained Project scheduling problem (RCPSP) was firstly developed with the aim of reducing make span of the project Kelley in 1963[18]. Multi-mode resource constraint project scheduling problems (MRCPSP) are distinctive resource-constraint problems where each activity can be carried out via different modes (regarding to technologies or material etc.). As consequence, the execution period (activity duration), resource requirement level and even the cash flow may be vary form a mode to another. The MPRCPSP problem was initially developed for minimizing the project make span and was proved to be a NP-hard problem [21]. Węglarz et al. provided a wide research on literature of the multimode project scheduling[30]. One of the most important issues in MRCPSP studies is financial issues which can be considered in two ways of positive or negative cash flows. Positive cash flows are supposed to earn as scheduled milestones. Despite, negative cash flows are referred to those expenses which must be spent for making positive cash flows (as human resource salary, equipment and machinery purchasing and maintenance costs, raw material providing etc.). In such models, cash flow can be influenced by activity due date, duration, resource requirements and also payment method which will affect on activity execution mode as well.

Kolisch & Drexl[21] found that MRCPSP is NP-hard if more than one resource is considered. To come up with such problem, many heuristics and met heuristics approaches are applied so far. Yan et al.[31] applied heuristics to solve project scheduling problems for quick response to maritime disaster rescue. Laslo[22] presented an integrated method using simulation for resource planning and scheduling to minimize scheduling dependent expenses. But among all met heuristic algorithms, Genetic algorithm (GA) and Simulated Annealing (SA) have been used more times to solve MRCPSP ([26]; [27]; [12]; [1]; [24]; [23]). [19] proposed a hybrid GA and fuzzy logic controller (FLC-

HGA) to solve the resource-constrained multiple project scheduling problem (RC-MPSP). Their objectives were minimizing total project time and total tardiness penalty. Ke & Liu [17]used hybrid fuzzy set and GA to minimize total cost with completion time limits (see also [5]; [13]). Jarboui et al. [16] used particle swarm optimization (PSO) to show the effectiveness of PSO for solving MRCPSPs (see also [33]).

Maximizing net present value (NPV) of projects is considered as an important objectives in financial studies of scheduling problems. The idea of maximizing NPV was first proposed by Russell[29]. The proposed model was nonlinear without taking limitations of resources. They assumed activity on art (AOA) to present network charts. Afterward, Grinold[11] converted the model proposed by Russell into linear model and developed two optimal finder algorithms considering fixed and variable due date of project. Elmaghraby & Herroelen [9] proposed an optimal-finder algorithm which includes resource constraints for maximizing NPV. Etgar et al. [10]showed that resources beyond time limit can have significant effect on make span of project Meanwhile, De Reyck [7] offered an algorithm based on which both positive and negative cash flows had been considered. A lower and upper bound were considered for each activities where coupled with limited resources. Icmeli et al. [15] discussed that adding resources limitations caused turning model into a non-poly nominal model which cannot been solved easily by optimizing algorithms. Then, they considered discounted rate in the proposed a model a way that more cash flows will be earned in case of completing an activity in shorter period (RCPSPDCF).. Afterward, many researchers tried their utmost effort with the aim of solving the problem of maximizing NPV while discount rate are taken into consideration. Baroum & Patterson [3] solved a RCPSPDCF model with 50 variables where only positive cash flows were considered.

Afterward, Icmeli & Erenguc [14] used Tabu search (TS) algorithm in solving RCPSPDCF problem. They set penalty for activities later than the due date. Yang et al. [32] statistical programming rules for solving RCPSPDCF problems with considering positive cash flows. Moreover, Zhu & Padman [34] used TS for solving RCPSPDCF problems. Mika et al. [25] presented a model with the aim of maximizing NPV of project with taking discounted rate and also both renewable/non-renewable resources. They used hybrid of SA and TS to solve the problems.

During 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. Demeulemeester & Herroelen [8] presented an optimal solution for RCPSP while they considered pre-emptive resources in their model. Buddhakulsomsiri & Kim [4] discussed that considering pre-emption resources is vital while studying make span of the project. Damay et al.[6] applied linear programming algorithms for pre-emptive RCPSP studies while Ballestín et al. [2]proposed heuristic for solving pre-emptive RCPSP. Currently, Peteghem & Vanhoucke [28] used GA to minimize make span of MRCPSP while they considered pre-emptive resources which allow activity splitting through their research.

To the best knowledge of authors, the idea maximizing NPV in RCPSPs while activity split ability are supposed and pre-emptive resources are taken into consideration, is less developed. To overcome such shortcoming, a multi-mode resource constrained scheduling problem with discounted cash flows (MRCPSP-DCF) is developed to study impacts of pre-emptive resources on maximizing NPV.

2. THE PROPOSED MODEL

2.1 Subscript

Subscripts used in the model are considered as follows:

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

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

t=available time horizon for production (t=1, 2 … T)

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

2.2 Parameters

The list of parameters and notations is as follows:

Resource_Capacity: illustrates available resource in sub periods:

$$[\mathrm{RC}_1 \quad ... \quad \mathrm{RC}_k]_{1*k} \qquad (1)$$

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

$$\mathrm{RC}_k = \mathrm{R}_k - \textstyle\sum_{i=1}^{n} r_{(i,k)} * y_{(i,m,t)} \quad ;\forall\ (k \in K)\&(t \in \mathrm{TH}) \qquad (2)$$

Activity time: shows duration of each activity considering different execution modes.

$$\begin{bmatrix} 11 & \cdots & 1n \\ \vdots & \ddots & \vdots \\ m1 & \cdots & mn \end{bmatrix}_{m*n} \qquad (3)$$

Activity sequence matrix is used in mathematical programming to show precedence relations between activities.

$$\begin{bmatrix} 11 & \cdots & 1n \\ \vdots & \ddots & \vdots \\ n1 & \cdots & in \end{bmatrix}_{n*n} \qquad (4)$$

$\mathrm{CF(i, m)}$: positive cash flow of activity i while it performs in mode m

$\mathrm{r(i, k)}$: usage amount of resource type k for activity i (6)

$\mathrm{R(k)}$: available level of resource type k (7)

$\mathrm{D(i, m)}$: duration of Activity i while it performs in mode m (8)

TH: time horizon of the project (9)

$\alpha =$ discounted rate (10)

$\mathrm{P} =$ time slots (11)

2.3 Decision Variables

$$y(i, m, t) = \begin{cases} 1 & \text{if activity i performs on mode m during period t} \\ 0 & \text{otherwise} \end{cases} \qquad (12)$$

$$S(i) = \text{start time of activity i} \qquad (13)$$

2.4 Mathematical Model

The mathematical model is now written as follows:

$$\text{Max: } \sum_{p=1}^{P}\sum_{i=1}^{n}\sum_{t=s_1}^{\min(s_i+d_i,m,tp)} y_{(i,m,t)} \cdot (CF_{i,m}.d_{i,m}).e^{\frac{\alpha}{(s_i+d_{i,m})}} \quad (14)$$

$S.T$:

$$S_i : \left\{ \sum_{t=ES_i}^{LF_i} y_{(i,m,t)} - \sum_{t=ES_i}^{LF_i} y_{(i,m,t+1)} = 1 \middle| y_{(i,m,t-1)} = 0 \, , y_{(i,m,t=ES_i)} \right\} ; \forall\ (i,j) \in P_i \quad (15)$$

$$S_i + d_{i,m} < S_j \ ; \forall \ (i,j) \in P_i \quad (16)$$

$$S_1 = 0 \quad (17)$$

$$S_n \leq TH - d_{n,m} \quad (18)$$

$$\sum_{t=s_i}^{LF_i} y_{i,m,t} = d_{i,m} \cdot y_{(i,m,st)} \ ; \forall\ i = 1,..,n \ \& \ m = 1,..,M \quad (19)$$

$$\sum_{m=1}^{M} y_{i,m,t} = 1 \ ; \ \forall\ i = 1,..,n \ \& \ t = 1,..,TH \quad (20)$$

$$\sum_{i=1}^{n}\sum_{m=1}^{M} r_{i,k} \leq R_k \ ; \ t = 1,..,T \ \& \ k = 1,..,K \quad (21)$$

$$S_i = integer \geq 0 \quad (22)$$

$$y_{i,m,t} = \text{bin} \quad (23)$$

In the proposed model, maximizing positive cash flow of project activities, while considering pre-emptive resources, is considered as main objective. Existence of a fixed discount rate in cash flows of the model will guarantee maximizing NPV of project.

First Limitation in model is defined as determination of exact start date of each activity. Second Limitation ensures the feasibility of the activities precedence relations. The third limitation is considered to ensure inception of project from a feasible time. The fourth limitation is set to be sure that make span of the model will not exceed from the pre-determined time horizon. Fifth and Sixth limitations guarantee that each activity should select only one executive mode and be loyal to it until end of the project. The seventh limitation is set to respect resource capacity.

Following are 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 pre-emptive.
4. The pre-emptive 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. SIMULATED ANNEALING ALGORITHM

Simulated Annealing is inspired by an analogy of annealing process in heat treatment process. For the first time, Kirkpatrick [20] applied SA to combinatorial and other optimization problems.

The SA algorithm starts from a very high temperature where solutions can move to far distances with no sense of direction and speed limit. Such movements allows SA to search wider areas in solution space to find better areas. As colling process continues, the solutions visit closer neighbours in more reasonable direction that help to search more deeply a suspected area. Such high sped of convergence may cause falling in local optima trap. Hence, there must be a mechanism for escaping from local optima as well.

The proposed SA includes two phases to obtain maximum NPV. In the first phase, an engine is developed to generate an initial upper bond solutions string using serial programming method. This will ensure that project does not exceed the pre-determined time horizon. In second phase, an improvement algorithm is developed for improving solutions by finding best movements that maximize NPV using remained pre-emptive resources. The algorithm is loyal to precedence relations constraints through the annealing process.

3.1 **Initial Solution Engine (Phase I)**

At first, SA set an upper bond for problem by adding durations of all activities regardless to the precedence relations.

$$\text{Upper. bound} = \Sigma \text{ duration (i)} \quad (24)$$

Afterward, in this phase, all activities are planned using forward serial programming method considering priority of activities, that generates appropriate candidate for rescheduling process.

3.2 **Improving Process (Phase II)**

Neighbourhoods

Choosing appropriate neighbours plays a key role in directing SA to provide solutions with better quality. Number of neighbours in each temperature (also known as markov chain) is calculate using below function:

$$n = (\varphi_n - \text{rem } \varphi_0) * i \quad (25)$$

In this study a set of functions are employed to choose the most suitable set of neighbours from a solution in each stage:

1. Mode-selector operator: will choose activities in way that project gains maximum NPV while using lowest amount of pre-emptive resources:

$$\text{mode} - \text{indicator}_i = \max\left\{\frac{CF_{i1}.d_{i1}}{\sum_{i=1}^{n} CF_{i1}.d_{i1}}, \frac{CF_{i2}.d_{i2}}{\sum_{i=1}^{n} CF_{i2}.d_{i2}}, \dots, \frac{CF_{im}.d_{im}}{\sum_{i=1}^{n} CF_{im}.d_{im}}\right\}; \forall\ i = 1,..,n \quad (25)$$

2. Precedence operator: will rearrange activities early start of the activities.
3. Cash Flow operator: will set activities with higher cash flows in earlier steps:

$$\text{activity. order}_i = \max\{CF_{1m}.d_{1m}, CF_{2m}.d_{2m}, \dots, CF_{nm}.d_{nm}\} \quad (26)$$

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 then by filling the remained resources by other activities or replacing them, more NPV will be expected.

Solution representing.

A feasible solution is presenting by a solution-string includes $n + m$ elements where the first 'n' elements shows scheduling of activities and last m elements presents total amount of remained preemptive resources and splitting in activities are shown with parenthesis(Figure 1).

Set of Activities						Remained		
A	B	C(1)-	C(2)	E-F	G	27	54	17

Figure 1. A representing scheme sample considering splits in activities

Cooling scheme

Cooling process is expressed using below formula:

$$\varphi_n = \begin{cases} \varphi_0 & ; \text{where } n = 1 \\ \varphi_{n-1} - \dfrac{\beta}{\varphi_{n-1}} & ; \text{where temp}(n) > 0 \\ 0 & ; \text{where temp}(n) \leq 0 \end{cases} \quad (27)$$

Where φ_0 is initial temperature, β is pre-determined cooling rate and φ_{n-1} is calculated temperature in previous iteration.

To illustrate the impacts of choosing appropriate φ_0 and β in providing good solutions let us explain a simple example.

Suppose a cooling process which started at 100 □ C. The appropriate speed and slope of declining the initial temperature allows SA to search more widely in higher temperatures to find the area of optimum solution and then by declining the temperature smoothly, the shorter steps will help to search that area more attentively. Figure 2 shows the steps of cooling process in different iterations (as a simple example) considering various β rates.

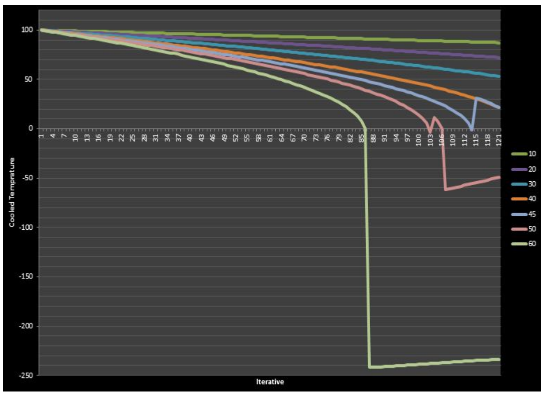

Figure 2. Sample of cooling process diagram considering different β rates

Stop criterion

Searching process will stop whenever one of following criteria is happened:

1. Reaching zero temperature.
2. A set of solutions become converge so that no improvement emerges.
3. There is not any possible repetition for rescheduling activities.

Table 1. Pseudo-code for SA

Procedure:
Initialization of i, m, k, α, φ, β
Begin
Calculate the *UB*
Calculate the *Remained-resource*,
While φ>0
$\varphi_n = \varphi_{n-1} - \frac{\beta}{\varphi_{n-1}}$
Calculate *Markov chain member (n)*
while *n>0*
Run *Select Activity Sub-procedure*
$mode - indicator_i = max\left\{\frac{CF_{i1}.d_{i1}}{\sum_{i=1}^{n} CF_{i1}.d_{i1}}, \frac{CF_{i2}.d_{i2}}{\sum_{i=1}^{n} CF_{i2}.d_{i2}}, \ldots, \frac{CF_{im}.d_{im}}{\sum_{i=1}^{n} CF_{im}.d_{im}}\right\}$
Find *Activity precedence Matrix*
For all *e∈Activity-precedence-Matrix*
Calculate $ES(e_j)+d(e_j)$
For j
For t ∈ $(ES(e_j)+d(e_j)$*,ES(select-activity))*
IF CF(select-activity)> CF(Activity-precedence-Matrix)
For R_k
IF $Resource - remained_k > R_k$
GANTT(select-activity ,t)=1; GANTT(Activity-precedence-Matrix,t)=1
Calculate *NPV*
n=n-1
End

4. COMPUTATIONAL EXPERIMENT

To examine and verify robustness of the proposed backward method in maximizing NPV considering pre-emptive resources, several problems in small, medium and large sizes are solved by the Matlab R2009a software on an Intel® Core i7 laptop which is supported by 4 Mb RAM.

The upper bound is considered as time horizon of each problem. A comparative evaluation of the proposed approach is made comparing gained results with feasible upper bound limits for each problem.

Comparing the results demonstrates that the proposed backward method can significantly improve NPV by using remained resources.

This methodology can effectively decrease make span of the project as well (see Table 2& Table **3**).

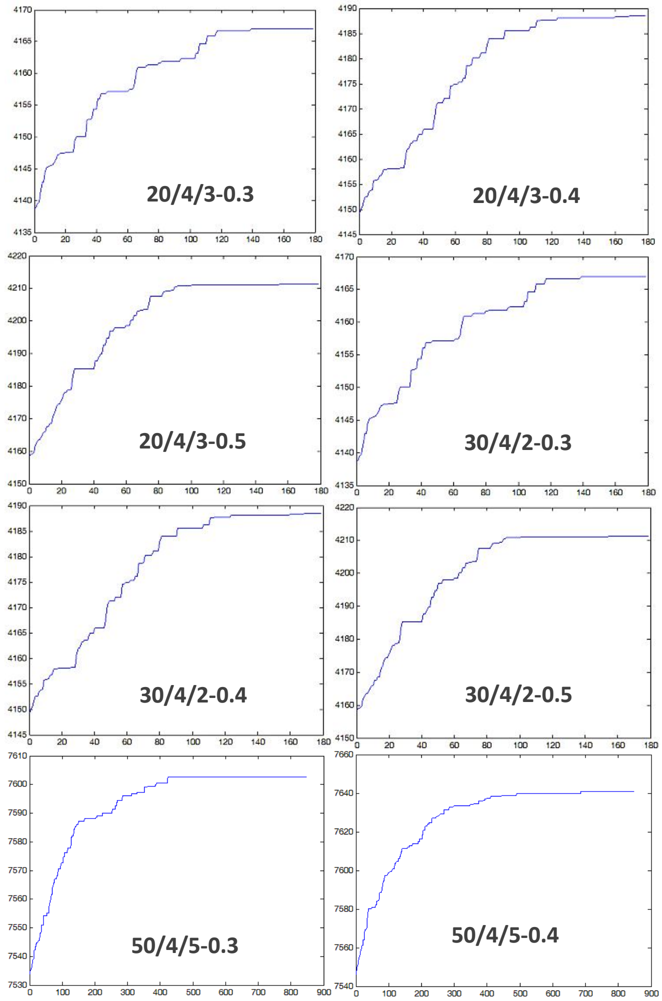

Figure 3. NPV graphs for examples in Table 1

In above figures, improving the NPV while running the MATLAB program is recorded and presented to show how using the remained pre-emptive resources can be used for maximizing NPV through cooling scheme. For each case in Table 2 & Table **3** each problem is solved considering 3 discounted rates (α) to check the performance of the objective function.

The results are then evaluated with serial programming method.

In all presented graphs, SA effectively improved NPV with a reasonable convergence rate in initial steps which makes SA strong enough to solve MRCPSP-DCF using proposed backward method.

Then, shifting activities with more positive cash flows to use remained resources or by making split in less important activities better solutions are achieved (Table **4**). For instance, in example 15/4/3-0.5, respecting to priority relations, SA split activity D into 2 separated parts in order to shift activities E,F,G & H back to improve NPV as these activities have more cash flows (which is shown as D(1)-E-F-G-H-D(2)).

Figure 4 shows power graphs for the last two examples (50/4/5-0.3 & 50/4/5-0.4) and also power functions which reveal the efficiency of the proposed method to improve NPV.

As seen, high rate of R^2 in both examples (0.9547 and 0.9413 respectively) shows high tendency of solution convergence as a consequence of backward method which means the proposed method follows a logical way to find the optimal (or semi-optimal) solution while no illogical answer reveals during the process of cooling the temperature otherwise the amount of R^2will significantly dropped.

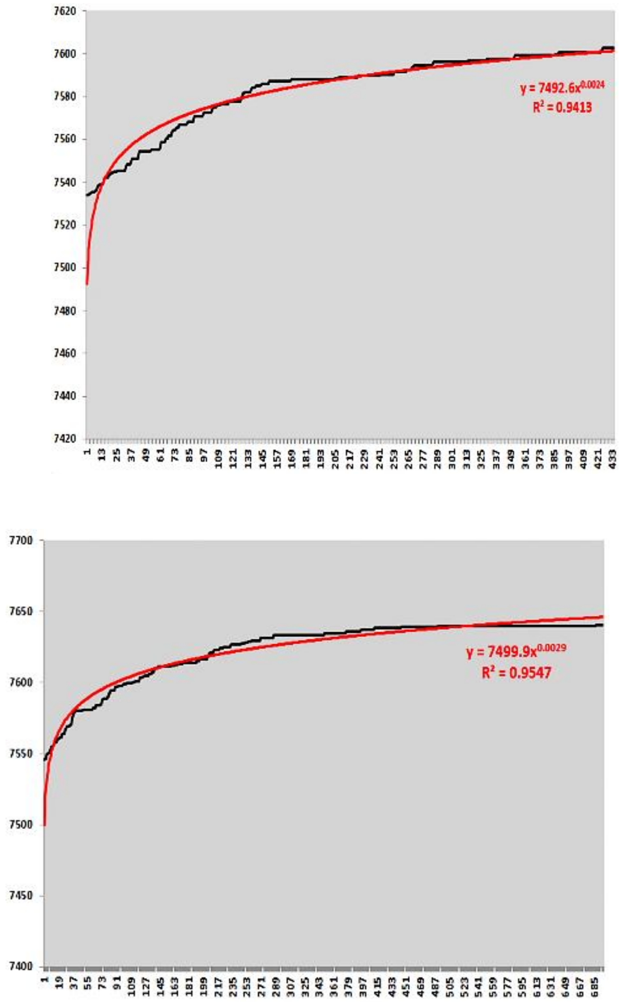

Figure 4. Power graph for example 50/4/5-0.3 & 0.4 respectively

Table 2. Numerical examples for proposed backward method

No.	Problem	α	Available Pre-emptive Resources	φ_0	β	UB	Predecence Matrix	FSP*	SA	ΔNPV	RMS**	Itr.
1	6/2/3	0.3	30/40	4	1	28	A, B, C; -, A, A; D, E, F; B, C, D-E	724.6722	720.3227	4.3495	19	32
2	6/2/3	0.4	30/40	4	1	28		731.9818	727.264	**4.7178**	19	17
3	6/2/3	0.5	30/40	4	1	28		738.8922	734.2912	4.601	19	13
4	13/2/2	0.3	30/40	5	1	73	A, B, C, D, E, F, G; A, A, A, B, B, C; H, I, J, K, L, M; D, D, E-F, G, H-I, J-K-L	2006.8	1987.4	19.4	42	78
5	13/2/2	0.4	30/40	5	1	73		2023.6	2001.4	**22.2**	24	59
6	13/2/2	0.5	30/40	5	1	73		2043.7	2015.6	**28.1**	15	34
7	15/4/3	0.3	30/40/40	4	1	81	A, B, C, D, E, F, G, H; -, A, A, A, B, B, C, C; I, J, K, L, M, N, O; D, D, E-F, G-H, I-J, K-L, M-N	2556.3	2539.5	16.8	28	11
8	15/4/3	0.4	30/40/40	4	1	81		2585.6	2559.7	25.9	46	152
9	15/4/3	0.5	30/40/40	4	1	81		2611.2	2577.5	**33.7**	46	108
10	18/3/3	0.3	40/45/45	5	1	99	A, B, C, D, E, F, G, H, I; -, A, A, B, B, C, C, D, E; J, K, L, M, N, O, P, Q, R; F, G, H, H, I, J, K, K, L-M-N-O-P-Q	2791.6	2774.3	17.3	59	127
11	18/3/3	0.4	40/45/45	5	1	99		2811.7	2789.2	22.5	59	91
12	18/3/3	0.5	40/45/45	5	1	99		2833.8	2804.3	**29.5**	59	131

FSP: Serial Programming *RMS**:Reduced Makespan*

Table 3. Numerical examples for proposed backward method

No.	Problem	α	Available Pre-emptive Resources	φ_0	β	UB	Predecence Matrix	FSP*	SA	ΔNPV	RMS**	Itr.
13	20/4/3	0.3	25/30/25	9	1	105	A, B, C, D, E, F, G, H, I, J; -, A, A, A, B, B, C, D, D, E; K, L, M, N, O, P, R, S, T; F, F, G, H, I, I, J-K, L-M, N-O-P, Q-R-S	2935	2916.7	18.3	38	98
14	20/4/3	0.4	25/30/25	9	1	105		2958.3	2932.6	25.7	58	104
15	20/4/3	0.5	25/30/25	9	1	105		2984	2950.7	**33.3**	42	154
16	30/4/2	0.3	40/30	3	1	150	A, B, C, D, E, F, G, H, I, J; -, A, A, A, B, B, C, D, D; K, L, M, N, O, P, Q, R, S, T; E, F, F, G, H, H, I, J, J, K; U, V, W, X, Y, Z, AA, AB, AC, AD; L-M, O, P, Q-R, S-T, N-U, V-W, X-Y, Z-AA, AB-AC	4194.9	4166.9	28	61	119
17	30/4/2	0.4	40/30	3	1	150		4227.8	4188.5	39.3	26	180
18	30/4/2	0.5	40/30	3	1	150		4264.4	4211.4	**53**	26	134
19	50/4/5	0.3	80/90/70/80/70	4	1	255	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O; -, A, A, A, A, B, B, B, C, C, D, E, E, F, F-G; P, Q, R, S, T, U, V, W, X, Y, Z, AA, AB, AC, AD; H, I, J, J, K, K, L, M, N, N, O-P, Q, R, S, T; AE, AF, AG, AH, AI, AJ, AK, AL, AM, AN, AO, AP, AQ, AR, AS; T, U-V, W, X-Y, Z-AA, AA, AB-AC, AD-AE, AF-AG, AH-AI, AJ-AK, AL, AM, AM, AN-AO; AT, AU, AV, AW, AX; AP, AP, AQ-AR, AT-AU, AS-AW-AV, AV	7671.4	7602.7	68.7	204	425
20	50/4/5	0.4	80/90/70/80/70	4	1	255		7735.4	7640.7	94.7	204	687
21	50/4/5	0.5	80/90/70/80/70	4	1	255		7797.9	7679.1	**118.8**	203	507

FSP: Serial Programming *RMS**:Reduced Makespan*

Table 4. Solution String Presentation for Solved Case studies

No.	Example	Solution String
1	6/2/3	A-BC-DE-F
2	6/2/3	A-BC-DE-F
3	6/2/3	A-BC-DE-F
4	13/2/2	A-B-C-D-G(1)-H-I-E-FG(2)-L-J-K-M
5	13/2/2	A-B-C-D-G(1)-H-I-E-FG(2)-J-L- G(3)-K-M
6	13/2/2	A-B-C-D-G-H(1)-I(1)-E-FI(2)-J-K-H(2)-L-M
7	15/4/3	A-B-C-D-G(1)H-E-F-G(2)-J(1)-I-J(2)-K-J(3)-M-L-N-O
8	15/4/3	A-B-C-D(1)- G-H-E-F-D(2)- L-I-J-K-NM -O
9	15/4/3	A-B-C-D(1)- G-H-E-F-D(2)- L-I-J-K-N-M-O
10	18/3/3	A-B-C-D-E-F-G-I-H-K-J-N-L-M-P-Q-O-R
11	18/3/3	A-B-C-D-E-F-G-I-H-K-J-N-L-M-P-Q-O-R
12	18/3/3	A-B-C-D-E-F-G-I-H-K-J-N-L-M-P-Q-O-R
13	20/4/3	A-B-C-D-G-E-F-H-M-I(1)-J-K-L-N-I(2)-O-P-Q
14	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
15	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
16	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
17	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
18	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
19	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
20	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
21	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

5. DISCUSSION

This paper presents a mathematical model for a resource constrained scheduling considering pre-emptive resources and activity splitting ability. The proposed method determines activity scheduling with the aim of maximizing the net present value of activities through life-cycle of the project. A Simulated annealing algorithm is used for handling model. It is observed that role of scarce resources with pre-emption right in scheduling issues is of paramount importance as they can change the scheduling route of a project entirely and increase (or decrease) the make span of the project by making delay on activities.

As far as the issue of maximizing net present value of project is concerned, there exists inclination to use available resources remaining. By using such opportunities, more efforts have been taken in a backward scheduling with the aim of adding present value of activities. Based on obtained results, it can be conclude that the proposed method is efficient enough to find optimal or near optimal solutions for MRCPSP-DCF problems with the objective of maximizing NPV.

6. CONCLUSION

By developing a backward dynamic procedure an appropriate and logical method was developed for maximizing positive cash flow of activities using available unscheduled pre-emptive resources. The novel mathematical calculation procedure shows that given the need to assign activities, considering the limits of pre-emptive resources availability, the efficient scheduling will maximize net present value while decline make span of the project.

Acknowledgment

The authors would like to be thankful of editors and anonymous reviewers for their positive comments through the publishing periods.

7. REFERENCES

[1] Alcaraz, J., Maroto, C., & Ruiz, R. (2003). "Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms". Journal of the Operational Research Society, 54(6), 614-626.

[2] Ballestín, F., Valls, V., & Quintanilla, S. (2008). "Pre-emption in resource-constrained project scheduling". European Journal of Operational Research, 189(3), 1136-1152.

[3] Baroum, S. M., & Patterson, J. H. (1999). An exact solution procedure for maximizing the net present value of cash flows in a network Project Scheduling (pp. 107-134): Springer.

[4] 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.

[5] 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.

[6] 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.

[7] De Reyck, B. (1998). "A branch-and-bound procedure for the resource-constrained project scheduling problem with

generalized precedence relations". European Journal of Operational Research, 111(1), 152-174.
[8] Demeulemeester, E. L., & Herroelen, W. S. (1996). "An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem". European Journal of Operational Research, 90(2), 334-348.
[9] 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.
[10] Etgar, R., Shtub, A., & LeBlanc, L. J. (1997). "Scheduling projects to maximize net present value—the case of time-dependent, contingent cash flows". European Journal of Operational Research, 96(1), 90-96.
[11] Grinold, R. C. (1972). "The payment scheduling problem". Naval Research Logistics Quarterly, 19(1), 123-136.
[12] Hartmann, S. (2001). "Project scheduling with multiple modes: a genetic algorithm". Annals of Operations Research, 102(1-4), 111-135.
[13] 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.
[14] Icmeli, O., & Erenguc, S. S. (1994). "A tabu search procedure for the resource constrained project scheduling problem with discounted cash flows". Computers & operations research, 21(8), 841-853.
[15] Icmeli, O., Erenguc, S. S., & Zappe, C. J. (1993). "Project scheduling problems: a survey". International Journal of Operations & Production Management, 13(11), 80-91.
[16] 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.
[17] Ke, H., & Liu, B. (2010). "Fuzzy project scheduling problem and its hybrid intelligent algorithm". Applied Mathematical Modelling, 34(2), 301-308.
[18] Kelley, J. E. (1963). "The critical-path method: Resources planning and scheduling". Industrial scheduling, 13, 347-365.
[19] 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.
[20] Kirkpatrick, S. (1984). "Optimization by simulated annealing: Quantitative studies". Journal of statistical physics, 34(5-6), 975-986.
[21] Kolisch, R., & Drexl, A. (1997). "Local search for nonpreemptive multi-mode resource-constrained project scheduling". IIE transactions, 29(11), 987-999.
[22] 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.
[23] 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.
[24] Lova, A., Tormos, P., & Barber, F. (2006). "Multi-mode resource constrained project scheduling: Scheduling schemes, priority rules and mode selection rules". Inteligencia artificial, 30(10), 69-86.
[25] 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.
[26] Mori, M., & Tseng, C. C. (1997). "A genetic algorithm for multi-mode resource constrained project scheduling problem". European Journal of Operational Research, 100(1), 134-141.
[27] Ozdamar, L. (1999). "A genetic algorithm approach to a general category project scheduling problem". Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 29(1), 44-59.
[28] Peteghem, V. 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.
[29] Russell, A. (1970). "Cash flows in networks". Management Science, 16(5), 357-373.
[30] 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.
[31] 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.
[32] Yang, K. K., Talbot, F. B., & Patterson, J. H. (1993). "Scheduling a project to maximize its net present value: an integer programming approach". European Journal of Operational Research, 64(2), 188-198.
[33] Zhang, H., Li, H., & Tam, C. (2006). "Particle swarm optimization for resource-constrained project scheduling". International Journal of Project Management, 24(1), 83-92.
[34] Zhu, D., & Padman, R. (1999). "A metaheuristic scheduling procedure for resource□constrained projects with cash flows". Naval Research Logistics (NRL), 46(8), 912-927.

Obrnuti pristup za povećanje neto sadašnje vrednosti višemodnog preventivnog resursno ograničenog projektno određenog problema sa sniženim protocima novca pomoću algoritma simultanog otpuštanja

Aidin Delgoshaei, Mohd Khairol Ariffin, B. T. Hang Tuah Bin Baharudin, Zulkiflle Leman

Primljen (25.07.2014) Recenziran (10.09.2014); Prihvaćen (01.10.2014)

Apstrakt

U ovom radu se predlaže obrnuti pristup za povećanje neto sadašnje vrednosti (NPV) u višemodnom resursno ograničenom projektno određenom problemu dok se pretpostavljaju sniženi pozitivni tokovi novca (MRCPSP-DC). Koristi se progresivni metod plaćanja i svi resursi su posmatrani kao preventivni. Predloženi pristup povećava NPV uz pomoć neisplaniranih resursa preko resursnog kalendara u obrnutom modu. Algoritam simultanog otpuštanja je razvijen radi rešavanja eksperimentalnih slučajeva sa 50 varijabli i rezultati su upoređeni sa metodom obučavanja serijskim programiranjem. Izvanredni rezultati pokazuju da se predloženi metod može efikasno poboljšati pomoću NPV u MRCPSP-DC dok je podela aktivnosti dozvoljena.

Ključne reči: *planiranje projekta, preventivno, resursno ograničeno, simultano otpuštanje*