World Applied Sciences Journal 7 (1): 113-123, 2009
ISSN 1818-4952

An Algorithm for Solving Project Scheduling to Maximize Net Present Value

[1]Ahmad Ali Bozorgzad and [2]A. Hadi-Vencheh

[1]Department of Industrial Engineering, Azad University, Najafabad, Isfahan, Iran
[2]Department of Mathematics, Azad University, Khorasgan, Isfahan, Iran

Abstract: In this paper, we present an innovative and applicable algorithm for solving the project scheduling to maximize net present value (NPV). Our algorithm has been coded in visual C++ and tested by some random problems. The obtained results show that the proposed algorithm can solve the projects with more than 700 activities and two complexity network coefficient (CNC) in less than two seconds. Also it is able to improve the NPV of project with 200 activities, 2 CNC, discount rate of 24% and 100 days receiving periods, about 10.73%.

Key words: Net present value, Critical path method, Cash flow, Float, Complexity network coefficient

INTRODUCTION

One of the most important goals of a project is profitability and trying to increase the amount of profitability. The problem of project time-budget in order to maximize net present value (NPV) is a certain case of problem in project time-budget that addition to time-budget under different conditions and limitations, it possible to maximize floats and time opportunities and to maximize the profitability of the project. With regard to long duration of implementing the most projects in under developing countries, increased volume of cash flow during implementation of projects, discount rate, capital limitation and so on. We conclude that the best factor for maximizing profitability is NPV factor.

The main goal of this study is to achieve a method such that in addition to considering time-limitation and pre-request relations of activities maximize NPV and profitability of a project. In addition to main goal, the following goals are achievable too:

- By specifying manner and amount of cash flow during implementation of a project and final scheduling of projects, it will be possible to provide a comprehensive cost planning. By cost planning before starting a project, it is possible to get aware of time and needed capital in different time sections.
- By determining time and cost planning, it is possible to do an efficient cost control during implementing a project. And in addition to examine deviations in cost and income, it is possible to track the effects of real progress on NPV and the situation of project in terms of profit or loss in different time sections.

The rest of this Paper is organized as follow: in the following section we review the literature. In section 3 we state pre-hypotheses of our model. Section 4 devoted to the proposed algorithm. In section 5 we test our proposed model. Section 6 concludes.

Literature Review: The problems of project time-budget, henceforth PS-Max-NPV, have been analyzed and investigated from its raising in 1970 until now in different aspect and states. A general method that has been similarly applied in all previous researches, is time-budget of standard problems (without considering NPV) by using common techniques of project management (PN, GEPT, PERT, CPM, …) and getting the time needed for completing a project, determination of early and late time of starting and finishing of activities, computation of rate of activity float, determination of critical path, determination of critical activities and so on. And then using different methods and algorithms (branch and bound, tabu search, simulated annealing, innovative and …) in order to use floats and displacement capacities of project activities for maximizing NPV. Generally, the field and range of problem diversity of PS - MAX - NPV and types of performed studies can be summarized as follows:

- Resolution of PS-MAX-NPV problems with regard to considering or not considering resource restriction.
- Resolution of PS-MAX-NPV problems in different states of cash flow (the type of contract mastery on project).
- Resolution of PS-MAX-NPV problems in different network states (AON-AOA).

Corresponding Author: Ahmad Ali Bozorgzad, Department of Industrial Engineering, Azad University, Najafabad, Isfahan, Iran

For solving PS-MAX-NPV problems in different states, resource limitation, the type of network used and the type of cash flow, different solutions have been offered. For the first time, Russell proposed the problems of project time-budget for maximizing NPV [8]. The hypotheses of Russell's model printed in the journal of management science were considered a network AOA type, fixed cash flow and independent of time of events and activities and prerequisite relations of FS type with zero delay rates. Russell proposed a non linear mathematical model for maximizing NPV and then solved objective function by using first sentence of Taylor series (Russell, 1970, 357).

Grinold converted Russell's model to a linear mathematical model and proposed two solutions for PS-MAX-NPV in AOA network and without resource limitation (Grinold, 1972, 123).

Doerch and Patterson proposed a planning mathematical model with zero and one in the state of budget limitation (Doersh, 1977, 882). Russell in completing his previous work and adding resource limitation to that model tested six innovative solutions for PS-MAX-NPV problems in 80 problems and concluded that no method solely can be appropriate for problems (Russell, 1986, 291). Smith-Daniel and Aquilano [12] by continuing Russell's work and considering the time of actualizing positive and negative cash flow, is actualized in early stage of each activity and positive cash flow solely is actualized at the end of project and after completing project. Then, they with regard to above hypotheses and by using an example showed that time-budget of all non critical activities that have negative cash flow will maximize NPV in latest possible time and time-budget activities with positive cash flow will maximize NPV in earliest possible time. Elmaghraby and Herrolen [2] presented a method known as Elmaghraby interpolation for resolving PS-MAX-NPV problems with at least 20 node and 190 activities that were produced randomly.They argued that produced programs by this algorithm were resolved in less than 21 seconds by PC computers. Sepil and Ortac [10] showed that the method presented by Elmaghraby and Herroelen only offer one better solution (partial optimal) and can not always offer final optimal solution. Etgar *et al*. entered non increasing cash flow in PS - Max - NPV for the first time [3]. They raised dependency of cash flow activities to completing time and argued that the volume of cash flow of every activity changes non- increasing with regard to its completion and then offered proximity method know as simulated annealing. Kazaz and Sepil presented a mathematical method called mixed integer programming (MIP) and resolved random produced problems by using Lindo software and argued that presented model offers optimal solution [7]. Sepil and Ortac expanded Kazaz and Sepil model and by considering limitation of renewable resources they presented an innovative method for resolving PS-MAX NPV problems [10]. Shtub and Etgar fixed all of their previous hypotheses and used simulated annealing instead of the branch and bound for resolving PS - MAX - NPV [4]. They produced 168 random problems and tested branch and bound algorithm and concluded that the quality of branch and bound solutions is somewhat better than simulated annealing (SA). Also they concluded that computation time comparison to SA method has been decreased a great deal. Etgar and Shtub [4] expanded Elmaghraby and Herroelen method and for the first time considered cash flow as non - increasing linear. They explained their algorithm with an example, but did not present any computation results. Vanhoucke *et al*. [3] by considering non - increasing linear process raised by Etgar and Shtub (1999) used recursive search algorithm for its resolution. They tested their model by some random problems.

Pre-Hypotheses of Model: Considering the fact that the main goal of this study is to use floats and time deadline of project activities, in order to maximize the NPV; so by considering project deadline and other possible restrictions time situation of project activities change in such a way that maximize NPV and as a result profitability of implementing project reaches maximum.

In the presented model it has been attempted to consider pre - hypotheses in such a way that have maximum adaptability to realities.

Cash Flow Mode: Cash inflow and outflow during implementing project has been considered separately. Cash outflow or a negative amount that is payments of main contractor to sub - contractors is done on nodes after completing related activity. Cash inflow or positive amounts received from employer are actualized in fixed time interval (1T, 2T, 3T …) for completing activities during that period. The reason for considering positive cash flow is that indeed cash inflow or amount received by contractor is not done separately at starting or ending every activity separately, but is done in certain situations of project and with regard to performed work until the time and in definite time intervals, for instance monthly or each three months. One of the main differences between our algorithm and the previous studies which have increased efficiency and realism of the model a great deal is considering positive cash flow as above.

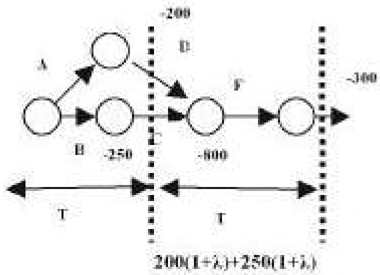


Fig. 1: Positive and negative cash flow

In Fig. 1, separate positive and negative cash flow has been showed. Numbers on each node represent total costs of activities ended to that node and relation shown at the end of T period represents net calculation positive cash flow in T time.

In the mentioned relation at the end of T period, λ is project profit coefficient that is estimated experimentally on the basis of previous similar work or is determined by employer on the basis of contract between employer and contractor. λ shows that contractor for undergoing cost C will receive amount of $C\lambda$ profit.

Type of Network: By considering the structure of cash flow and in order to simplify the model and to coordinate network with using algorithm that is a recursive search algorithm, AOA networks has been applied. In this network, with regard to cash flow mode, every node has a negative cash flow (outflow) that represents total costs of implementing activities to that node.

Type of Cash Flow: In the presented model what is important is considering project deadline or in the other words, time of actualizing end node. Also type of cash flow and performing activities one by one are very important. that is, fixed cash flow type.

Type of Limitations: The only limitation of model is limitation of prerequisite relation in type of end to start (FS) with delay rate of zero.

The Algorithm: According to what mentioned, the most complete and the best method in resolving project scheduling problems in order to maximize NPV in the case of non-limitation of resources are [11] and [13]. Vanhouke *et al*. presented an innovative algorithm called recursive search algorithm for resolving PS-MAX-NPV problems and tested their model by producing random problem in different states, nodes numbers, complexity coefficient and project deadline and showed the results.

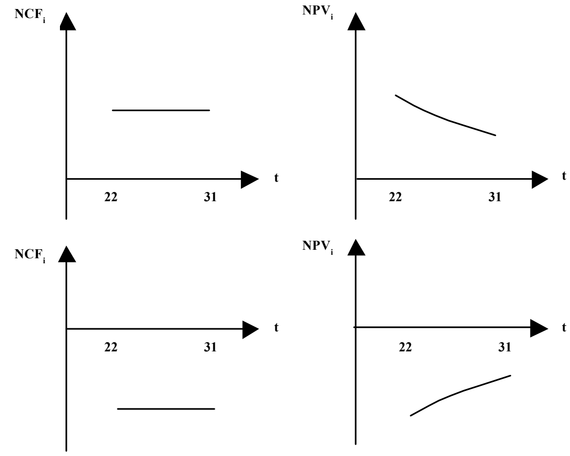


Fig. 2: Net cash flow and NPV for node i

Shtub and Etgar by considering AOA network expanded branch and bound algorithm and like Vanhouke *et al*. tested their algorithm by random problems and presented their results. The proposed algorithm in this study is derived from recursive search algorithm, but major differences in the pre-hypotheses, different ours from Vanhouke model. By comparing net function of cash flow in Etgar and Shtub with net function of cash flow in the proposed model (Fig. 2) and also the type of proposed network in two networks, AOA has been used; It can be seen that the presented innovative algorithm is a combination of branch and bound method and recursive search of Vanhouke *et al*.

The proposed recursive search algorithm in this study hereafter is called "step by step recursive search algorithm" due to its specific orientation.

In order to simplify work and better understanding of method the hypotheses of actualizing positive cash flow in fixed period is omitted and it is supposed that positive cash flow similar to negative cash flow is actualized on nodes. Hence for every node a cash flow that can be positive, negative or zero is computable. So this type of cash flow is viewed positive. In our method firstly, project network is drawn and the earliest and the latest time of all nodes are computed. Then, all nodes are scheduled in the earliest time and initial NPV is computed and this NPV is considered as low bound. Then search starts from end of project and first it is examined for every node that whether this is allowed to change time or not (because all nodes are scheduled in the earliest time. By changing time nodes we mean movement of time node toward project deadline). Allowable nodes first, have time deadline, second change their time because increasing project NPV.

According to Fig. 2, transfer of time of every node with negative cash flow toward the latest time of node, increases NPV of node, contrary, change of nodes time positive cash flow from the earliest time cause decreasing NPV of that node. So, for every node it is examined that weather is has negative or positive cash flow. If it has positive cash flow is not allowable to change time and may cause NPV and project decreasing. Thus, these nodes are relieved and search within previous nodes is continued. But if these nodes have negative cash flow are allowable to change time. It is clear from Fig. 2 that the best situation for this node is scheduling in the latest possible time. So, this node temporary is scheduled at the latest time. Because delaying this node may cause delaying post requisite nodes, that part of network after expected node is rescheduled and new NPV is computed on this basis. If computed NPV is larger than low bound of problem, this scheduling is accepted and search with previous nodes is continued. If NPV has not been better, scheduling of network is converted to previous state (because transfer forward of one or several nodes with positive NPV has been considered due to transfer forward).

In the next step, by considering the fact that every transfer to forward of this node with negative cash flow cause increasing NPV, minimum interval of this node that its scheduling of it, if this rate is larger than zero, this node in time [minimum interval of node with later nodes + the earliest time of node], is scheduled and new NPV is computed, by making sure of the fact that new NPV is related to low bound of problem it is stored as low bound and search with previous nodes is continued.

Minimum interval of every node with later node is the period that can shift the time of one node forward, without effecting on latter nodes. The rate of this interval is minimum free float of all activities that is started. Considering concept of minimum interval of every node with latter nodes and considering Fig. 2 one can conclude that shift of one node with negative cash flow certainly will improve NPV of the node project.

After clarifying the method, the hypothesis of actualizing positive cash flow in fixed time period that increases model realism and considered indicator of model, is added to model. By adding this, selection of allowable nodes needs change and modification. According to mentioned, those nodes were allowable to change time that had negative cash flow but by adding this hypothesis to model cash flow of each node is calculated by ($-c_i+c_i(1+\lambda)$) and it is positive for all nodes. Thus, identifying nodes allowable to change time is not generalization to this state and an appropriate method is needed for selecting these types of nodes. The following example shows how to select allowable nodes to change time and computation of optimal value of this change time.

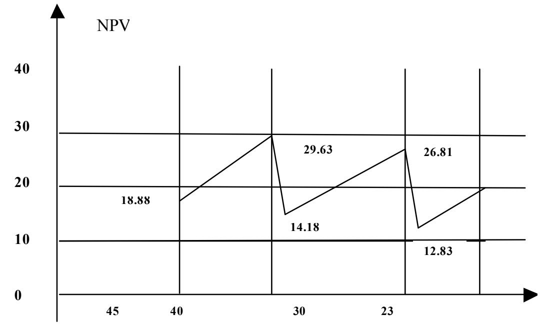


Fig. 3: NPV for node i in Example-1

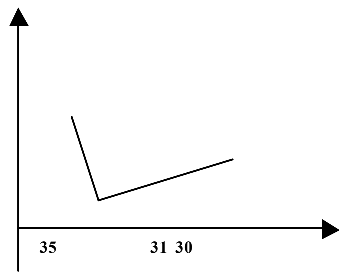


Fig. 4: NPV for node i when [Ei, Li] =[30, 35]

Example 1. Consider node i with the earliest time of 23 and the latest time of 45 unit and cash outflow of -200. Suppose actualizing positive cash flow, interest rate and coefficient profitability for this node are 10 time unit (T=10), 0.01 and 0.2, respectively. Fig. 3 shows NPV for this node.

According to Fig. 3 optimal state for maximizing NPV of this node is in time 30 or first receive of positive cash flow is in interval between the earliest and the latest time of actualizing this node. On the hypotheses three cases can be considered for every node, say i, in terms of time actualization:

- If integer coefficient like f (f = 1, 2, 3 …. m, where m is the number of activities) is found, such that $E_i = fT$ (E_i is the earliest time of node i). Then, optimal time of this node is the earliest time and any time variation toward deadline of project cause decreasing NPV.
- If integer coefficients like f is found such that (f=1, 2, …, m) $E_i < fT \leq L_i$, L_i is the latest time of node i, then according to example 1, the optimal situation of this node for maximizing NPV is scheduling in the first time of receive between interval [E_i, L_i].

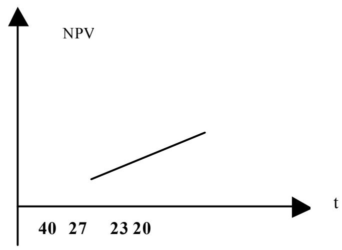


Fig. 5: NPV for node i when T=10 and [Ei, Li] = [23, 27]

- No integer coefficient like f (f=1, 2,…, m) can be found such that Ei≤fT≤Li, thus according to Figure 5 the best situation for this node is its scheduling in the latest possible time.

According to what mentioned above, the general scheme of our algorithm is as follow: first network of project is drawn and the earliest and the latest time and negative and positive cash flow is computed for every node. Then, all nodes are scheduled in the earliest possible time and NPV resulted from this scheduling is stored as low bound of problem. After computation of the initial NPV, search is started from the end node to the first one. As positive cash flow of the last node is actualized in the time of its actualization, the optimal state of this node is scheduling in the earliest possible time so, the last node is relieved and search is continued from one node to last.

In the algorithm for every node it is examined that whether there is a receive time that be between the earliest and latest time of actualization of node or not (whether Ei≤fT≤ Li). If no, then node i is scheduled in the latest time and if this time change is greater than minimum interval with latter nodes, then all of the latter nodes are rescheduled and NPV resulted from this rescheduling is computed and compared to low bound of problem. In the case of improving obtained NPV is stored as low bound and search is continued with previous nodes; otherwise this node in time [the earliest time of the node + minimum interval with latter node] is scheduled and new NPV is computed. This new NPV is stored as low bound.

If a coefficient like f is found (f: 1, 2…, m) such that Ei≤fT≤Li, it is examined that if scheduling of the node in time of the first period of receive between the earliest and the latest time does not need more time change from minimum interval between the node with latter node. Hence this node is scheduled in time of first receive between the earliest and the latest time of node and new NPV is computed, without comparison with low bound. And it is stored as new low bound of the problem and search is continued with previous nodes. But if there is a need to more time change from minimum interval of node i with latter nodes, temporarily this node is scheduled in time of first period of receive and all the other nodes are scheduled after this node and NPV of the project is computed based on the new scheduling and compared to low bound of the problem. In the case of improving, obtained NPV is stored as low bound and search is continued with previous nodes. In the case of not improving, node in time [the earliest node time + minimum interval of node with latter nodes], is scheduled and resulted NPV without comparison to lower bound is stored as low bound. In the same way search is continued from one node to last to the first node.

According to what mentioned the proposed algorithm is as follows:

Step. 1: The network of the project is drawn and by helping critical path method the earliest and the latest time of each node is computed.

Since the cost of every activity is known, hence compute the cash flow of each node and by having coefficients λ and -ci compute positive cash flow of each node by ci(1+λ). In this step all nodes are scheduled at the earliest time and by considering negative and positive cash flow (for each node) and interest rate and time period of positive cash flow the NPV is computed and considered as low bound of problem.

Set: i= n- 1

Step. 2: For node I, examine if there is any integer like f (f= 1, 2 ….m), such that Ei ≤fT ≤Li. If so, go to the next step otherwise continue from step 6.

Step. 3: Calculate fi such that Ei≤fiT≤Li. If there is more than one fi, then consider the smallest one. Compute minimum interval of node i with latter nodes.

If fiT≤Ei+mfi, then consider fiT as the time for node i and compute new NPV and without comparison store it as low bound and go to step 8, otherwise go to the next step.

Step. 4: Schedule node i in fiT (ti = fiT) and reschedule all nodes after node i. Compute NPV obtained from new scheduling. If NPV is better than low bound accept new scheduling and store resulted NPV as low bound and go to step 8, otherwise go to the next step.

Step. 5: Compute minimum interval of node i with later nodes. If it is greater than zero continue the process, otherwise go to step 8.

Schedule node i in Ei+ mfi time, m is the number of activities and compute new NPV and by ensuring that NPV is less than lower bound and without comparison, store it as low bound and continue from step 8.

Step. 6: Temporarily schedule node i in the latest time and reschedule all nodes after the node i. If obtained NPV is greater than low bound accept this scheduling and store NPV as low bound, otherwise go to next step.

Step. 7: If minimum interval of node i with latter nodes is greater than zero then set the time of node i equal to Ei+ mfi and compute NPV resulted from this time change and store it as low bound, otherwise go to next step.

Step. 8: Replace i = i-1. If i > = 1 go to step 2, otherwise accept presented scheduling as optimal scheduling store it and end.

As we see the output of algorithm is the optimal scheduling of nodes of a project and value of NPV. But, the main goal of this study is optimal scheduling of project activities in order to maximize project NPV, hence the algorithm needs a final step in order to turn nodes scheduling to scheduling of project activates. Considering definition of AOA network and also based on the pre- hypotheses (actualization of cash out flow in finishing every activity and actualization of cash in flow after completing activity and at the end of fixed time period), the best time for doing every activity for increasing NPV is its scheduling in the latest possible time. Hence, a step is added as follows:

Step. 9: Replace j = 2.

9.1. Place time of completing all activates ended to node j equal to tj.

Set si= fi - di for all activities that was selected in the previous stage.

If j $\leq$ n go to 9.1, otherwise store scheduling of project activities as optimal scheduling.

Test the Model: The proposed algorithm has been programmed by C^{++} software and its performance accuracy has been verified by previous manual solved problems. The inputs of the program are: program interest rate, project profitability, estimated cost and time for every activity and finally pre-requisite. And two output as first and final results are shown. After providing software program efficiency coefficient and capacity of the algorithm is examined by several randomly produced problems. In order to examine efficiency coefficient of presented algorithm, sensitivity rate of two variable of model indicator i.e. computation time and NPV improvement percent as a result of performing algorithm related to changes of four effective parameters, m (number of activities), q (project deadline) CNC (project complexity coefficient) and T (interval between receive periods) are examined. See Figs.6-13.

Fig. 6 shows computation time related to increasing activities. CPU computation time is also increased gradually; this increase of time after 600 activities has more severity. Increasing time of computation is due to logical activities, because by fixing CNC and increasing number of activities, number of nodes increase too and considering node to node search, by increasing nodes computation time is also increased. As it has been increasing time of computation is due to increasing number of activities, our algorithm can resolve projects with 700 activates and complexity coefficient of 2 in less then 2 second. Another parameter that its sensitivity has been examined is NPV improvement percent. According to Fig.7, NPV improvement percent related to number of activities is also mildly increased with regard to very little increase in NPV percent. One can argue that NPV improvement percent of projects is not sensitive to the number of project activity and utilizing proposed algorithm for every project, regardless of number of activities is appropriate. According to Fig.8, with increasing CNC, computation time is decreased. It is due to the fact that CNC is ratio of number of activities to number of nodes, by fixing number of activities and increasing CNC, number of nodes is decreased and by decreasing number of nodes, time consumption (algorithm is based on node to node search process), is decreased. Also, according to Fig.9, by increasing CNC, NPV improvement percent is mildly decreased, as mentioned, by increasing complexity coefficient and fixing number of activities, project nodes related to number of activities is decreased. Considering the proposed algorithm, by decreasing number of nodes with respect to number of activities, number of states that can increase NPV, is decreased and as a result by increasing CNC, followed by decreasing nodes of project, NPV improvement percent is decreased. Figures 10 and 11 show time change and improvement of NPV with respect to increasing of the project deadline. From thes figures we see that

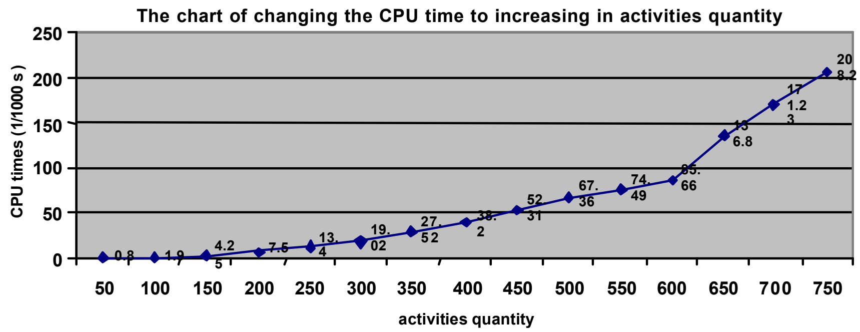


Fig. 6: Changing the CPU time with respect to increasing in activities quantity

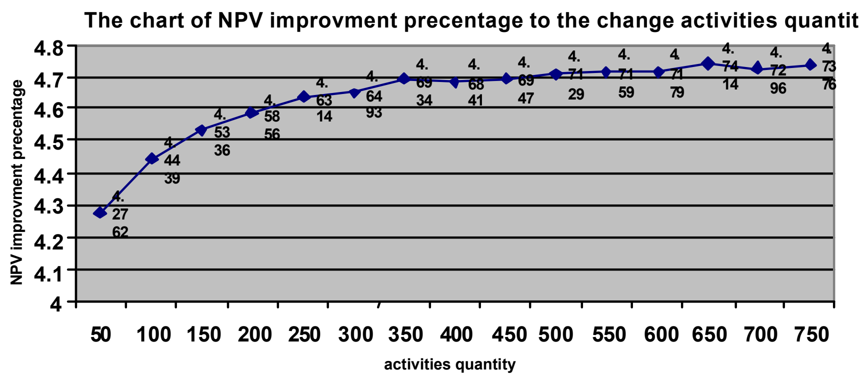


Fig. 7: NPV improvement percentage with respect to the change activities quantity

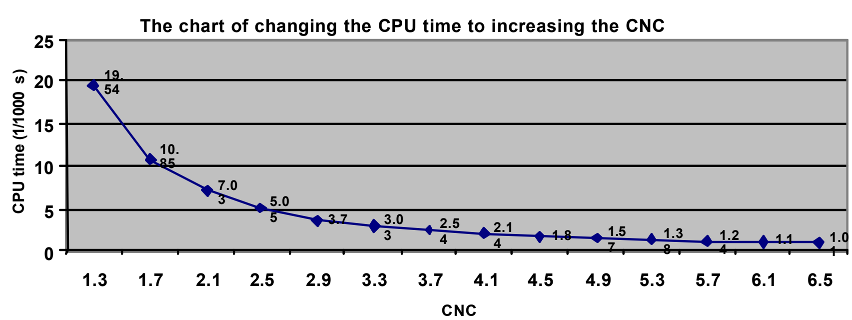


Fig. 8: Changing the CPU time with respect to increasing the CNC

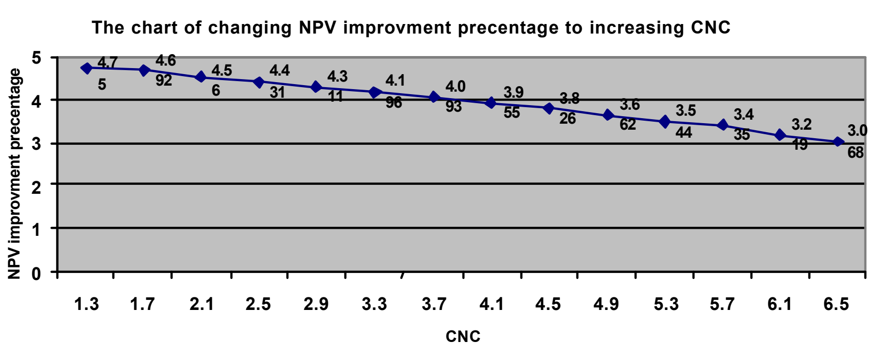


Fig. 9: NPV improvement percentage to increasing the CNC

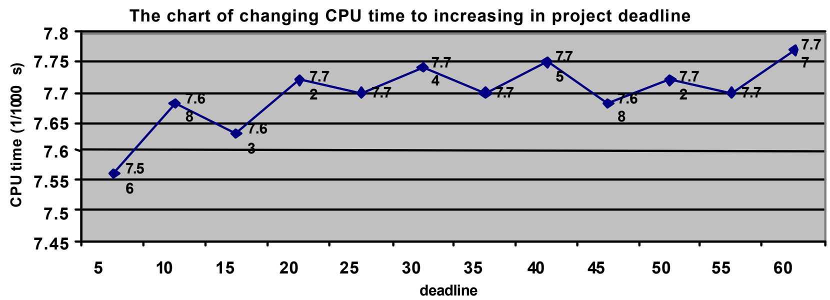


Fig. 10: Changing the CPU time to increasing in project deadline

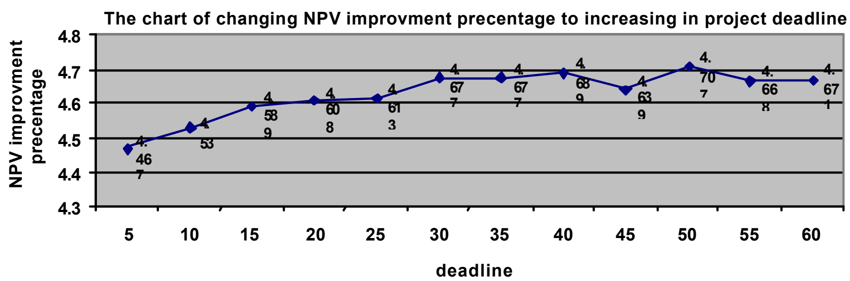


Fig. 11: Changing NPV improvement percentage to increasing the in project deadline

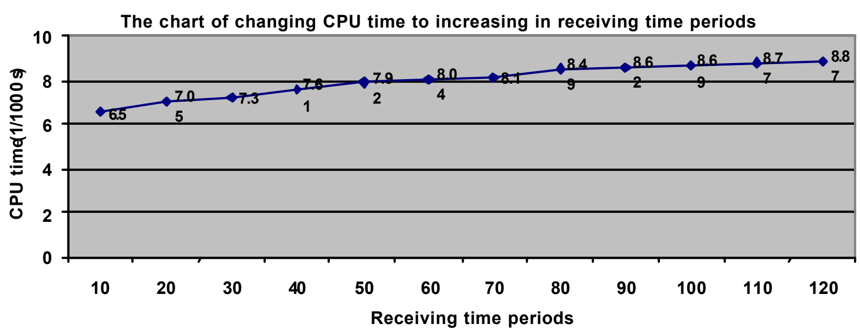


Fig. 12: Changing the CPU time to increasing in receiving time periods

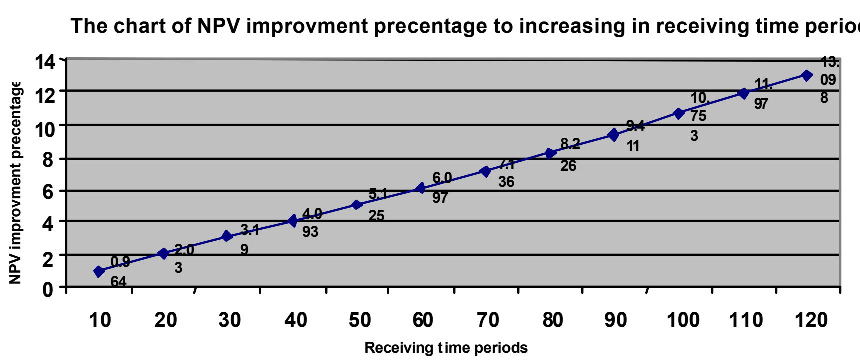


Fig. 13: NPV improvement percentage to increasing receiving time periods

improvement of NPV and time consumption of the proposed algorithm are not sensitive to change deadline. According to Fig. 12, improvement of NPV percent is severely sensitive related to interval change, between received periods. And by increasing this period NPV is almost increased linearly. It should be noted that increasing NPV percent due to increasing T rate does not mean that increasing time interval of received period is a benefit for contactor and increase NPV of project, but by increasing interval between received periods early NPV is decreased. Because, contactor should wait longer for receiving spent costs. Hence, increasing T is not benefit for contractor and decreases NPV of project. The trend of NPV improvement percent due to increasing interval between received represent increasing effect of using recursive search algorithm in order to maximize NPV and by increasing T the necessity utilizing this algorithm is increased. Also with attention to Fig. 13, one can conclude that if time of the project is greater, then effectiveness of performing the algorithm to obtain more profit will be greater.

CONCLUSION

According to obtained results, one can conclude that the proposed algorithm has a high efficiency and is able to solve more than 700 activities and complexity coefficient of 2 in less than 2 seconds. Also the rate of algorithm effectiveness is outstanding, such that percent of NPV improvement by using our algorithm for projects with 200 activities, CNC = 2, á = 24% and received period of 100 days is moderately 10.73 %.

By examining sensitivity analysis it is clear that the time consumption of the algorithm with respect to project deadline and received interval period is indifference; and by generating more severe changes in these two parameters, tangible change is not produced in the computation time. But, computational time is sensitive to number of activities and complexity coefficient of project and has direct relation with number of activities and contrariwise with CNC change (this is due to the nature of node to node search and activity to activity trend of this method). Another important parameter that its sensitivity to four mentioned parameters was examined is NPV improvement percent due to using the proposed algorithm. According to obtained results, NPV improvement percent with respect to two parameters, number of activities and complexity coefficient of project, has very low sensitivity and is indifference to parameter of project deadline, but is severely sensitive to change of interval of received periods. And by doubling this interval, percent of NPV improvement is also double. Hence, by increasing interval between received amounts, effectiveness rate of the algorithm is increased toward NPV improvement. For example, for a project with monthly received period and CNC=2, q=15, m=200 and á = 24% (annual), NPV improvement is 3.19%, but for this project with three month received interval this percent is 9.41%.

REFERENCES

1. Doersch, R.H. and J.H. Patterson, 1977. Scheduling a project to maximize its present value: A zero-one programming approach. Management Science, 23: 882-889.
2. Elmaghraby, S.E. and W. Herroelen, 1990. The scheduling of activities to maximize the net present value of project. European Journal of Operational Research, 49: 35-49.
3. Etgar, R., A. Shtub and L.J. LeBlanc, 1996. Scheduling projects to maximize net present value- the case of time-dependent, contingent cash flows. European Journal of Operational Research, 96: 90-96.
4. Etgar, R. and A. Shtub, 1999. Scheduling project activities to maximize net present value- the case of linear time-dependent, contingent cash flows. International Journal of Production Research, 37: 329-339.
5. Grinold, R.C., 1972. The payment scheduling problem. Naval Research Logistics Quarterly, 19: 123-136.
6. Herroelen, W. and E. Gallens, 1993. Computational experience with an optimal procedure for the scheduling of activities to maximize the net present value of projects. European Journal of operational Research, 65: 274-277.
7. Kazaz, B. and C.B. Sepil, 1996. Project scheduling with discounted cash flows and progress payments. Journal of the Operational Research Society, 47: 1262-1272.
8. Russell, A.H., 1970. Cash flows in networks. Management Science, 16: 357-373.
9. Russell, R.A., 1986. A comparision of heuristics for scheduling projects with cash flows and resource restrictions. Management Science, 32: 291-300.
10. Sepil, C. and N. Ortac, 1997. Performance of the heuristic procedures for constrained projects with progress payment. Journal of the Operational Research Society, 48: 1123-1130.

11. Shtub, A. and R. Etgar, 1997. A branch-and bound algorithm for scheduling projects to maximize the net present value: The case of time independent, contingent cash flows. International Journal of Production Research, 35: 367-3378.
12. Smith-Daniels, D.E. and N.J. Aquilano, 1987. Using a late-start resource-constrained project schedule to improve project net present value. Decision Sciences, 18: 617-630.
13. Vanhouke, M., E. Demeulemeester and W. Herrielen, 2001. Scheduling projects with linearly time-dependent cash flows to maximize the net present value. International Journal of Production Research, 39: 3156-3181.

Appendix 1:

In this appendix we give flowchart of the proposed algorithm. Parameters are:

A: interest rate

N: number of nodes

∂n: project deadline

fii: net inflow of node i

pi: pre-requisite node of node i

h: project profitability coefficient

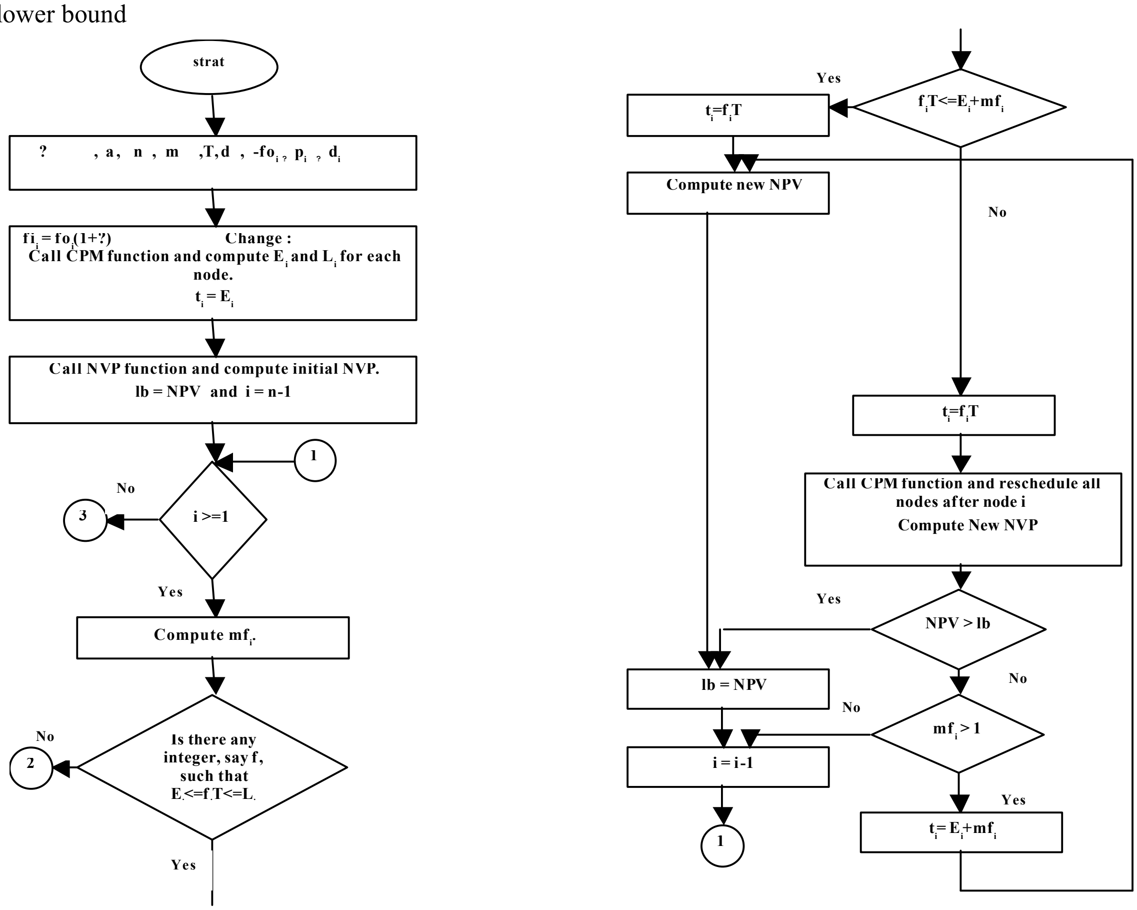
m: number of activities

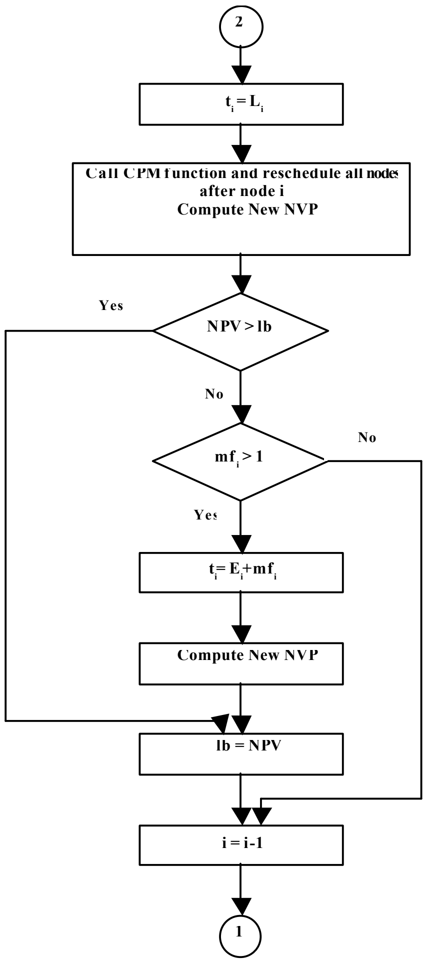
T: recived interval

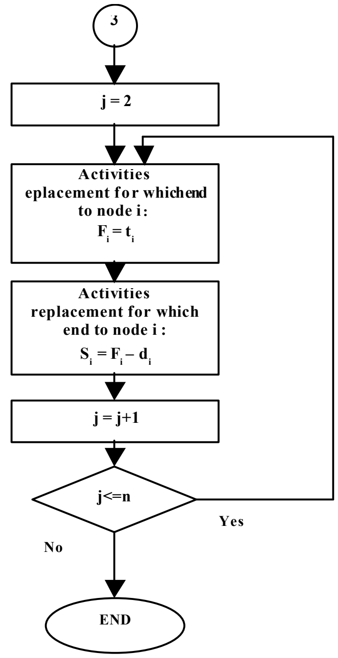
foi: net outflow of node i

di: duration of activity i

lb: lower bound




2
t_i = L_i
Call CPM function and reschedule all nodes after node i
Compute New NVP
Yes
NPV > lb
No
mf_i > 1
No
Yes
t_i= E_i+mf_i
Compute New NVP
lb = NPV
i = i-1
1


3
j = 2
Activities eplacement for which end to node i :
F_i = t_i
Activities replacement for which end to node i :
S_i = F_i – d_i
j = j+1
j<=n
Yes
No
END