Iranian Journal of Management Studies (IJMS) *http://ijms.ut.ac.ir/*
Vol. 8, No. 4, October 2015 *Print ISSN: 2008-7055*
pp: 653-685 *Online ISSN: 2345-3745*

A multi-objective resource-constrained optimization of time-cost trade-off problems in scheduling project

Mostafa Zareei*, Hossein Ali Hassan-Pour

Faculty of Industrial Engineering, Imam Hossein (AS) University, Tehran, Iran

(Received: 8 August 2014; Revised: 22 December, 2014; Accepted: 24 December, 2014)

Abstract

This paper presents a multi-objective resource-constrained project scheduling problem with positive and negative cash flows. The net present value (NPV) maximization and making span minimization are this study objectives. And since this problem is considered as complex optimization in NP-Hard context, we present a mathematical model for the given problem and solve three evolutionary algorithms; NSGA-II, MOSA and MOPSO are applied to find the set of Pareto solutions for this multi-objective scheduling problem. In order to show performance of the algorithms, different metrics are applied and comparisons between the two algorithms are also considered. The computational results for a set of test problems taken from the project scheduling problem Bandar Abbas Gas condensate Refinery project and library are presented and discussed. Finally, the computational results illustrate the superior performance of the NSGA-II, MOSA and MOPSO algorithm with regard to the proposed metrics. In order to solve proposed method from NSGA-II algorithm, the results are compared with GAMS software in some problems. The proposed method is a Converge to the optimum and efficient solution algorithm.

Keywords

Comparative indicators of evolutionary algorithms, MOSA and MOPSO algorithm, NSGA-II, Payment patterns, Project scheduling, Resource constraints.

* **Corresponding Author, Email:** mozare66@yahoo.com

Introduction

Resource-constraint project scheduling problem (RCPSP) is a class of project scheduling that one's activities should be scheduled subject to precedence and resource constraints and it is proven to be NP-hard (Aboutalebi1 *et al*., 2012). Minimization of project duration is often used as an objective of a general project scheduling problem while other objectives such as Maximizing of net present value of cash flows, and leveling of resource usage are also considered. Resources involved in a project can be single or multiple varieties, and can be renewable or nonrenewable (Ritwik & Paul, 2013). The time–cost tradeoff problem in project management originates when activity time can be reduced with some extra direct cost (Jongyul *et al*., 2012). Time Cost Trade off analysis is the compression of the project schedule to achieve a more favorable outcome in terms of project duration, cost, and projected revenues. The objectives of the Time Cost Trade off analysis are compressing the project to the optimum duration which minimizes the total project cost (Rifat & Önder Halis, 2012).

Another important type of objective emerges if cash flows occur while the project is carried out. Cash outflows are induced by the execution of activities and the usage of resources. On the other hand, cash inflows result from payments due to the completion of specified parts of the project. Typically, discount rates are also included. Note that cash flows related to activity j might occur at several points in time during execution of j. However, they can easily be compounded to a single cash flow at the beginning or the end of j. These considerations result in problems with the objective to maximize the net present value (NPV) of the project which subject to the standard RCPSP constraints (Shu-Shun & Chang-Jung, 2008). Project payment scheduling problem involves how to schedule progress payments effectively 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.

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 (Zhengwen & Yu, 2008). Time-Cost optimization (TCO) problem has been extensively examined by a number of research studies. Various approaches have been proposed for optimizing construction time and cost including (1) heuristic methods (Moselhi, 1993; Siemens, 1971); (2) mathematical programming (Liu *et al*., 1995; Moussourakis & Haksever, 2004); and (3) meta-heuristic methods. Mathematical programming such as linear programming is suitable for problems with linear time-cost relationships, but they often fail to solve the problem with discrete time-cost relationships (Feng *et al*., 1997). Moreover, it requires a lot of computational efforts to solve a large scale project network. Heuristic methods are able to overcome such limitation of a large scale problem, but fail to guarantee optimal solutions. Therefore, many research studies have focused on utilizing meta-heuristic methods in time-cost tradeoff analysis to overcome the limitation of heuristic methods and mathematical programming.

Liu *et al*. (1995) have developed optimization model using a hybrid method that integrates linear and integer programming. Linear programming was used to find lower bounds of the solutions, and then integer programming was used to obtain the exact solution. The integer programming was then used to minimize total project cost with the constraints of activity precedence and the selection of a single resource utilization option for each activity. The hybrid model was developed using Microsoft Excel to provide a construction planner with an efficient means of analyzing time-cost trade-off decisions.

Zheng *et al*. (2005) have developed GA-based multi-objective optimization model that simultaneously minimizes time and cost. In order to overcome the problem of genetic drift, the model utilized

Pareto ranking, niche formation, and adaptive mutation rate. The genetic drift occurs when GA converge to a single peak due to the stochastic errors during processing. The model incorporated which modified adaptive weight approach (MAWA) to exert a search pressure in GA. Pareto ranking mechanism overcomes the limitation of traditional proportional selections. Niche formulation is useful for stabilizing multiple subpopulations that arise along the Pareto-optimal front, and thereby it maintains diversity of population.

Xiong and Kuang (2008) have used ant colony optimization (ACO) algorithm to solve time-cost tradeoff problem. The modified adaptive weight approach (MAWA) proposed by Zheng *et al*. (2004) was incorporated in the model to generate optimal time-cost tradeoff curve. The performance of developed model was compared with GA by two test examples. The results showed that ACO-based multi-objective approach provides an attractive alternative to solving construction time-cost optimization. El-Rayes and Kandil (2005) have presented the multi-objective optimization model for time-cost-quality tradeoff analysis. Multi-objective genetic algorithm was used in the model to generate the optimal solutions for those conflicting objectives. The multi-objective genetic algorithm was implemented in the model to search for optimal resource utilization options for each activity that provides minimum project cost and time while maximizing quality performance. The model also provided visualization of tradeoffs among time, cost, and quality for the analyzed example.

Najafi and Niaki (2006) consider a cash flow related to a subset of activities, that is, a cash flow is initiated when the last activity of the subset is finished. Furthermore, the cash flow associated with activity j might depend on the mode chosen for j as lined out in Icmeli and Erenguc (1996). Smith-Daniels *et al*. (1996) propose the objective to maximize the discounted amount of cash available in each period. This amount is influenced by cash flows associated with each activity. Ulusoy and Cebelli (2000) investigate the negotiation process to find the timing of payments and the amount of each specific payment between a client and a contractor. Obviously, the client seeks to minimize the NPV while the contractor aims at maximizing it. The objective in Ulusoy and Cebelli (2000) is to find the payment structure which minimizes each party's loss in comparison to the respective ideal payment structure. Dayanand and Padman (2001) treat a similar problem, but restrict themselves to the client's point of view. The client might associate a specific value with each event (starting or completion of a job). Cash outflows can be assigned to each event having a positive value. The problem is to find a project schedule and decide cash outflows to happen at a given number of events.

The total outflow might exceed the total value of finished activities at no point of time. The objective is to find a solution such that discounted cash inflow (associated with finishing the project) minus total discounted cash outflow will maximize. Dorner *et al*. (2008) employ three objectives within a variant of the time-cost tradeoff problem. The first objective is a function of the project make span, while the second and the third are functions of the monetary and non-monetary costs for crashing the activities, respectively. Afshar *et al*. (2009) proposed Non-dominated Archiving ACO (NA-ACO) algorithm in which all ant colonies are initiated by the same number of ants, and arbitrary order is given to the colonies. Ants in a certain colony simultaneously explore a solution according to the objective assigned to that colony. If there is an improvement, the optimal path is updated. The progress payment model corresponds to what Dayanand and Padman (1998) refer to as the periodic payment model. Dayanand and Padman provide mixed integer linear programming formulations for the so-called basic client, equal time intervals and periodic payment models. They provide insights about the characteristics of optimal payment schedules obtained with each model. Kim *et al*. (2012) present an improved genetic algorithm to solve a multi-mode resource-constrained discrete time–cost tradeoff problem; which is applicable to special knowledge intensive projects, and does not consider project activities' quality.

Zhou (2011) has used ant colony algorithm to trade-off costs and time. The problem is considered as a multimode discrete and the objective function as the sum of the direct and indirect costs which indirect cost is defined as a linear function of project time (Xu, 2011). Aladini *et al*. (2011) have proposed a multi-objective ant colony optimization model for minimizing the project direct cost by calculating discounted cash flow to solve costs and time tradeoff problem considering value of money over time (considering the discount rate), and importance of discounted cash flow analysis for owners and contractors. Vanhoucke (2010) has presented a scatter search algorithm for project scheduling problem with constrained resource and with discounted cash flows. He has assumed payment in connection with the implementation of the project activities as a constant parameter and has developed a heuristic optimization method for maximizing the project net present value of the objective function with respect to priority and renewable resources constraints (Khalilzadeh *et al*., 2011).

Table 1. Literature of multi-objective optimization of time-cost trade-off problems in project scheduling

NO.	Name	Years	Using of Way Type
1	Liu *et al*.	1995	Have developed optimization model using a hybrid method that integrates linear and integer programming to minimize total project cost with the constraints of activity precedence and the selection of a single resource utilization option for each activity.
2	Dayanand and Padman	1998	Provide mixed integer linear programming formulations for the so-called basic client, equal time intervals and periodic payment models.
3	El-Rayes and Kandil	2005	Have presented the Multi-objective genetic algorithm for time-cost-quality tradeoff analysis.
4	Zheng *et al*.	2005	Have developed GA-based multi-objective optimization model that simultaneously minimizes time and cost. The model incorporated modified adaptive weight approach (MAWA) to exert a search pressure in GA.
5	Najafi and Niaki	2006	consider a cash flow related to a subset of activities, that is, a cash flow is initiated when the last activity of the subset is finished
6	Xiong and Kuang	2008	have used ant colony optimization (ACO) algorithm to solve time-cost tradeoff problem
7	Afshar *et al*.	2009	Proposed Non-dominated Archiving ACO (NA-ACO) algorithm in which all ant colonies are initiated by the same number of ants and arbitrary order is given to the colonies.
8	Vanhoucke	2010	Has presented a scatter search algorithm for project scheduling Problem with constrained resource and with discounted cash flows.
9	Zhou	2011	Has used ant colony algorithm to trade-off costs and time.
10	Kim *et al*.	2012	Present an improved genetic algorithm to solve a multi-mode resource-constrained discrete time–cost tradeoff problem

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, that is 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,…,n) executed is d_j. There are R renewable resources. The number of available units of renewable resource k= (k=1,2,…,R) is R_k. Each activity j is executed requiring r_{jk} units of renewable resource k= (k=1,2,…,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 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: Setting of all predecessors of activity j
R: Number of renewable resources
R_k: Number of available units of renewable resource k, k=1,2,…R
r_{jk}: Number of units of renewable resource k required by activity j executed
α: Discount rate
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

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

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

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

$$\max NPV \tag{1}$$

$$\min c\max = \max(\sum_{t=EF_j}^{LF_j} tX_{jt}) \tag{2}$$

st:

$$\sum_{t=EF_j}^{LF_j} tx_{jt} \leq \sum_{t=EF_j}^{LF_j} (t-d_j)x_{jt} \forall_{j,w} \in p_j \tag{3}$$

$$C_j = \sum_{t=EF_j}^{LF_j} tX_{jt} \forall_j = 1,2,...,n \tag{4}$$

$$C\max \geq C_j \forall_j = 1,2,...,n \tag{5}$$

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

$$C\max \leq T \tag{7}$$

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

$$ES_1 = 0 \tag{9}$$

$$EF_i = ES_i + d_i, i = 1,2,\dots,n \tag{10}$$

$$ESj = \max\{EF_i\} \forall_i \in p_j, j = 1,2,\dots,n \tag{11}$$

$$d_0 = 0, d_{n+1} = 0 \tag{12}$$

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

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

$$\sum_{k=1}^{k} P_k = u, k = 1,2,\dots,k \tag{15}$$

$$P_k \geq 0, k = 1,2,\dots,k \tag{16}$$

Equation (1) represents the objective function which 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 is completed before project planning horizon. Constraints (8) are 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) are related to transposition relations (without delay) between project activities. In such a way that no activity can start before the end of all its prerequisite activities, and on the other hand, projects activities are continuous. Constraints (11) show that j the activity start time is equal or larger than its prerequisite activities end time. Constraints (12) show that 0 and n+1 activity are virtual activities. Constraints (13) shows number of payments K for certain event m. constraint (14) ensures that one payment is allocated at the end of event. Constraint (15) ensures that summations of all payments are equal to project contractor price. Constraint (16) also shows that payments values always are positive.

In real life situations, there are at least two parties involved in the project: the client, that is, the owner of the project and the contractor who undertakes the execution of the project. The legal basis of the execution of a project is provided by a contract organizing aspects of the interactions between the stakeholders. There are a large number of contract types with considerable amount of detail involved. A treatise of different contract types is given by Herroelen *et al*. (1997). For the purposes of this paper, we are interested in the basic payment structures specified in the contracts. Four types of payment scheduling models are of particular interest in practice: Lump-sum payment, payment at event occurrences, payment at equal time intervals, and progress payment.

Lump-sum payment (LSP) is one of the more commonly used payment structures in the literature. Here, the whole payment is paid by the client to the contractor upon successful termination of the project (Seifi & Tavakkoli-Moghaddam, 2008). 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 (Marek *et al*., 2005).

$$\max z = CF_{lsp}(1+\alpha)^{-FT_n} - \sum_{i=1}^{n}\sum_{t=EF_i}^{LF_j}\sum_{m=1}^{M_j}\frac{CF_{jm}}{(1+\alpha)^t}\times X_{jmt}$$
$$CF_{lsp} = \sum_{j=1}^{n} CF_j^+ \qquad (17)$$

In the payments at event occurrences (PEO) model, payments are made at predetermined set of event nodes. The problem is to determine the amount and timing of these payments (Seifi & Tavakkoli-Moghaddam, 2008). PEO is a very reasonable model, where the contractor gets his payments for successful completion of each activity (Marek *et al.*, 2005).

$$\max z = \sum_{j=1}^{n} CF_j^{+} (1+\alpha)^{-FT_j} - \sum_{j=1}^{n} \sum_{t=EF_i}^{LF_j} \sum_{m=1}^{M_j} \frac{CF_{jm}}{(1+\alpha)^t} \times X_{jmt} \tag{18}$$

In the equal time intervals (ETI) model, the client makes H payments for the project. Of these payments, the first (H-1) are scheduled at equal time intervals over the duration of the project, and the final payment is scheduled on project completion (Seifi & Tavakkoli-Moghaddam, 2008). In the ETI model the client and the contractor agree about the number of payments over the course of the project. The payments are then made at equal time intervals (Marek *et al.*, 2005).

$$\max z = \sum_{P=1}^{H} P_p (1+\alpha)^{-T_P} - \sum_{j=1}^{n} \sum_{t=EF_i}^{LF_j} \sum_{m=1}^{M_j} \frac{CF_{jm}}{(1+\alpha)^t} \times X_{jmt} \tag{19}$$

In the progress payment (PP) model, the contractor receives the project payments from the client at regular time intervals until the project is completed. For example, the contractor might receive at the end of each month a payment for the work accomplished during that month multiplied by a profit rate agreed upon by both the client and the contractor (Seifi & Tavakkoli-Moghaddam, 2008). A similar situation concerns the PP model, where the payments are also made at regular time intervals, but in this case the two parties agree about the length of this interval, not the number of payments (Marek *et al.*, 2005).

The difference between the ETI and PP models is that in the latter case the number of payments is not known in advance.

$$\max z = \left(\sum_{P=1}^{H-1} P_p (1+\alpha_i)^{-PT} + P_H (1+\alpha_i)^{-FT_n}\right) - \sum_{j=1}^{n} \sum_{t=EF_i}^{LF_j} \sum_{m=1}^{M_j} \frac{CF_{jm}}{(1+\alpha)^t} \times X_{jmt} \quad (20)$$

Schedule Generation Schemes

Schedule generation schemes (SGS) are the core of most heuristic solution procedures for the RCPSP. SGS start from scratch and build a feasible schedule by stepwise extension of a partial schedule. A partial schedule is a schedule where only a subset of the n+2 activities have been scheduled. The so-called serial SGS performs activity incrimination and the so-called parallel SGS performs time-incrimination.

Serial schedule generation scheme

We begin with a description of the serial SGS. It consists of g = 1… n stages, in each of which one activity is selected and scheduled at the earliest precedence and resource feasible completion time. Associated with each stage, g is two disjoint activity sets. The scheduled set S_g comprises the activities which have been already scheduled; the eligible set D_g comprises all activities which are eligible for scheduling. Note that the conjunction of S_g and D_g does not give the set of all activities J because, generally, there are so-called ineligible activities, that is activities which have not been scheduled and cannot be scheduled at stage g because not all of their predecessors have been scheduled. Consider $R_k(t)=R_k-\Sigma_{j\varepsilon A(t)}r_{j,k}$ is the remaining capacity of resource type k at time instant t and $F_g = \{F_j | j \in S_g\}$ is the set of all finish times, and further is $D_g = \{j \in J \backslash S_g | P_g \subseteq S_g$ the set of eligible activities. The initialization assigns the dummy source activity j = 0 a completion time of 0 and puts it into the partial schedule. At the beginning of each step g, the decision set D_g, the set of finish times F_g, and the remaining capacities $R_k(t)$ at the finish time's $t\in F_g$are calculated. Afterwards, one activity j is selected from the decision set. The finish time of j is calculated by first determining the earliest precedence feasible finish time EF_j and then calculating the earliest (precedence-and) resource-feasible finish time F_j within $[EF_j, LF_j]$. LF_j denotes the latest finish time as calculated by backward recursion (cf. Elmaghraby, 1977) from an upper bound of the project's finish time T (Möhring & Stork, 2000).

NSGA-II methodology

The NSGA-II algorithm is the first and one of the commonly used evolutionary multi-objective optimization (EMO) algorithms which search for solution space to find Pareto-optimal solutions in a multi objective optimization problem. NSGA-II uses the elitist principle and an explicit diversity preserving mechanism. In addition, it emphasizes non-dominated solutions, and forms the Pareto front as Pareto-optimal solutions. The NSGA-II algorithm uses two effective strategies including an elite-preserving and an explicit diversity-preserving. NSGA-II uses an explicit diversity-preservation or niching strategy to assign a diversity rank to all the individuals that are in the same non-dominated front and thus have the same non-dominated rank in the population. The members within each non-dominated front that are in the least crowded region in that front are assigned a higher rank. For calculating the density of solutions surrounding a particular solution in the population, a crowding distance metric is used that is achieved from the average distance of the two solutions on either side of the solution along each of the objectives. Respecting that this particular niching strategy does not require any external parameters; therefore, it was chosen for NSGA II. Details can be found elsewhere. Because of the nature of the models of the multi-objective optimization problems, non-dominated sorting genetic algorithm (NSGA) can be used to find the non-dominant optimal solutions. In the absence of any additional information about multi-objective optimization problem, one of these Pareto-optimal solutions cannot be considered as better solution than the others. Superiority and Suitability of one solution over the others depends on several factors including user’s choice and problem environment. Therefore, the NSGA II determines a set of dominant solution and as a result Pareto front is obtained (Salimi et al., 2013).

Initial population for NSGA-II algorithm

Initial individuals are obtained by fixing the activities modes randomly from the remaining set of modes after the preprocessing procedure. Then, the activity list is constructed with respect to the resulting mode assignment. The ensuing schedule is precedence and nonrenewable resource feasible.

The initial generation is obtained by repeating the next steps POP times. Firstly, a mode assignment is generated by randomly selecting $m(j) \in M_j$ for activities j=1,…J. Secondly, the resulting mode assignment is checked for nonrenewable resource feasibility. In presence of violation, a local search procedure is applied trying to improve the current mode assignment. The process consists of randomly selecting a job j which has more than one mode alternative, and its current mode is changed by randomly selecting $m'(j) \in M_j \setminus \{m(j)\}$. The result is a new mode assignment m'. If there is improvement, $m(j)$ is replaced by $m'(j)$. This process is repeated until J consecutive unsuccessful trials to improve the mode assignment have been made, or in the best case, until the individuals become nonrenewable resource feasible. Thirdly, we adopt the mode assignment and construct a precedence feasible schedule by randomly sampling. The procedure consists of J stages; at each stage, a decision (say also eligible) set is determined; this set includes all unscheduled activities which every predecessor has scheduled yet. An activity is chosen randomly and scheduled at its earliest precedence and renewable resource feasible start time. These steps are repeated until the J steps are fulfilled. Finally, activities finish times are computed and the makespan of the project is derived (Elloumi & Fortemps, 2010).

Updated the population

Crossover operator

p_c Parameter is considered as Crossover probability and for selecting parent's chromosome in Crossover, we repeat the following process $(pop - size \times p_c)$ times. For i=1, 2… pop-size, we use three Crossover types as one point and two point unified Crossover. This process is described as follow. First, we must select a stochastic number in one point Crossover 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 Crossover 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 Crossover operation. Then, we reach the number of (pop-size) p_c parents for Crossover 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 Crossover between two parents like $x'_1 = (x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)})$, $x'_2 = (x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)})$, 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 is possible then we keep that and repeat Crossover operations to reach another possible child. If both of them are not possible, we repeat the operation to two possible childe.

Mutation operator

p_m Parameter is considered as probability of mutation. Parent chromosome is selected by the same method which was mentioned in intersection operations. Parent chromosome is selected which is almost as many as $(\text{pop} - \text{size}) \times p_m$. Then mutation operation is applied as the following method. In this research, gussing 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 a standard value is produced and multiplied by σ^2 and summed by X value and σ^2 is equal to $0.1 \times (V_{max} - V_{min})$. Which σ^2 is equal to mutation steps. Therefore, mu% (mutation ratio) is selected randomly and to have an integer value for mutants and at least one case may be found, value is rounded up.

The Multi-Objective Particle Swarm Optimization (MOPSO)

PSO simulates a social behavior such as bird flocking to a promising position or region for food or other objectives in an area or space. Like evolutionary algorithm, PSO conducts search using a population, which is called swarm, of individuals, which are called particles. Each particle represents a candidate position or solution to the problem at hand to represent a potential solution. During searching for optima each PSO particle adjusts its trajectory towards its own previous best position, and towards the best previous position attained by any member of its neighborhood (i.e., the whole swarm). Thus, global sharing of experience or information takes place and particles profit from the discoveries of themselves (i.e., local best), and previous experience of all other companions (i.e., global best) during search process. PSO is initialized with a population of M random particles and then searches for best position (solution or optimum) by updating generations until getting a relatively steady position or exceeding the limit of iteration number (i.e., T). In every iteration or generation, the local bests and global bests are determined through evaluating the performances (Azimi *et al.*, 2011; Bashiri *et al.*, 2011). A swarm consists of a set of particles and each particle represents a potential solution. $x^i[t]$ is the position of each particle that is defined by adding a velocity to a current position:

$$x^i[t+1] = x^i[t] + v^i[t+1]$$

That the velocity vector is defined as fallow:

$$v^i[t+1] = wv^i[t] + c_1 r_1 (x^{i.best}[t] - x^i[t] + c_2 r_2 (x^{g.best}[t] - x^i[t]$$

where $x^{i.best}[t]$ is position of the best particle member of the neighborhood of the given particle, $x^{g.best}[t]$ is the best position of the best particle member of the entire swarm (leader),w is inertia weight, c_1 is the cognitive learning factor and c_2 is the social learning factor (usually defined as constants), and $r_1, r_2 \in [0,1]$ are random values (Ritwik & Ginu, 2013; Bashiri *et al.*, 2011).

Main algorithm of MOPSO

In case of the relative simplicity of PSO, multi objective particle swarm optimization allows PSO algorithm to solve multi objective problems. This algorithm is based on Pareto dominance and it considers every non-dominated solution as new leader. This approach also uses a crowding factor to filter out the list of available leaders. This algorithm works thus. First, a swarm is initialized and a set of leaders is also initialized with the non-dominated particles from the swarm. This set is usually stored in an external archive. Then, some sort of quality measure is calculated for all the leaders in order to select one leader for each particle of the swarm. At each generation for each particle, a leader is selected and a flight is performed. Then, the particle is evaluated and its corresponding $x^{i.best}$ is updated. A new particle replaces its $x^{i.best}$ particle usually when this particle is dominated or if both are non-dominated with respect to each other. After all the particles have been updated, the set of leader is updated, too. Finally, the quality measure of the set of leaders is re-calculated and this procedure is repeated for a certain number of criterions. External repository: The main objective of the external repository is to keep a record of non-dominated vectors found during the search process. The external repository consists of two components; the archive controller and the grid, which are discussed in more details. The function of archive controller is to decide whether a certain solution should be added to archive or not. The mechanism of the grid is to produce well-distributed Pareto fronts (Aboutalebi *et al*., 2013).

Multi-Objective Simulated Annealing

Simulated annealing (SA) is one of the stochastic search algorithms which have been originally motivated by thermodynamic process of annealing in physics. Though SA was designed originally to use only one search agent, and therefore it does not need large memory to keep the population, there are some techniques for empowering SA to find the set of estimated Pareto-optimal solutions for multi-objective problems. Considering a minimization problem, SA allows “uphill” moves so as to avoid getting stuck at a local optimum by accepting worse solutions with a probability defined by "acceptance probability function." We find that the performance of the MOSA depends on the type of acceptance probability function applied and the rule of replacing current solution by candidate solution (Varadharajan & Rajendran, 2005). The SA algorithm starts with an initial solution for the given problem and repeats an iterative neighbor generation procedure that improves the objective function. During searching for the solution space and in order to escape from local minima, the SA algorithm offers the possibility to accept the worse neighbor solutions in a controlled manner. A neighboring solution (S') of the current solution (S) is generated in each iteration of the inner loop. If the objective function value of S' is better than S, then the generated solution replaces with the current one; otherwise, the solution can be also accepted with a probability $p = e^{-\frac{\Delta}{T}}$. where T is the value of current temperature (i.e., higher values of T give a higher acceptance probability) and $\Delta = f(S) - f(S')$. The acceptance probability is compared to a number $y \in [0, 1]$ generated randomly, and S' is accepted whenever $p > y$ (Chen *et al*., 2010).

A target-vector approach to solve a bi-objective optimization problem has been used. Ulungu et al (1999) have proposed a complete MOSA algorithm which they had tested on a multiobjective combinatorial optimization problem. A weighted aggregating function to evaluate the fitness of solutions has been used. The algorithm worked with only one current solution but maintained a population with the non-dominated solutions found during the search. The algorithm uses only one solution and the annealing process adjusts each temperature independently according to the performance of the solution in each criterion during the search. An archive set stores all the non-dominated solutions between each of the multiple objectives. A new acceptance probability formulation based on an annealing schedule with multiple temperatures (one for each objective) has also been proposed. The acceptance probability of a new solution depends on whether or not it is added to the set of potentially Pareto-optimal solutions. If it is added to this set, it is accepted to be the current solution with probability equal to one. Otherwise, a multiobjective acceptance rule is used. However, the acceptance decision of new solution has to take into consideration the improvement (or deterioration) of k objectives, simultaneously. In case of two optimization objectives (i.e. k = 2), the comparison of the actual and the new solution (i.e. decision vector) results depicted in three cases.

Case (a): The move from X_{act} to X_{new} is improving with respect to all k objectives. This means that

$\Delta f_{k'} = f_{k'}(X_{new}) - f_{k'}(X_{act}) \leq 0$ (Supposing a minimization MOP) for $k' \in \{1, \ldots k\}$.

Case (b): An improvement and deterioration can be simultaneously observed on different objectives. This means, there exist a k' and a k'' with $\Delta f_{k'} < 0$ and $\Delta f_{k''} > 0$. This is the case where the new solution is indifferent to the actual one. Thus, a strategy has to be defined to decide if the new solution should be accepted as current solution for the next iteration.

Case (c): All objectives are deteriorated, with $\Delta f_{k'} \geq 0$ for all k' and $\exists k'' \in \{1, \ldots k\}$ such that $\Delta f_{k''} > 0$. In this case, an acceptance probability to accept X_{new} has to be calculated.

Parameter settings for the NSGA-II, MOPSO and MOSA algorithms

Tables 2, 3, 4 present the control parameters for the NSGA-II, MOPSO and MOSA algorithms.

Table 2. Control parameters for the NSGA-II algorithms

Parameter	The parameters amount
Maximum Number of Iterations	200
Population Size	100
Crossover Percent	0.9
Number of Crossover	2*round(pCrossover*nPop/2)
Mutation Percent	1/ String length of chromosome
Number of Mutation	round(pMutation*nPop)
Mutation rate	0.01
Mutation step size	0.2*(VarMax-VarMin)
Parent selection method	Binary tournament
Stopping rule	Production of 200 generations

Table 3. Control parameters for the MOPSO algorithms

Parameter	The parameters amount
Maximum Number of Iterations	200
Population Size	100
Repository Size	100
Inertia Weight	0.5
Inertia Weight Damping Rate	0.99
Personal Learning Coefficient(c1)	1
Global Learning Coefficient(c2)	2
Number of Grids per Dimension	5
Inflation Rate	0.1
Leader Selection Pressure	2
Deletion Selection Pressure	2
Mutation Rate	0.1
Stopping rule	Production of 200 generations

Table 4. Control parameters for the MOSA algorithms

Parameter	The parameters amount
Maximum Number of Iterations	1000
Population Size	1
Number Variables	35
Number move	5
Initial Temperature	Pareto. Cost/10000

Computational Results

Proposed method presented in this research is coded by using multi objective genetic algorithm which proposed in MATLAB software. In this part, input parameters which consider general and control variables are presented and results of proposed algorithm solving are discussed and the proposed multi objective genetic algorithm is validated by GAMS. In Table 5, 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 5. 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

With the algorithm implementation 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.

Therefore to prove efficiency of the proposed algorithms, several sample problems in small scale including sub sets of Reference are listed in addition Bandar Abbas Gas condensate Refinery project with14 (Pan *et al*., 2008), 18 (Rifat & Önder, 2012), 20 (Luong & Ario, 2008), and 25 (Kwan *et al*., 2003) activity is solved by the proposed algorithms based on the four types of payments.

Indicators Performance

Because of the multi objective nature of the problem, the numerical results obtained by each algorithm were evaluated in terms of quality of the produced set of non-dominated solutions and of the associated approximation of the Pareto front. This quality of the Pareto front commonly includes not only the number of non-dominated solutions, but also convergence and distribution concepts. In our experiments, we evaluated the results (i.e., the pareto fronts) obtained by NSGA-II and MOPSO algorithms for each problem according to different quality metrics usually adopted in the literature of MOO. The adopted metrics were: Spacing, Maximum Spread, and Spread. Each metric considers a different aspect of the pareto front.

Spacing (S): this metric was proposed by Schott. S estimates the diversity of the achieved Pareto Front. S is derived by computing the relative distance between adjacent solutions of the Pareto Front as follows:

$$S = \frac{\sum_{i=1}^{N-1} (d_i - d_{mean})}{(N-1)d_{mean}} \qquad (21)$$

where n is the number of non-dominated solutions, di is the distance between adjacent solutions to the solution VI and is the average distance between the adjacent solutions. S=0 means that all solutions of the Pareto Front are equally spaced. Hence, values of S near zero are preferred (Kashif Gill *et al*., 2006).

Maximum Spread (MS): it was proposed by Zitzler *et al*. (2000) and evaluates the maximum extension covered by the non-dominated solutions in the Pareto Front. MS is computed by using as follows.

$$D = \sqrt{\sum_{j=1}^{m} (\max f_i^{\,j} - \min f_i^{\,j})^2} \qquad (22)$$

where n is number of solutions in the Pareto front, k is the number of objectives. This measure can be used to compare the techniques and thus define which of them covers a bigger extension of the search space. Hence, large values of this metric are preferred.

Criterion of Pareto solutions: Criterion of Pareto solutions numbers represents Pareto optimum solutions which can be found in every algorithms and the more number of Pareto front lead to a better situation (Zitzler *et al*., 2000).

Diversity: The scale measures disturbance among a set of non-dominated solutions. The related formula is as follows:

$$d = \sqrt{\sum_{i=1}^{m} \max(\|x_i^j - y_i^j\|} \qquad (23)$$

In place where, $\|x_i^j - y_i^j\|$ is direct distance between the x_i non-dominated solutions and y_i non-dominated solutions. The more the criterion, the less similar solutions and more diversity among the responses will be which covers a larger space; this shows the extent of the Pareto Front.

The algorithm execution time criterion: the algorithm execution time is considered as quality assessment criteria.

Comparative results: The performance of proposed NSGA-II algorithms and MOPSO are analyzed to solve condensate Abbas the algorithm with 35 activities and show examples of problems with number 14 (Pan *et al*., 2008), 18 (Rifat & Önder, 2012), 20 (Luong & Ario, 2008), and 25 (Kwan *et al*., 2003) activities based on the four types of payments. The algorithms results are shown in Tables 6, 7, 8, and 9.

Table 6. Comparative results based on the LSP

Number of activities \ Comparative Indicators		Spacing	Maximum Spread	Number of Pareto solution	Diversification Metric	Time
J=14	NSGA-II	0.5	259.9	35	361.67	258.3
	MOPSO	1	265.03	16	243.89	157.29
	MOSA	0.54	258.09	28	323.04	39.6
J=18	NSGA-II	1	7330.63	16	1055.24	282.55
	MOPSO	0.52	6873.13	14	993.45	151.51
	MOSA	0.37	7010.87	15	1026.42	55.62
J=20	NSGA-II	0.18	438.16	20	652.58	265.8
	MOPSO	0.36	436.13	19	635.95	160.49
	MOSA	0.28	436.08	20	652.74	50.94
J=25	NSGA-II	0.17	431.13	22	638.47	303.05
	MOPSO	0.24	366	19	593.89	204.91
	MOSA	0.26	429.01	22	636.54	87.43
J=35	NSGA-II	0.66	775236.35	73	6265.14	756.42
	MOPSO	0.78	752524.75	35	4936.62	997.06
	MOSA	0.9	677365.5	37	4568.86	286.92

Table 7. Comparative results based on the PEO

Number of activities \ Comparative Indicators		Spacing	Maximum Spread	Number of Pareto solution	Diversification Metric	Time
J=14	NSGA-II	0.51	54.82	13	492.01	252.48
	MOPSO	0.28	54.68	9	230.92	161.54
	MOSA	0.25	55.04	10	202.43	41.74
J=18	NSGA-II	0.74	808.15	9	745.42	291.15
	MOPSO	1.06	956.76	7	681.1	155.14
	MOSA	59	958.78	5	681.11	46
J=20	NSGA-II	0.34	146.94	15	579.18	276.39
	MOPSO	1.18	149.67	15	579.23	174.35
	MOSA	0.51	151.54	14	559.54	54.47
J=25	NSGA-II	0.55	95.14	16	557.09	327.13
	MOPSO	1.22	94.03	13	502.2	177.17
	MOSA	1.49	89.61	11	462.01	87.57
J=35	NSGA-II	0.79	365502.22	33	13116.45	776.43
	MOPSO	1.13	318008.8	12	7946.42	814.57
	MOSA	0.76	306008.4	15	9218.79	286.3

Table 8. Comparative results based on the ETI

Number of activities \ Comparative Indicators		Spacing	Maximum Spread	Number of Pareto solution	Diversification Metric	Time
J=14	NSGA-II	0.48	63.63	9	138.56	262.98
	MOPSO	0.71	249.49	8	182.12	137.92
	MOSA	0.41	99.3	6	133.69	41.78
J=18	NSGA-II	0.44	1209.15	6	607.69	296.56
	MOPSO	1.33	2471.8	4	545.39	153.37
	MOSA	0.99	6373.86	4	610.39	55
J=20	NSGA-II	0.47	154.49	10	468.39	280.54
	MOPSO	1.56	324.81	6	357.88	120.77
	MOSA	0.49	394.57	6	358.11	54.32
J=25	NSGA-II	0.31	111.29	8	384.28	329.09
	MOPSO	1.55	98.54	6	303.81	180.73
	MOSA	0.54	64.73	5	303.81	90.31
J=35	NSGA-II	0.9	516742.1	10	5565.03	707.15
	MOPSO	0.97	498873.4	8	5843.67	786.45
	MOSA	0.91	334016.3	9	6134.72	299.64

Table 9. Comparative results based on the PP

Number of activities \ Comparative Indicators		Spacing	Maximum Spread	Number of Pareto solution	Diversification Metric	Time
J=14	NSGA-II	1	1272.56	12	124.51	253.8
	MOPSO	1.61	1080.91	7	118.45	155.67
	MOSA	1.65	1080.7	8	134.78	42.14
J=18	NSGA-II	1.2	6919.69	10	780.44	288.01
	MOPSO	1.29	5981.53	6	680.54	161.06
	MOSA	1.27	6527.37	8	876.52	60.17
J=20	NSGA-II	0.25	3252.61	11	306.01	274.51
	MOPSO	0.3	2612.6	7	269.89	169.67
	MOSA	0.93	2106.09	7	291.64	54.44
J=25	NSGA-II	0.96	1333.59	10	251.27	320.54
	MOPSO	0.69	2522.06	8	146.44	211.9
	MOSA	0.58	2127.25	5	179.26	89.23
J=35	NSGA-II	0.89	1717730.93	13	4251.28	727.16
	MOPSO	0.58	907479.38	7	3318.16	521.72
	MOSA	0.57	159564.1	10	3619.34	281.79

The main problem that tables and the selected sample problems with 10, 18, 20 and 25 activity tables represent are the following topics:

NSGA-II algorithms have a greater ability to achieving higher number of answers in Pareto front, that from this perspective, contractor is faced with more number of options to choose administrative methods. NSGA-II algorithms have more extensive solutions in many cases. NSGA-II algorithm have more Pareto front extensive compared to other algorithms and show more diversity among the responses. NSGA-II algorithm has more uniformity in Pareto solutions than other algorithms. MOSA algorithm has less solving time compared to the NSGA-II and MOPSO algorithm in four payment and all studied problems.

Validation of the proposed algorithm

For proving efficiency of the Meta heuristic algorithm, solution of the algorithm is compared with solution of GAMS software. Therefore, 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) are solved by the proposed NSGA-II algorithm and GAMS software. Results and duration of executing NSGA-II algorithm and GAMS software are shown and analyzed which compared in Table 10 to 12. To compute means of different 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 10. Results from GAMS software and NSGA-II

Problem \ Type of payment		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 11. Percentage Average difference between Results from GAMS software and NSGA-II

Problem \ Type of payment	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	3.7	0.29	2.95	0.5	3	0.22	2.33	2.27

Table 12. 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 are presented in Table 10. This problem is solved by Meta heuristic simulation method, too. Result of differences for proposed method and precise method are presented in Table 11 in which their differences are very small and less than 3%. Also based on Table 12 and Figure 1, time to reach a solution is constant in the proposed method, but it increases as a quadratic function whereas above results shows the proposed method is a convergence to an optimum and solution algorithm.

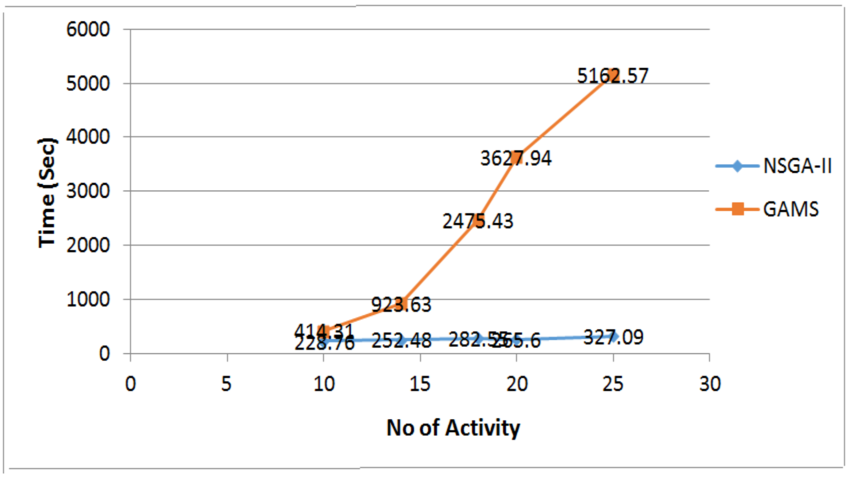

Fig. 1. Time of sample problems solving in the proposed NSGA-II algorithm and GAMS software

Conclusion

In this research, the scheduling section limited portion of the construction of a refinery by using a meta-heuristic approach investigated. The objectives of this model have been considered, minimizing project completion time and maximizing the net present value of the project. Also, the major Constraints and multi-objective model and Computational time complexity are classified as in the group NP-Hard problems. Therefore, in this paper, NSGA-II, MOPSO and MOSA algorithms are used in order to achieve optimal scheduling. Since every algorithm must be validated before use, the current study is applied for a real project which is progressive; we cannot also compare algorithm results with project results; therefore, to prove efficiency of the algorithm, the algorithm results are compared with results of solving the problem which is solved by GAMS software for some problems in small scales. Results represent that they are almost similar and smaller than 3%. Also, time to reach a solution in proposed method is constant, but in GAMS software increases as quadratic function. These results show that the proposed method is a convergence algorithm to optimal and efficient solution. The results of the NSGA-II, MOPSO and MOSA algorithm were investigated with comparative indices. The results of the NSGA-II, MOPSO and MOSA algorithm for the main problem and sample problems Indicates NSGA-II algorithm in the different criteria, have performed better than the other algorithms. For example, the NSGA-II algorithm in the number of Pareto solutions in problems all have been more of MOPSO, and MOSA algorithm provides more options for the decision makers. In the diversity, Maximum Spread (MS) and Spacing (S) index in the overwhelming of cases performed better than the other algorithms that Indicates are considered the extent and greater distribution of response space and uniformity between the solutions.

Suggestion that can be implemented in process of the project:

- Considering other objectives, such as robust, resources leveling, and project quality in the objective function

- Considering the time and activities Implementation cost and the amount of resources consumed as a fuzzy matter
- Considering non-renewable resource constraints
- Using the other algorithms for improvement and production solutions

References

Aboutalebi, R.S.; Najafi, A.A. & Ghorashi, B. (2012). "Solving multi-mode resource-constrained project scheduling problem using two multi objective evolutionary algorithms". *African Journal of Business Management*, 6, 4057-4065.

Afshar, A.; Ziaraty, A.K.; Kaveh, A. & Sharifi, F. (2009). "Non-dominated Archiving Multi colony Ant Algorithm in Time-Cost Trade-Off Optimization". *Journal of Construction Engineering and Management*, 135, 668–674.

Azimi, F.; Aboutalebi, R. S. & Najafi, A. A. (2011). "Using Multi-Objective Particle Swarm Optimization for Bi-objective Multi-Mode Resource Constrained Project Scheduling Problem". *World Academy of Science, Engineering and Technology*, 54, 285-289.

Aladini, K.; Afshar, A. & Kalhor, E. (2011). "Discount Cash Flow Time-Cost Trade off Problem Optimization; ACO Approach". *Asian Journal of Civil Engineering (Bulding and Housing)*, 12, 511-522.

Bashiri, M.; Kazemzadeh, R.; Atkinson, A.C. & Karimi, H. (2011). "Met-heuristic Based Multiple Response Process Optimization". *Journal of Industrial Engineering. University of Tehran, Special Issue*, 13-23.

Chen, W.N.; Zhang, J. & Liu, H. (2010). *A Monte-Carlo Ant Colony System for Scheduling Multi-mode Project with Uncertainties to Optimize Cash flows*. proceeding of IEEE Congress on Evolutionary Computation (CEC), 1-8.

Dayanand, N. & Padman, R. (2001). "Project contracts and payment schedules: the client's problem". *Management Science*, 47, 1654-67.

D¨orner, K.F.; Gutjahr, W.J.; Hartl, R.F.; Strauss, C. & Stummer, C. (2008). "Nature-inspired meta-heuristics for multi objective activity crashing". *Omega*, 36, 1019–1037.

Dayanand, N. & Padman, R. (1998). *Project contracts and payment schedules: The client's problem.* Working Paper, The Heinz School, Carnegie Mellon University, Pittsburgh, Pennsylvania.

El-Rayes, K. & Kandil, A. (2005). "Time-Cost-Quality Trade-Off Analysis for Highway Construction". *Journal of Construction Engineering and Management,* 131, 477-486.

Elloumi, S. & Fortemps, P. (2010). "A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem". *European Journal of Operational Research*, 205, 31–41.

Feng, C.; Liu, L. & Burns, S.A. (1997). "Using Genetic Algorithms to Solve

Construction Time-Cost Trade-Off Problems". *Journal of Computing in Civil Engineering*, 11, 184-189.

Herroelen, W.S.; De Reyck, B. & Demeulemeester, E.L. (1997). "Project network models with discounted cash flows: A guided tour through recent developments". *European Journal of Operational Research*, 100, 97-121.

Jongyul, K.; Changwook, K. & Inkeuk, H. (2012). "A practical approach to project scheduling: considering the potential quality loss cost in the time–cost tradeoff problem". *International Journal of Project Management*, 30, 264–272.

Kwan, W.K.; Mitsuo, G. & Genji,Y. (2003). "Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling". *Applied Soft Computing*, 2, 174–188.

Kashif Gill, M.; Kaheil, H.Y.; Khalil, A.; McKee, M. & Bastidas, L. (2006). "Multi objective particle swarm optimization for parameter estimation in hydrology". *Water Resources Research*, 42(Iss.7).

Kim, J.Y.; Kang, C.W. & InKeuk, H. (2012). "A practical approach to project scheduling: considering the potential quality loss cost in the time–cost tradeoff problem". *International Journal of Project Management*, 30, 264–272.

Khalilzadeh, M.; Kianfar, F. & Ranjbar, M. (2011). "A Scatter Search Algorithm for the RCPSP with Discounted Weighted Earliness-Tardiness Costs". *Life Science Journal*, 8, 634-641.

Liu, L.; Burns, S.A. & Feng, C. (1995). "Construction Time-Cost Trade-Off Analysis Using LP/IP Hybrid Method". *Journal of Construction Engineering and Management*, 121, 446-454.

Luong, D.L. & Ario, O. (2008). "Fuzzy critical chain method for project scheduling under resource constraints and uncertainty". *International Journal of Project Managemen*, 26, 688–698.

Moselhi, O. (1993). "Schedule compression using the direct stiffness method". *Canadian Journal of Civil. Engineering*, 20(1), 65-72.

Siemens, N. (1971). "A Simple CPM Time-Cost Tradeoff Algorithm". *Management Science*, 17, 354-363.

Moussourakis, J. & Haksever, C. (2004). "Flexible Model for Time/Cost Tradeoff Problem". *Journal of Construction Engineering and Management*, 130, 307-314.

Marek, M.; Grzegorz, W. O. & Wezglarz, 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, 639–668.

Möhring, R.H. & Stork, F. (2000). “Linear pre selective policies for stochastic project scheduling”. *Math Methods Operation Research*, 52, 501–15.

Najafi, A.A. & Niaki, S.T.A. (2006). “A genetic algorithm for resource investment problem with discounted cash flows”. *Applied Mathematics and Computation*, 183, 1057–1070.

Pan, H.; Robert, J. & Wilish, C.H. (2008). “Resource Constrained Project Scheduling with Fuzziness”. *Industrial engineering and management systems conference*.

Ritwik, A. & Paul, G. (2013). “A Heuristic Algorithm for Resource Constrained Project Scheduling Problem with Discounted Cash Flows”. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 3, 99-102.

Rifat, S. & Önder Halis, B. (2012). “A hybrid genetic algorithm for the discrete time–cost trade-off problem”. *Expert Systems with Applications*, 39, 11428–11434.

Shu-Shun, L. & Chang-Jung, W. (2008). “Resource-constrained construction project scheduling model for profit maximization considering cash flow”. *Automation in Construction*, 17, 966–974.

Smith-Daniels, D.E.; Padman, R. & Smith- Daniels, V.L. (1996). “Heuristic scheduling of capital constrained projects”. *Journal of Operations Management*, 14, 241–254.

Seifi, M. & Tavakkoli-Moghaddam, R. (2008). “A new bi-objective model for a multi-mode resource-constrained project scheduling problem with discounted cash flows and four payment models”. *IJE Transactions A: Basics*, 21, 347-360.

Salimi, R.; Bazrkar, N. & Nemati, M. (2013). “Task Scheduling for Computational Grids Using NSGA- II with Fuzzy Variance Based Crossover”. *Advances in Computing*, 3, 22-29.

Ulusoy, G. & Cebelli, S. (2000). “An equitable approach to the payment scheduling problem in project management”. *European Journal of Operational Research*, 127, 262–278.

Varadharajan, T.K. & Rajendran, C. (2005). “A multi-objective simulated annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs”. *European Journal of Operational Research*, 167, 772–795.

Xu, S. (2011). “Applying Ant Colony System to Solve Construction Time-Cost Trade off Problem”. *Advances Materials Research*, 179, 1390-1395.

Xiong, Y. & Kuang, Y. (2008). “Applying an Ant Colony Optimization

Algorithm-Based Multi-objective Approach for Time-Cost Trade-Off". *Journal of Construction Engineering and Management*, 134, 153-156.

Zhengwen, H. & Yu, X. (2008). "Multi-mode project payment scheduling problems with bonus–penalty structure". *European Journal of Operational Research*, 189, 1191–1207.

Zheng, D. X. M.; Ng, S.T. & Kumaraswamy, M. M. (2005). "Applying Pareto Ranking and Niche Formation to Genetic Algorithm-Based Multi objective Time--Cost Optimization". *Journal of Construction Engineering and Management*, 131, 81-91.

Zitzler, E.; Deb, K. & Thiele, L. (2000). "Comparison of multi objective evolutionary algorithms: Empirical results". *Evolutionary Computation Journal*, 8, 125–148.