Contents list available at JMCS
Journal of Mathematics and Computer Science
Journal Homepage: www.tjmcs.com

Time-Cost Tradeoff for Optimizing Contractor NPV by Cost Payment and Resource Constraints Using NSGA-II Algorithm (Case Study:Bandar Abbas Gas Condensate Refinery Project)

M. Zareei[1a], H.A. Hassan-Pour [a], M. Mosadegh-Khah [a]

[a] *Department of Industrial Engineering, Faculty of Engineering, Imam Hossein University, Tehran, Iran*

mozare66@yahoo.com
hahassan0@yahoo.com
mozare66@hotmail.com

Article history:
Received February 2014
Accepted June 2014
Available online July 2014

Abstract

In this paper, a mathematical model is presented for project scheduling with multiple purposes based on considering cost payment and resource constrains and since this this problem is considered as complex optimization in NP-Hard context, in order to solve proposed method from NSGA-II algorithm and the results are compared with GAMS software in some problems. The proposed method is a Converge to the optimum and efficient solution algorithm. Besides this algorithm is used in some parts of refinery project.

Keywords: Project-scheduling problems, maximize the NPV, the Payments pattern, Time-cost tradeoff, NSGA-II algorithm, GAMS software

1. Introduction

Project-scheduling problems are combination of precedence constraints, resource constraints, completion-time constraint and some side-constraints in which the goal is to find a schedule such that an objective function like project duration, project total costs or net present value (NPV) optimizes[1]. The objective may be based on time, such as minimize the project duration, or on economic aspects, such as minimize the project cost. However, success relative to time does not imply success in economic terms. Often, time-based objectives are in conflict with cost-based objectives. A recurrent situation encountered in practice is the need to complete a project by its due date and maximize profit [2]. Blazewicz et al. (1983) have shown that the RCPSP belongs to the class of the strongly NP-hard problems [3]. Existing techniques for DTCTP can be categorized into two areas: namely, exact and heuristic. Exact algorithms, including linear programming [4], dynamic programming [5], enumeration algorithm [6], or branch and bound algorithm [7], have been extensively employed to solve DTCTPs [8]. However, none of the exact algorithms are able to solve large and hard instances measured in terms of, say, the number of activities. In terms of what current state-of-art algorithms can do, and considering the structure of the project networks as

[1] *Corresponding author*

well as the number of modes per activity, instances with a large number of activities cannot be solved optimally in reasonable amount of time [9]. When the financial aspects of a project are taken into account, the most frequent performance measure used in project scheduling is a net present value (NPV) method. This criterion is computed with cash flows generated by project activities. A cash flow can be positive (cash inflow) or negative (cash outflow). For the contractor, cash outflows represent the expenses caused by manpower, equipment, and/or raw materials while cash inflows represent the client's payments (they are often proportional to the project's advancement). If positive cash flows are considered, the maximization of the net present value is a regular performance measure. Some specific non-regular objective functions are to maximize the NPV (unconstrained resource problem) that was first introduced by Russel, et al [10], maximize discounted cash flow (resourceconstrained problem) [11], minimize the total (weighted) resource consumption , and maximize smoothness of resource usage (resource leveling) [12].In scheduling of construction projects, the project duration can be compressed (crashed) by expediting some of its activities in several ways including; increasing crew size above the normal level, working overtime, or using alternative construction methods. The crashing alternatives come at an additional cost. This trade-off between time and cost has been studied extensively since development of the critical path method (Vanhoucke & Debels, 2007). The objective of time–cost trade-off problem is to identify the set (or sets) of time–cost alternatives that will provide the optimal schedule [13]. Each activity can be implemented by any one out of a finite number of alternative execution modes. In the situation of time–cost tradeoff, an execution mode with longer duration may cost less money, whilst a mode with shorter duration may cost more money. In the situation of time–resource tradeoff, an execution mode with longer duration consumes fewer resources, vice versa [14]. Cash flow means the amount of cash being received and spent during a defined period of time. Without positive cash flows, basic obligations such as payments to suppliers and payrolls cannot be met [15] Cash flow consists of the flows of cash into and out of a business; typical cash out flows on a construction project include interest, material, labor cost, etc., and cash inflows include various payments, such as bonuses. Numerous approaches have been developed for investigating cash flow in construction [16]. The objective of the client is to minimize the net present value (NPV) of the payments to the contractor, whereas the objective of the contractor is to maximize the net return.

Project payment scheduling problem involves how to schedule effectively progress payments including the amount, time or spots (i.e. the key activities or events associated with payments), and so on of payments in the project so as to maximize the profits of the contractor or/and the client [17]. In real life situations there are at least two parties involved in the project: the client, who is the owner of the project, and the contractor, whose job is to execute the project. They have to agree with the method of payment transferring from the client to the contractor for the execution of the project. The ideal situation for the client would be a single payment at the end of the project. The contractor, on the other hand, would like to receive the whole payment at the beginning of the project. An interesting problem is considered in [18], where the goals of the contractor and the client are joined in one model. The authors use a double-loop genetic algorithm to find an equitable solution, which is defined such that both the contractor and the client deviate from their respective ideal solutions by an equal percentage [18].

According to all performed studies, Russell (1970) has studied cash flow in time tabling problems for the first time. Russell considered a promotional NPV problem without considering of resources limitations and presented a non-linear programming method for solving it. Then the case of improving NPV was considered with others and was centralized on RCPSP condition [10].The real goals of contractor and Client is to increase their financial returns (their NPV) in a way that contractor is interested to receive total budget amount within shortest possible period of time and in contrast when we have highest amount of Client's NPV with delayed payments.

Deckro et al. (1995) consider the continuous version of the time-cost tradeoff problem where activity durations are not limited to discrete values. A deadline is imposed, and the cost function is quadratic and increases with an increasing deviation of the actual duration from the given normal duration. Extensions include a budget contraint and bonus or penalty payments in case of early or late project completion, respectively [19]. Bey et al. (1981) considered the implications of a bonus/penalty structure on optimal project schedules for the NPV problem.Icmeli and Erenguc

(1996) introduced a branch and bound procedure to solve the resource constrained project-scheduling problem with discounted cash flows [20]. Dayanand and Padman (2001) also investigate the payment scheduling problems from the client's viewpoint. Several mixed integer linear programming models are introduced according to practical payment rules. The analysis shows that the client obtains the greatest benefit by scheduling the project for early completion such that the payments are not made at regular intervals [21]. Vanhoucke (2010) presented a scatter search algorithm for the resource-constrained project scheduling problem with discounted cash flows. He assumed fixed payments associated with the execution of project activities and developed a heuristic optimization procedure to maximize the net present value of the project subject to the precedence and renewable resource constraints [22].

Nadjafi and Shadrokh (2009) studied unconstrained resource project scheduling problem considering the time value of the money by continuous discounting the cash flows and minimum as well as maximum time-lags between different activities. They proposed a branch and bound algorithm for this project scheduling problem with generalized precedence relations among activities. The literature on solution procedures for the weighted earliness-tardiness project scheduling problem (WETPSP) is very limited [23]. Kazemi and Tavakkoli-Moghaddam (2011) considered a bi-objective multi-mode resource constrained project scheduling problem with positive and negative cash flows. Their first objective is maximization of the net present value (NPV) and the second is to minimize the makespan and floating time (i.e. maximization of robustness) [24]. Mika et. al (2005) who tried to increase contractor's NPV by the use of simulation of melting and gradual freezing and prohibited search and considering 4 types of payment from Client to contractor which are: payment of total amount of contract at the end of project, payment at completion date of activities, payment at equal time tables and payment against project progress [25]. Yang (2005b) took budget uncertainty into account on project time–cost trade-off in a CCP model. Ke et al. (2009) proposed two stochastic models, which were based on two decision-making criteria, CCP and dependent-chance programming (DCP) for the time–cost trade-off problem. A hybrid intelligent algorithm integrating simulation and genetic algorithm (GA) was designed for solving the proposed models [26].

2. Problem description and mathematical formulation model

Our proposed model is categorized in resource-constrained project scheduling problem with discounted cash flows (RCPSPDCF) that can be defined as follows. A project consisting of n activities is represented by an activity-on-node network, $G = (J, E)$, $|J| = n$, where nodes and arcs correspond to activities and precedence constraints between activities, respectively. Nodes in graph G are topologic and numerically numbered, i.e. an activity has always a higher number than all its predecessors. No activity may be started before all its predecessors are finished. The duration of activity $j = (1,2, \dots, n)$ executed is d_j. There are R renewable resources. The number of available units of renewable resource $k = (k = 1,2, \dots, R)$ is R_k. Each activity j is executed requiring r_{jk} units of renewable resource $k = (k = 1,2, \dots, R)$ for its processing. A negative cash flow CF_j^- is associated with the execution of activity j. For each completed activity occurs a negative cash flow amount of until the completion time of a project. Finally, the contractor receives amount of cash flows CF_j^+ for each activity that has completed successfully. The value of an amount of money is a function of the time of receipt or disbursement of cash. Money received today is more valuable than money to be received some time in the future, since today's money can be invested immediately. In order to calculate the value of NPV, a discount rate i α has to be chosen, which represents the return following from investing in the project. The objective is to find an assignment of modes to activities as well as precedence and resource-feasible starting times for all activities such that the net present value of the project is maximized.

All the parameters are used in the proposed RCPSPDCF model are summarized below:

n	Number of activities
G	Acyclic digraph representing the project
d_j	Duration of activity j executed
CF_j^-	Negative cash flow associated with activity j executed

CF_j^+	Positive cash flow associated with activity j executed
ST_j	Starting time of activity j
FT_j	Finishing time of activity j
EF_j	Earliest finishing time of activity j
LF_j	Latest finishing time of activity j
P_j	Set of all predecessors of activity j
R	Number of renewable resources
R_k	Number of available units of renewable resource $k, k = 1,2, \dots, R$
r_{jk}	Number of units of renewable resource k required by activity j executed
α	Discount rate
$\mathrm{C}_{\max}$	The maximum time for completion
T	Horizon of Project Scheduling
NPV	Net Present Value of the project
P_{K}	paid the amount of k
K	The number of continuous payments
U	The total amount of payments

$$X_{jt=}\begin{cases}1 & \text{If completed j activities at time t}\\ 0 & \text{Otherwise}\end{cases}$$

$$\mathrm{y}_{\mathrm{jk}=}\begin{cases}1 & \text{If payment k is done for j}\\ 0 & \text{Otherwise}\end{cases}$$

By using the above notations, the proposed model can be formulated as the following mathematical programming problem:

$$\max NPV \qquad (1)$$

$$\min Cmax = \max\left(\sum_{t=EF_j}^{LF_j} t\, X_{jt}\right) \qquad (2)$$

st:

$$\sum_{t=EF_j}^{LF_j} tX_{jt} \leq \sum_{t=EF_j}^{LF_j} (\, t - d_j)X_{jt} \qquad \forall_{j,w}\in p_j \qquad (3)$$

$$C_j = \sum_{t=EF_J}^{LF_j} tX_{jt} \qquad \forall_j = 1,2, \dots, n \qquad (4)$$

$$C_{max} \geq C_j \qquad \forall_j = 1,2, \dots, n \qquad (5)$$

$$T \leq \sum_{j=1}^{n} \max(\, d_j\,) \qquad (6)$$

$$C_{max} \leq T \qquad (7)$$

$$\sum_{j=1}^{n} r_{jkm} \sum_{b=t}^{t+d_j-1} X_{jb} \leq R_k \qquad , \forall_{k,t} \qquad (8)$$

$$ES_1 = 0 \qquad (9)$$

$$EF_i = ES_i + d_i \qquad , i = 1,2, \dots, n \qquad (10)$$

$$ES_j = \max\{EF_i\} \qquad \forall_i \in p_j \;, j = 1,2, \dots, n \qquad (11)$$

$$d_0 = 0, d_{n+1} = 0 \qquad (12)$$

$$\sum_{j=1}^{n} y_{jk} = 1 \qquad , k = 1,2, \dots, k-1 \qquad (13)$$

$$\sum_{k=1}^{k} y_{jk} \leq 1 \qquad ,j = 1,2,\ldots,n \tag{14}$$

$$\sum_{k=1}^{k} P_k = U \quad k = 1,2,\ldots,k \tag{15}$$

$$P_k \geq 0 \; k = 1,2,\ldots,K \tag{16}$$

Equation (1) represents the objective function is to maximize the net present value of the project and the contractor is calculated according to the method of payment. Equation (2) represents the objective function the maximum completion time of activity n +1 that should be minimized. The constraint set (3) makes sure that all precedence relations are satisfied. The Constraints set (4) shows the completion time of project activities. Constraint (5) calculates maximum project completion. Constraint (6) calculates project planning horizon which is equal to all project activities. Constraint (7) ensures that the project be completed before project planning horizon. Constraints (8) is for applying renewable resource constraints and in each period summation of consumption of all activities from each resource in each time unit cannot exceed from maximum amount of that resource (Rk) in its relevant time unit. Constraints (9) expresses project starts time. Constraints (10) is related to transposition relations (without delay) between project activities. In a way that no activity can start before end of all its prerequisite activities and from the other hand projects activities are continuous. Constraints (11) shows that jth activity start time is equal or larger than its prerequisite activities end time. Constraints (12) shows that 0 and n+1 activities are virtual activities. Constraints (13) shows number of payments K for certain event m. constraint (14) ensures that one payment be allocated at the end of event. Constraint (15) ensures that summation of all payments are equal to project contractor price. Constraint (16) shows that payments value always be positive.

As mentioned before, we present the model from the contractor point of view. We also assume that this contractor has to pay the activities cost at the completion time of each activity; however it receives positive cash flows from client based on the project contract. All of various contracts can be defined by four basic payments model named Lump-sum payment, payment at event occurrences, payment at equal time intervals, and progress payment [25,27].

•*Lump-sum payment.* Here, the client pays the total payment to the contractor upon successful completion of the project.

$$\max z = \mathrm{CF}_{\mathrm{lsp}}(1+\alpha)^{-\mathrm{FT}_{\mathrm{n}}} - \sum_{\mathrm{j}=1}^{\mathrm{n}} \sum_{\mathrm{t}=\mathrm{EF}_{\mathrm{j}}}^{\mathrm{LF}_{\mathrm{j}}} \frac{\mathrm{CF}_{\mathrm{j}}}{(1+\alpha)^{\mathrm{t}}} \times \mathrm{x}_{\mathrm{jt}}$$

$$\mathrm{CF}_{\mathrm{lsp}} = \sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{CF}_{\mathrm{j}}^{+}$$

Lump-sum payment (LSP) at the completion of the project, The LSP model represents the ideal situation for the client—he makes a single payment to the contractor only at the end of the project. However, in general, this shifts the entire financial burden on the contractor, which may not be acceptable in some project environments. In consequence, the two parties often attempt to negotiate a method of payments over the duration of the project [28] [29].

• *Payment at event occurrences.* Payments are made at a set of event nodes. The problem is to determine the amount, location, and timing of these payments. Dayanand and Padman (1993, 1997) attacked the problem of simultaneously determining the amount, location, and timing of the payments by the client so as to maximize the contractor's NPV. They have dealt with this problem further from the perspective of the client (Dayanand and Padman, 1998). Later, Dayanand and Padman (1999) investigated the problem in the context of client and contractor negotiation stressing the need for a joint view. Ulusoy and Cebelli (2000) extended this payment model so as to include both the client and the contractor in a joint model. They introduce the concept of ideal solution, where the ideal solution for the contractor would be to receive the whole payment at the start of the project and for the client it would be a single payment at the completion of the project. They search for a solution where the client and the contractor deviate from their respective ideal solutions by an equal percentage. They call such a solution an equitable solution.

$$\max z = \sum_{j=1}^{n} CF_j^{+}\,(1+\alpha)^{-FT_j} - \sum_{j=1}^{n}\sum_{t=EF_j}^{LF_j} \frac{CF_j}{(1+\alpha)^t} \times x_{jt}$$

PAC is a very practical payment model, where the client pays the contractor for the completion of each activity of the project. Once an activity is finished, the contractor gets the amount of money equal to the cash flow associated with this activity [28] [29].

• *Progress payment.* The contractor receives payments at regular time intervals until the project is completed. The amount of payment is based on the amount of work accomplished since the last payment. Kazaz and Sepil (1996) presented a mixed-integer formulation of the progress payment with the objective of maximizing the NPV of the cash flows for the contractor. Sepil and Ortaç (1997) tested the performance of some heuristic procedures for resource-constrained projects with progress payments. They defined cash inflows occurring periodically as progress payments, and cash outflows as costs incurred whenever an activity is completed [28] [29].

$$\max z = \left(\sum_{P=1}^{H-1} P_P(1+\alpha_i)^{-PT} + P_H(1+\alpha_i)^{-FT_n}\right) - \sum_{j=1}^{n}\sum_{t=EF_j}^{LF_j} \frac{CF_j}{(1+\alpha)^t} \times x_{jt}$$

•*Payment at equal time intervals.* In this payment model, payments are made at predetermined equal time intervals over the duration of the project, and the final payment is scheduled on project completion. The amounts of the payments are either predetermined and fixed or are based on the amount of work accomplished since the last payment. Note that the number of payments in this payment model is known and fixed in advance, whereas in progress payment model this is not the case [28] [29].

$$\max z = \sum_{P=1}^{H} P_P\,(1+\alpha)^{-T_P} - \sum_{j=1}^{n}\sum_{t=EF_j}^{LF_j} \frac{CF_j}{(1+\alpha)^t} \times x_{jt}$$

3. The Model solution method

Project scheduling by desirable cost payment models with considering resource constraint problem, due to calculations complexities are considered as NP-Hard problems and solving them in big scale by linear programing and existing applications are not possible or require a lot of time. Therefore Meta heuristic methods are used for these kinds of problems. Also Meta heuristic methods are very efficient to solve complex optimization rather than precise algorithm and many other heuristic methods. In multi objective optimization, first we try to have a proper estimation of Parto Front and then obtained results can cover its parto and simultaneously emphasize to defeated solutions of each group and therefor both density and convergence of optimal parto solutions are emphasized.

3.1. Solution representation and schedule generation scheme

The solution representation is an important component of NSGA-II. It has to be designed such as to allow easy neighbour generation and fast calculation. It must also guarantee accessibility for the entire solution space. Various types of solution representations for the RCPSP have been proposed in the literature and are reviewed in the paper of Kolisch and Hartmann [30].

3.2. Decoding based on serial schedule scheme

In this paper, the serial schedule scheme is used to generate project plan from the chromosome. In the decoding based on serial schedule scheme by priority- based encoding, one activity in project from the decision set is selected with the priority rule, in which the activity with highest priority value is selected firstly, and scheduled at the earliest precedence. We select an activity obeyed feasible finish time and resource through the decoding of revised serial scheme, and the selected activity is removed from the decision set and put into the scheduled set with a selected mode and a selected way and the corresponding duration determined by both the selected mode and selected way. This, in turn, may replace a number of activities into the decision set, since all their predecessors are scheduled. The algorithm terminates at stage number n, when all activities are in the scheduled set.

This SGS alternates two operations of ‘‘start time assignment to activities’’ and ‘‘time incrementing’’ until all activities of the project, including the dummy finish activity, are scheduled. The procedure is initialised by fixing the time at Start time = 0 after scheduling and finishing the dummy start activity. The ‘‘start time assignment’’ operations are always performed when the time is fixed, and consist in consecutively testing the unscheduled activities one after the other following the list order. However, no activity can be tested before its list-predecessor has not been scheduled [30].

3.3. NSGA-II methodology

NSGA-II, proposed by Deb et al. (2002a), has been demonstrated to be one of the most efficient and popular algorithms for multi-objective optimization. NSGA-II (Deb et al. 2002a) is a multi-objective evolutionary algorithm that uses non-dominated sorting and a crowded-comparison approach to find a set of evenly distributed solutions to a multiobjective optimization problem [31]. NSGA-II is an improved version of the nondominated sorting genetic algorithm NSGA. Initially, a parent population P_0 is created randomly. Besides, this population is sorted based on the nondomination 1 concept. For each individual, fitness is assigned and it is equal to its non-domination rank. After that, a child population Q_0 is created from P_0 by performing binary tournament selection, crossover and mutation operations. From the first generation award, the procedure is different. First, the two populations P_t and Q_t are combined to form a population R_t of size 2N ($|P_t|$ = $|Q_t|$ = N). Second, a non-dominated sorting is used to classify the entire population Rt. Once, the non-dominated sorting is over, the population R_t becomes subdivided in several categories. All elements of the same category have the same non-domination rank. Each category is called a non-dominated front. After that, the new parent population P_{t+1} is filled with individuals of the best non-dominated fronts, one at a time. Since the overall population size is 2N, not all fronts may be accommodated in N slots available in the new population P_{t+1}. When the last allowed front is being considered, it may contain more solutions then the remaining available slots in P_{t+1}. Instead of discarding arbitrary some elements from the last front, it will be judicious to use niching strategy to choose individuals from the last front which reside in the least crowded region in this front. At this point, it may be practical to discuss the crowding distance (CD) assignment and the crowded tournament selection operator used for the NSGA-II selection. But, we will not plunge into too many details on that. Figure 1 illustrates the above mentioned mechanism [32]. The NSGA-II procedure is also shown in Fig. 1

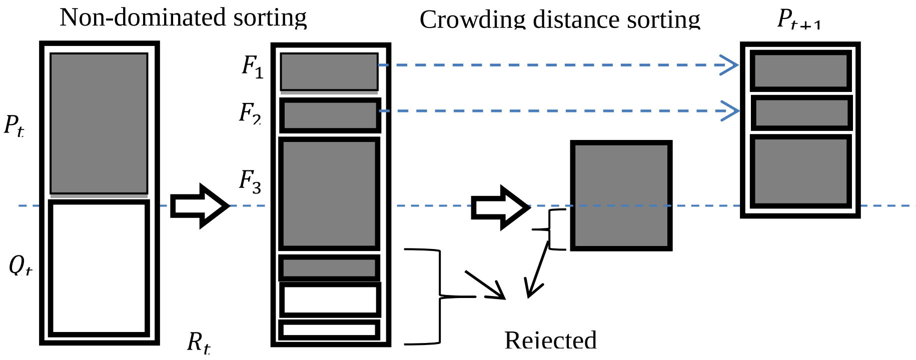

Fig. 1 NSGA-II Procedure

3.4. Create an initial population of solutions

In this research each operator produce a new population according to relevant percentage and main population size and finally after ranking all results population, population distance criteria are calculated and the results which have higher population quality and distance.

3.5. Creation an Pareto solutions for the intended problem

since project scheduling project based on cost payments models and considering resource constraint for Condensate refinery in Bandar Abbas has two objective and one final solution cannot be the problem solution A set of solutions are obtained and selecting final solution with regard to objective function and time importance and present net value from contractor point of view and a alternative is selected for execution. So according to partofront and domination, the problem solutions is obtained. These archive is in each algorithm repeat, will be updated. Updating method is designed in a way that the solution which has minimum distance from optimum boarder and also maximum distribution are stored in this archive. The size of archive is constant and all solution as Dominant and recessive are noted.

3.6. Updated the population

3.6.1. Crossover operator

p_c parameter is considered as intersection probability and for selecting parents chromosome in Intersection , we repeat the following process pop-size times. For i=1, 2, …, pop-size, we use three intersection types as one point and two point unified intersection. This process is described as follow. First we must select a stochastic number in one point intersection in [0, N-1] and then we break both parents in this point and by moving their sequence, we produce two new child. Then in two point intersection we select two different random number in [0, N-1] interval and we break both generator in these two points and by moving points between two parts of both generators, we produce two child and then in unified intersection we produce two random numbers like V in [0, 1] interval and if $V \leq p_c$ (in proposed algorithm is equal to 0.9), x_i chromosome is selected as a parent in intersection operation. Then we reach the number of (pop-size) p_c parents for intersection operation. We number them again from the start and specify them by prime sign (as $x_1^{'}, x_2^{'}$, ...) . in the next phase if we want to have an intersection between two parents like $x_1^{'} = (x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)})$ و $x_2^{'} = (x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)})$ we must first produce a random number in [0,1] interval and then do the intersection operation by using the following equation which are new chromosome and named as child chromosome and are signed by ". if both Childs are feasible, then we replace parents with them. If one of the parents are possible then we keep that and repeat intersection operations to reach another possible child. If both of them are not possible, we repeat the operation to two possible child.

3.7. Mutation operator

p_m parameter is considered as probability of mutation. Parent chromosome are selected by the same method which was mentioned in intersection operations. Parent chromosome are selected which are almost as many as (pop -size) p_m. Then mutation operation is applied as the following method. In this research, gussion method is used for producing mutants that for X variables which is X_{min} and X_{max}, new variable must have normal distribution with zero mean and σ^2 variance. That $X^{'} = X + \Delta X$ and $\Delta X \sim N(X, \sigma^2)$. This means that an standard value is produced and multiplied by σ^2 and summed by X value and σ^2 is equal to $0.1 * (V_{max} - V_{min})$. Which σ^2 is equal to mutation steps. Therefor mu (mutation ratio) % is selected randomly and to have an integer value for mutants and atleast one case be found, value are rounded up.

3.8. NSGA-II with modified non-dominated sorting

Altering the non-dominated sorting procedure can also affect the progress towards the Paretooptimal front. Examples of this approach include ε-MOEA, a proper domination, fuzzy domination (Deb 2008) and quick sort (Zheng *et al.* 2004). The aims of these improvements are to reduce the time to converge to the Pareto front and to reduce the computational complexity, but the subsequent penalty is that part of the real Pareto front may be excluded [31].

3.9. NSGA-II with modified crowding distance

Once the non-dominated sort is complete, the crowding distance is assigned. As the individuals are selected based on rank and crowding distance all the individuals in the swarm are assigned a crowding distance value. Crowding distance is allocated front wise and comparing the crowding distance between two individuals in different front is meaningless [33]. The crowding-distance computation requires sorting the population according to each objective function value in ascending order of magnitude. Thereafter, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. This calculation is continued with other objective functions. The overall crowding-distance value is calculated as the sum of individual distance values corresponding to each objective. Each objective function is normalized before calculating the crowding distance. Although Fig. 2 illustrates the crowding-distance computation for two objectives, the procedure is applicable to more than two objectives as well.

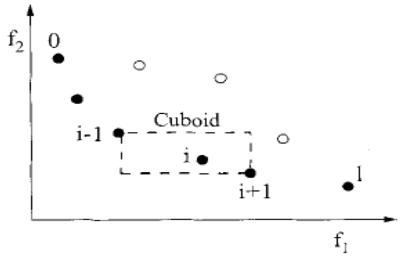

Fig. 2 Crowding-distance calculation.

```
Crowding-distance-assignment(I)
l = $|I|$                                  number of solutions in $I$
for each i, set $I[i]_{distance} = 0$      initialize distance
For each objective m
I=sort(I,m)                                sort using each objective value
$I[1]_{distance} = I[l]_{distance} = \infty$   so that boundary points are always  Selection
For i=2 to ($l$-1)                         for all other points
        $I[i]_{distance} = I[i]_{distance} + (I[i+1].m - I[i-1].m/(f_m^{max} - f_m^{min})$
```

Fig.3 Crowding-distance-assignment

4. Computational Results

Proposed method presented in this research is coded by using multi objective genetic algorithm proposed in MATLAB software. In this part, input parameters which consider general and control variables is presented and results of proposed algorithm solving are discussed and the proposed multi objective genetic algorithm is validated by GAMS. In table (1), required information for Bandar Abbas Gas Condensate Refinery Construction Project including activities time, prerequisite relations, required resources for activities and positive and negative financial flows for activities. In this project, it is assumed that there is no limit in non-renewable resources and maximum values of these resources are as follow: human resources equal to 150 (R1=150) and 100 unite machinery (R2=100).

Table 1 information about the activities of the installation of steel structures

Activities	duration	Prerequisite activities	Resource requirements	CF_j^-	CF_j^+

1	0	-	-	-	0	0
2	16	1	20	23	24800	55800
3	45	1	41	30	104400	234900
4	30	2	27	23	50700	114075
5	15	2	33	29	31650	71212.5
6	60	2	38	34	147600	332100
7	25	2	32	22	43500	97875
8	16	2	23	15	19360	43560
9	31	2	34	18	48980	110205
10	30	2	36	19	50100	112725
11	61	4	59	47	215330	484492.5
12	45	4,5	44	38	125100	281475
13	25	5,12	39	18	42000	94500
14	45	5	37	26	91800	206550
15	15	5	21	16	18300	41175
16	16	5	29	25	29280	65880
17	45	6	39	40	125100	281475
18	25	2,3,9	55	34	70000	157500
19	439	2,7,8	97	78	2569600	5781600
20	15	13,14,15	34	14	20700	46575
21	16	11,20	12	16	17680	39780
22	45	13,20	23	45	121950	274387.5
23	46	16	19	21	67210	151222.5
24	30	16	34	23	54900	123525
25	45	16,24	23	13	49950	112387.5
26	25	16,25	27	23	49010	110272.5
27	61	25	43	34	158720	357120
28	45	24	23	13	54390	122377.5
29	25	20,22	18	12	22080	49680
30	504	23,24,25,26	120	79	3321050	7472362.5
31	16	27,30	23	12	18020	40545
32	25	26,28,30	36	22	45500	102375
33	30	23,30	19	21	42900	96525
34	30	17,19	21	28	54600	122850
35	45	3,19	65	45	166850	375412.5
36	241	30	95	96	1654900	3723525
37	0	21,28,32,33,34,35,36	-	-	0	0

About applying the algorithm, it is noted that after entering general parameters of the problem, a initial solution is created by parallel scheduling production method which is considered as initial input of proposed multi objective genetic algorithm. After defining and initialing control parameters, multi objective genetic algorithm is applied. Multi objective genetic algorithm is repeated for 200 times. That produced as 100 population which forms initial population and each member of this population have same fitness that shows the adaption with objective function. Now intersection and mutation operators is used for optimizing fitness of population members and updating population. As mentioned In previous sections three operator types (one point, two point and unified method) is used and then for each operators a stochastic value equal to 0.1 and 0.3 and

0.7 is assigned in a way that summation of three operator be equal to one. Then among those, one operator is selected by Roulette wheel method and by using gussian mutations, mutation population is created. In each step of this approach, fitness of objective is analyzed and population members are compared with each other and since the problem is multi objective, a level of pareto front is considered (according to ranking and solutions swarm distances). Finally a set of solutions including a list of activities and executing order of activities and start and end of them and maximum time of project completion and project NPV is introduced as algorithm output and final solution. A level of final solution pareto front for the problem is depicted in figures (2), (3), (4), (5) based on mentioned payments by NSGA-II algorithm.

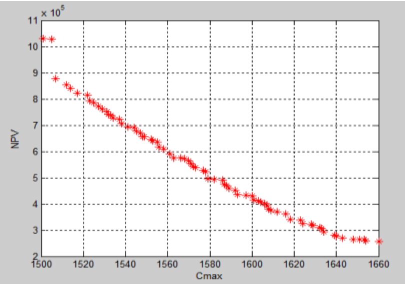

Fig. 4 final Solutions Pareto Front Based on the LSP (J=35)

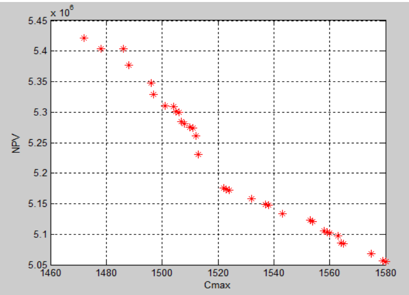

Fig. 5 final Solutions Pareto Front Based on the PEO (J=35)

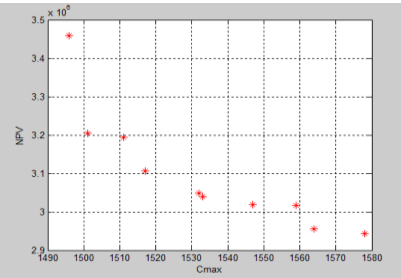

Fig. 6 final Solutions Pareto Front Based on the ETI (J=35)

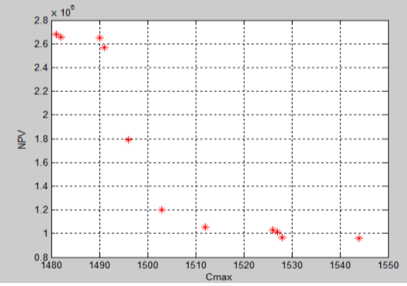

Fig. 7 final Solutions Pareto Front Based on the PP (J=35)

As the can be observed in the above figure, Surface of the front Pareto of generated solutions by the algorithm is based on payment type. According to these results, the contractor is faced with a set of answers which could be due to the importance of each objective function (Maximum completion time and maximizing the NPV of the project) one alternative that represents a method to select the mode is operational activity.

5. Validation of the proposed algorithm

For proving efficiency of the meta heuristic algorithm, solution of the algorithm is compared with solution of GAMS software. Therefor to prove efficiency of the proposed method, several sample problems in small scale including sub sets of real problem (with 10, 14, 18 and 20 activity) is solved by the proposed algorithm and GAMS software. Results and duration of executing NSGA-II algorithm and GAMS software is shown and analyzed compared in table (2) to (4). To compute means of differences percentage of the results of GAMS and NSGA-II, we use the following formulation.

Average difference percentage= ((NSGA-II result – GAMS result) / GAMS result) * 100

Table 2 Results from GAMS software and NSGA-II

Type of payment / Problem		LSP		PEO		ETI		PP	
		C_{max}	NPV	C_{max}	NPV	C_{max}	NPV	C_{max}	NPV
J=10	NSGA-II	15	4297.59	15	4335.43	14	3840.09	16	1773.26
	GAMS	14	4304.09	14	4436.2	14	3840.34	14	1948.15
J=14	NSGA-II	85	3959.68	85	4214.3	85	3954.71	87	2772.83
	GAMS	83	3969.90	83	4214.58	83	3962	83	2780.51
J=18	NSGA-II	190	73489.38	188	93336.19	196	77852.39	188	78445.25
	GAMS	187	73816.44	187	93427.17	187	78435.38	187	78611.21
J=20	NSGA-II	64	21516.57	64	22502.47	65	22079.6	66	12407.68
	GAMS	63	21595.11	63	22519.73	63	22100.15	63	13612.09
J=25	NSGA-II	67	18784.05	66	19516.25	68	18533.39	65	12968.16
	GAMS	65	18834.45	64	19529.84	65	18555.29	64	13412.05

Table 3 percentage Average difference between Results from GAMS software and NSGA-II

Type of payment / Problem	LSP		PEO		ETI		PP	
	C_{max}	NPV	C_{max}	NPV	C_{max}	NPV	C_{max}	NPV
J=10	7.1	0.15	7.1	2.2	0	0	0	1.28
J=14	2.4	0.25	2.4	0.006	2.4	0.18	4.8	0.27
J=18	1.6	0.44	0.53	0.097	4.8	0.74	0.53	0.21
J=20	1.5	0.36	1.5	0.07	3.1	0.09	4.7	8.8
J=25	3.7	0.26	3.1	0.06	4.6	0.11	1.5	3.3
The average percentage difference	2.7	2.3	0.22	2.9	0.48	2.9	0.29	3.2

Table 4 implementation time from the GAMS software and NSGA-II

Problem	J=10	J=14	J=18	J=20	J=25	J=35
NSGA-II	228.76	252.48	282.55	265.6	327.09	707.15
GAMS	414.31	923.63	2475.43	3627.94	5162.57	-

Completion time of the project and reaching to a solution time for different problems with different activities with different payments methods is presented in table (2). This problem is solved by meta heuristic simulation method, too. Result differences for proposed method and precise method is presented in table (3) which their differences are very small and less than 3%. Also based on table (4) and figure (6), time to reach a solution is constant in the proposed method but it increases as a quadratic function above results shows that the proposed method is a convergence to an optimum and solution algorithm.

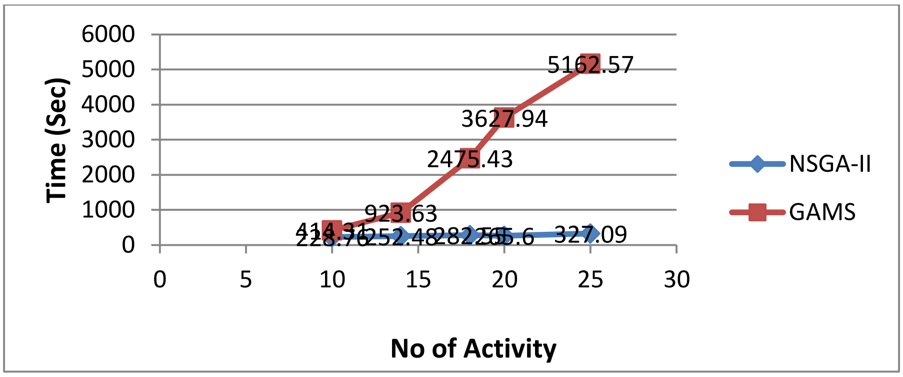

Fig. 8 Time of sample problems solving in the proposed algorithm and GAMS software

6. Conclusion

In this research, scheduling of part of refinery construction plan is analyzed by using a meta heuristic method. In this model project resources is assumed to be renewable. Objects is considered are as follow. Minimizing project completion time and maximizing present net value of the project. It is impossible to reach global or local solution using classic optimization methods due to many constraints in the model and multiple objective of the problem. Since the problem has complexity in calculation time and in other words it is classified as NP hard problems, in this research, a NSGA-II multi objective meta heuristic is used for optimized scheduling. Since every algorithm must be validated before use and the current study is applied for a real project which is progress, we cannot compare algorithm results with project results therefor to prove efficiency of the algorithm, the algorithm results are compared by results of solving the problem solved by GAMS software in some problems in small scales. The differences of proposed algorithm with precise algorithm is shown in table (3), represent that results are almost similar and smaller than 3%. Also according to table (4), time to reach a solution in proposed method is constant but in GAMS software increases as quadratic function. These results shows that the proposed method is a convergence algorithm to optimal and efficient solution.

References

[1]A.A. Najafi, S.T. AkhavanNiaki, M. Shahsavar, A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations, Computers & Operations Research. 36 (2009) 2994 – 3001.

[2]M.A. Santos, A.P. Tereso, ON THE MULTI-MODE, MULTI-SKILL RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM (MRCPSP-MS), 2nd International Conference on Engineering Optimization September. (2010) 6 – 9.

[3] S. Hartmann, D. Briskorn, A survey of variants and extensions of the resource-constrained project scheduling problem, European Journal of Operational Research. 207 (2010) 1–14.

[4] L. Liu, S.A. Burns, C.W. Feng, Construction time–cost trade-off analysis using LP/IP hybrid method. Journal of Construction Engineering and Management.121 (1995) 446–454.

[5]De P. James Dunne, E. Ghosh, J.B. Wells, C.E., The discrete time-cost tradeoff problem revisited. European Journal of Operational Research, 81, 225–238, 1995.

[6] R.T. Harvey, J.H. Patterson, An implicit enumeration algorithm for the time/cost tradeoff problem in project network analysis. Foundations of Control Engineering.4 (1979)107–117.

[7] E.L. Demeulemeester, W.S. Herroelen, S.E. Elmaghraby, Optimal procedures for the discrete time/cost trade-off problem in project networks. European Journal of Operational Research. 88 (1996) 50–68.
[8] E. Eshtehardian, A. Afshar, R. Abbasnia, Fuzzy-based MOGA approach to stochastic time–cost trade-off problem. Automation in Construction.18 (2009) 692–701.
[9] H.R. Tareghian, S.H. Taheri, A solution procedure for the discrete time, cost and quality tradeoff problem using electromagnetic scatter search. Applied Mathematics and Computation.190 (2007) 1136–1145.
[10] A. H. Russell, Cash Flows in Networks, Management Science. 16(5) (1970) 357-373.
[11] M. Mika, G. Waligora, J. Wezglarz, 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 (2005) 639-668.
[12] K. Brinkmann, K. Neumann, Heuristic Procedures for Resource-Constrained Project Scheduling with Minimal and Maximal Time Lags: The Resource Leveling and the Minimum Project-Duration Problem, Journal of Decision Systems. 5 (1996) 129-55.
[13]R. Sonmez, Ö.H. Bettemir, A hybrid genetic algorithm for the discrete time–cost trade-off problem, Expert Systems with Applications. 39 (2012) 11428–11434.
[14] O. Icmeli, S.S. Erenguc, The resource constrained time cost tradeoff project scheduling problem with discounted cash flows, Journal Operation. Managmente. 14 (1996) 255–275.
[15] M.U. Homburg, Cash flow shortage as an endogenous bankruptcy reason, Journal Banking Finance. 29(6) (2005) 1509–1534.
[16] S.S. Liu, C.J. Wang, Resource-constrained construction project scheduling model for profit maximization considering cash flow, Automation in Construction. 17 (2008) 966–974.
[17]Z. He, Y. Xu, Multi-mode project payment scheduling problems with bonus–penalty structure, European Journal of Operational Research. 189 (2008) 1191–1207.
[18]G. Waligora, Discrete–continuous project scheduling with discounted cash flows—A tabu search approach, Computers & Operations Research. 35 (2008) 2141 – 2153.
[19]S. Hartmann, D. Briskorn, A survey of variants and extensions of the resourceconstrained project scheduling problem, European Journal of Operational Research. 207 (2010) 1–14.
[20]A.A. Najafi, A. Niaki, A genetic algorithm for resource investment problem with discounted cash flows, Applied Mathematics and Computation. 183 (2006) 1057–1070.
[21] Z. He, N. Wang, T. Jia, Y. Xu, Simulated annealing and tabu search for multi-mode project payment scheduling, European Journal of Operational Research 198 (2009) 688–696.
[22] M. Vanhoucke, A scatter search heuristic for maximizing the net present value of a resource-constrained project with fixed activity cash flows. International Journal of Production Research. 48 (2010)1983-2001.
[23] B.A. Nadjafi, Sh. Shadrokh, An Algorithm for the Weighted Earliness-Tardiness Unconstrained Project Scheduling Problem. Journal of Applied Sciences. 8 (2009) 1651-1659.
[24] B.O. Odedairo, V. Oladokun, Relevance and Applicability of Multi-objective Resource Constrained Project Scheduling ProblemETASR-Engineering, Technology & Applied Science Research. 1(6) (2011) 144-150.
[25] M. Mika, G. Waligora, J. Wezglarz, Simulated annealing and tabu search for multimode resource-constrained project scheduling with positive discounted cash flows and different payment models, European Journal of Operation Research, forthcoming. 164(2005) 639-668.
[26] S.P. Chen, M.J. Tsai, Time–cost trade-off analysis of project networks in fuzzy environments, European Journal of Operational Research. 212 (2011) 386–397.
[27] G. Ulusoy, F.S. Serifoglo, S. Sahin, Four Payment Models for the Multi-Mode Resource Constrained Project Scheduling Problem with Discounted Cash Flows, Annals of Operations Research. 102 (2001) 237-261.
[28]G. Waligora, Discrete–continuous project scheduling with discounted cash flows—A tabu search approach, Computers & Operations Research. 35 (2008) 2141 – 2153.

[29]M. Mika, G. Walig, J. Wezglarz, 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 (2005) 639–668.
[30] K. Bouleimen, H. Lecocq, A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version, European Journal of Operational Research.149 (2003) 268–281.
[31]H. Ghiasi, D. Pasini, L. Lessar, A non-dominated sorting hybrid algorithm for multi-objective optimization of engineering problems , Engineering Optimization. 43(1) (2011) 39–59.
[32]S. Bechikh, N. Belgasmi, L.B. Said, Kh. Ghédira, PHC-NSGA-II: A Novel Multi-objective Memetic Algorithm for Continuous Optimization, 20th IEEE International Conference on Tools with Artificial Intelligence. (2008).
[33] H. Shayeghi A. Ghasem, Application of MOPSO for Economic load Dispatch Solution with Transmission Losses, International Journal on Technical and Physical Problems of Engineering (IJTPE). 4 (2012).