International Journal of Computational Intelligence Systems, Vol. 7, No. 3 (June 2014), 456-471

Integrated ANN-HMH Approach for Nonlinear Time-Cost Tradeoff Problem

Bhupendra Kumar Pathak [1] *, Sanjay Srivastava [2]

[1] *Mathematics Department, Jaypee University of Information Technology,*
Waknaghat,
Solan, Himachal Pradesh 173234, India
E-mail: pathak.maths@gmail.com

[2] *Mechanical Engineering Department, Dayalbagh Educational Institute (Deemed University),*
Dayalbagh,
Agra, Uttar Pradesh 282005, India
E-mail: ssrivastava@dei.ac.in

Received 11 May 2012

Accepted 30 December 2013

Abstract

This paper presents an integrated Artificial Neural Network - Hybrid Meta Heuristic(ANN-HMH) method to solve the nonlinear time-cost tradeoff(TCT) problem of real life engineering projects. ANN models help to capture the existing nonlinear time-cost relationship in project activities. ANN models are then integrated with HMH technique to search for optimal TCT profile. HMH is a proven evolutionary multiobjective optimization technique for solving TCT problems. The study has implication in real time monitoring and control of project scheduling processes.

Keywords: Artificial Neural Network (ANN), Hybrid Meta-Heuristic (HMH), Time-Cost Tradeoff (TCT).

1. Introduction

Time-cost tradeoff problem is one of the most important aspects of engineering projects. Usually there is a nonlinear and non-increasing relationship between time and cost. The tradeoff between time and cost gives project planners both challenges and opportunities to work out the best plan that optimizes time and cost to complete a project. TCT problem is essentially a multiobjective optimization (MOO) problem [1]. Exact methods or mathematical models require lot of computational effort to solve TCT problem. For real-life complex networks, not only exact methods but also simple heuristic techniques fail to obtain optimal/near-optimal solutions efficiently. Multiobjective evolutionary algorithms (MOEAs) such as genetic algorithms, nondominated sorting genetic algorithm-II (NSGA-II) are suitable for searching a true Pareto front [2]. Genetic algorithm (GA) based search techniques were originally developed by Holland [3], which are derived from the mechanics of natural selection and later refined by Goldberg [4]. An overview of the selection mechanisms in GA has been given in [5]. Multiojective GA has been used to solve TCT problems [6-7]. The solution to discrete TCT problem in presence of constrained resources using multiojective GA is described in [8]. Nonlinear TCT prob-

*Corresponding Author

lems of project scheduling have been solved by neural network embedded multiobjective GA [9]. A computer-based Pareto multiobjective optimization approach has been utilized for solving time-cost-quality tradeoff problems [10]. NSGA-II has been used for time-cost-resource optimization in a construction project [11]. Project scheduling problems have also been solved using ant colony optimization approaches [12-14]. A branch-and-bound algorithm for discrete time-cost tradeoff problem has been proposed in [15]. An intelligent approach using ANN has been applied to solve resource-constrained nonlinear multiobjective time-cost tradeoff problem [16]. A survey of various approaches to solve TCT problem is detailed out in [17]. Wide varieties of TCT problems encountered in real world engineering projects are dealt in [18].

In this study, we present an integrated ANN-HMH method. The ANN models basically facilitate the evaluation of fitness function of HMH. In real world projects the Pareto-optimal front is unknown, so all such metrics [19] which measure the extent of convergence to a known set of Pareto-optimal solutions are not appropriate for the problem considered. HMH is used as a searching mechanism to search for the optimal time-cost tradeoff profile. It is important to note that the working (fitness function evaluation etc.) of HMH used is quite unconventional in comparison to other MOEAs [19-20]. The solutions obtained on the Pareto-front are diverse enough and include the relevant solution points which are required by the decision-maker in real life projects. In this paper, elitism is incorporated to keep the individuals in the tradeoff profile for the next generation, as it helps in converging to the true tradeoff profile. HMH suits well to the problem of searching for optimal TCT profile.

1.1. *A Mathematical Description of Time-Cost Tradeoff Problem*

The mathematical description of TCT problem is as follows:

The set ϕ represents the space of all feasible instances θ of the network where an instance $\theta = \{< t_i, c_i >: \ CT_i \leqslant t_i \leqslant NT_i, \ \ i = 1,2,...,n\}$ with t_i and c_i are time and cost of i^{th} activity respectively. n denotes the number of activities in the network. CT_i , NT_i are crash time and normal time of i^{th} activity respectively. For i^{th} activity, $c_i = f_i(t_i)$ where $f_i : [CT_i, NT_i] \to R$ is a nonlinear map. t_θ and c_θ denote the project duration and project cost respectively. Three possible problem formulations for the TCT problem are:

(a) Find θ^* such that $c^*_\theta = \min\limits_{\theta \in \phi}\{c_\theta \ \ : t_\theta \leqslant d\}$ where d is the given project deadline.

(b) Find θ^* such that $t^*_\theta = \min\limits_{\theta \in \phi}\{c_\theta \ \ : t_\theta \leqslant b\}$ where b is the given project budget.

(c) When the objective is to identify the entire time-cost tradeoff profile for the project network, then the problem is to find $B = \{\theta^* \in \phi \ : \nexists\, \theta \in \phi$ with $(t_\theta \leqslant t^*_\theta) \wedge (c_\theta \leqslant c^*_\theta)\}$ with strict inequality in at least one case.

Here the set of instances θ^* represents the entire time-cost tradeoff profile over the set of feasible project durations for the network. The decision-maker is free to choose a θ^* depending on specific project requirements. This formulation is the most generalized one, which has been addressed in this study.

2. Methodology

Preliminaries of HMH scheme for TCT problem taken-up in this study are as follows:

2.1. *Structure of a Solution*

The solution is a string (as shown in Fig.1) representing an instance θ of the project schedule.

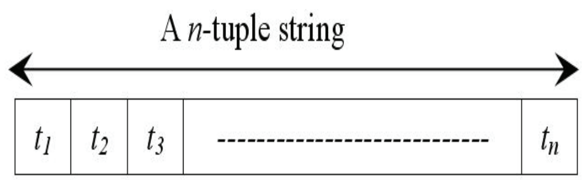

Fig. 1. An instance of project schedule.

Each element $t_i(time)$ of an $n-tuple$ string S: $[t_1, t_2, t_3, ..., t_n]$, can assume any value (a natural number) from $[CT_i,\ NT_i]$. Associated project duration

(t_θ) and project cost (c_θ) of each individual string is determined by computing the maximum path time and by summing up the corresponding cost for each activity respectively. The cost data of each activity is intelligently determined by the corresponding trained ANN.

2.2. *Initial Population*

The initial population consists of n_p solutions, where $(n_p - 2)$ strings are selected randomly from the feasible search space, i.e., each t_i of a string is chosen randomly from $[CT_i,\ NT_i]$. The remaining two strings are formed such that for the first string $t_i = NT_i,\ \forall i = 1,2,...,n$ and for the second string $t_i = CT_i,\ \forall i = 1,2,...,n$. This will ensure a good diversification of population in each generation of HMH while searching for optimal TCT profile. These solutions are referred to as 'parents'.

2.3. *Tradeoff Profile and Convex Hull*

Let θ_1 and θ_2 are two strings in a population $\mathbb{F}$, θ_1 dominates θ_2 if $t_{\theta_1} \leqslant t_{\theta_2}\, and c_{\theta_1} \leqslant c_{\theta_2}$ with either being $t_{\theta_1} < t_{\theta_2}\ or\ c_{\theta_1} < c_{\theta_2}$. Let $\mathbb{D}$ be a binary relation defined on the set $\mathbb{F}$ by $\mathbb{D} = \{(\theta_1, \theta_2)\ :\theta_1, \theta_2 \in \mathbb{F} \wedge \theta_1\ dominates\ \theta_2\}$, then the non-dominating set NDS is given by $NDS = \{\theta_i \in \mathbb{F}\ : (\theta_i, \theta_j) \notin \mathbb{D}\, \forall j,\ j \neq i\}$,i.e. it represents the strings (solutions) of $\mathbb{F}$ which are not dominated by any other string of $\mathbb{F}$. All the solutions of this set are joined with a curve as shown in Fig.2.

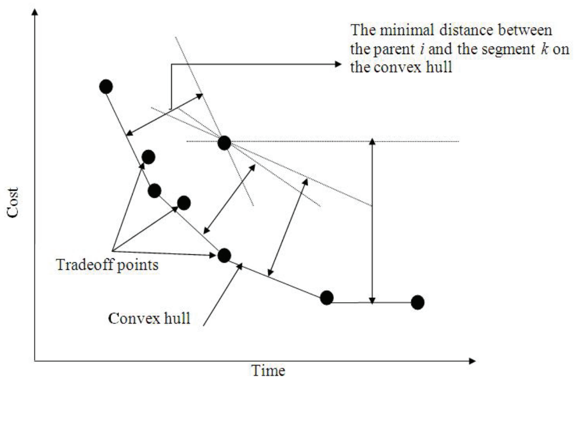

Fig. 2. Fitness evaluation of each member of the population.

The curve formed by joining these solutions is referred to as TCT profile and the solutions as the tradeoff points in the context of project management literature. We define a convex hull merely as a boundary (set of points) that encloses all members of a population with smallest convex set (Fig.2). This boundary is in the form of straight line segments. The purpose of drawing a convex hull for each population of HMH is to evaluate the fitness of each individual in the population [6]. A convex hull may not include all the solution points of the non-dominated set.

2.4. *Distance Measurement*

The distance d_i of an individual solution point in a population is determined by calculating the minimal Euclidean distance d_{ik} between the i^{th} solution point and each of the segment k of the convex hull, i.e., $d_i = \min\limits_{\forall k}(d_{ik})$ (Fig.2). The solutions with a lower value of distance are considered to be fitter than those having larger value of the distance.

2.5. *Crossover*

We consider one point crossover, in this, the i^{th} string, S_i produces a new string by performing crossover with another j^{th} string, S_j selected randomly. A random integer z with $1 \leqslant z \leqslant n$ is chosen, where z represents the crossover site. The first z positions of the new string are taken from the first z positions of S_i while the remaining $(n - z)$ positions are defined by the $(n - z)$ positions of S_j.

2.6. *Mutation*

The mutation operator modifies a randomly selected activity of a string with a probability p_m; that is $(p_m * \mid \mathbb{F} \mid)$ strings will undergo for mutation. The mutation operator works on a given string in the following manner:

The value of an element t_i, $1 \leqslant i \leqslant n$ in string S: $[t_1, t_2, t_3, ..., t_n]$ is randomly replaced by r, $r \in [CT_i,\ NT_i]$.

2.7. Simulated Annealing

Simulated annealing (SA) is a stochastic optimization method for searching the global optimum in the entire search space. Kirckpatrick et al. [21] introduced this method in the context of minimization problems. SA is motivated by an analogy to annealing in solids, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one [21]. By analogy with this physical process, SA chooses a random move to the neighbourhood of the original solution. If the move is better than its current position then simulated annealing will always take it. If the move is worse (i.e. lesser quality) then it will be accepted based on Boltzmann probability factor. The probability factor is regulated by a global parameter T (called the temperature), that is gradually decreased during the process and provides a mechanism for accepting a bad move. In the initial iterations this probability is high (almost one) and in the final stage of iterations it comes down to almost zero. In the context of this paper, the initial and final temperatures are denoted by $T^{(1)}$ and $T^{(f)}$ respectively.

2.8. Boltzmann Criterion

In SA, the selection of temperature is such that initially the probability of acceptance of a bad move is high (approximately 1) but as the temperature is slowly decreased, at the end, the probability of accepting a bad move is negligible (approximately 0). Such strategy enables the technique to seek the global optimum without getting stuck in any local optimum. The initial temperature, $T^{(1)}$ and final temperature, $T^{(f)}$ are calculated as follows:

Initially the probability of accepting a bad move is $e^{-\frac{\Delta d}{T^{(1)}}} = 0.99$ and finally it is $e^{-\frac{\Delta d}{T^{(f)}}} = 0.0001$, where Δd is the change of distance between the two neighborhood points in search space. This distance is calculated over the number of solutions. The initial temperature, $T^{(1)}$ is gradually decreased using the cooling ratio ($cool\_r$), and it comes down to almost zero in the final stage of iterations.

3. Modeling of Time-Cost Relationship with ANNs

Artificial Neural Networks (ANNs) have gained wide popularity in the intelligent decision making systems. The ANN approach is an inductive approach driven by data. The data driven approach of the ANNs enables them to behave as model free estimators, i.e., they can capture and model complex input-output relationships even without the help of a mathematical model.

In this work, time-cost relationships of each activity in project networks (Fig.3) is modeled by a function approximation capability of ANNs using Back Propagation Neural Network (BPNN)with Levenberg-Marquardt (LM) learning rule.

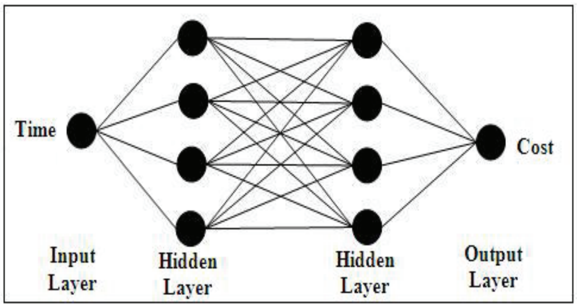

Fig. 3. Neural network architecture used for nonlinear TCT problem.

BPNN is a multiple layer network with an input layer, an output layer and some hidden layers between the input and output layers [22]. The LM learning rule is relatively faster [23-25] in modeling input/output relationships of complex processes. The LM approximation update rule is $\Delta W = [J^\top J + \mu I]^{-1} J^\top \varepsilon$, where ΔW is a 'weight update' matrix, J is a Jacobian matrix that contains first derivatives of the network errors with respect to the weights, $J^\top$ is the transpose of matrix J, I is the identity matrix, ε is a vector of network errors, and μ is a scalar. If the scalar μ is very large, the above expression becomes gradient descent method with a small step size. When the scalar μ is small the above expression is Gauss-Newton method. The Gauss-Newton method is faster and more accurate. Hence, the aim is to shift towards the Gauss-Newton method as quickly as possible. Thus the scalar μ is decreased after each successful step and increased only when a step increases the error.

4. Hybrid Meta Heuristic

Hybrid Meta Heuristic (HMH) begins with generating an initial population (see section 2.2) of n_p solutions (strings), referred to as 'parents', say; P_t, $t = 1,2,...,n_p$. Initially each parent P_t is allowed to produce $Q_t = \frac{n_c}{n_p}$ number of offsprings, where n_c is the total number of offsprings produced in a generation; this number is suitably chosen such that the search space can be extensively scanned for the selection process to follow. A parent P_t produces a offspring by performing a crossover with a randomly selected string from the remaining population of parents. A parent, P_t with its offsprings, Q_t constitute a family, say; R_t, all members of a family are referred to as 'solutions' of the family. Thus n_p families exist in a population. In the initial generation, each family, R_t has a single parent i.e. $P_t = 1$. Before performing any further process, mutation (see section 2.6) is applied on randomly selected strings of the offsprings population in each generation, in order to introduce random changes in subsequent generations. The tradeoff curve of the families is determined which represents the non-dominated set of solutions. Thereafter, the convex hull is drawn (see section 2.3). The basic idea is that if within a family, the distance of an individual from convex hull is smaller than other individuals, then this individual has better fitness with respect to either one or all of the objectives (Fig.2). For each family R_t, its members on the tradeoff curve are counted, say; $\mathscr{F}_t$. These $\mathscr{F}_t$ members become the parents for the t^{th} family for the next generation i.e. P'_t. However, if for a t^{th} family, no member appears on the tradeoff curve, then the t^{th} family is not rejected all together in the hope of its improvement in future. To decide the parent for the next generation from this family, a member of this family which is nearest to the tradeoff curve is selected. This proximity is measured by a fitness function (see section 2.4). The importance of the number $\mathscr{F}_t$ is twofold. Firstly it determines the parents for the next generation chosen from each family. This is how elitism is incorporated in the algorithm, which helps it in converging closer to true Pareto-optimal front (tradeoff curve). Elites of a current population are given an opportunity to be directly carried over to the next generation. Therefore, a 'good' solution found in a current generation will never be lost unless a better solution is discovered. The absence of elitism does not guarantee this feature [26]. Importantly the presence of elites enhances the probability of creating better offsprings [19]. It is observed that this heuristic helps in keeping a 'good' distribution of solutions over the tradeoff curve. The next step is to decide the number of offsprings, say; Q'_t, allocated to each family of the next generation. This number provides the information of how good each family is with respect to the diversification. To accomplish this, a distance measure has been defined (Fig. 2) which measures the 'nearness' of each member of a family to the tradeoff curve. To select the members so as to find Q'_t, the process of SA has been incorporated into the selection process using the procedure $find\_num$ (see section 4.2). The number Q'_t is proportional to the number of members of each family which are closer to the convex hull. Further, Q'_t also plays a direct role in measuring the fitness of each family R_t, that is, number of offsprings to be produced in the next generation by family R_t is determined by $\mathscr{F}_t$ plus the number of family members who qualify the Boltzman criterion (see section 2.8), say; $\mathscr{B}_t$. This is obvious as these $\mathscr{F}_t$ members are on the tradeoff curve. As mentioned earlier, initially each family has a single parent, but in subsequent generations the number of parents of each family may be more than one (as $\mathscr{F}_t \geqslant 1$ for the families whose members are on the tradeoff curve). In such a case, the number of offsprings, Q'_t is almost equally divided among P'_t parents for producing the offsprings. Now R_t family is updated by new family R'_t, which consists of P'_t parents plus Q'_t offsprings. The algorithm is able to search for the best family in the evolution process. The process is repeated until no improvement is observed in the tradeoff curve for a specified number of generations.

4.1. Pseudo-code of HMH

HMH is elucidated in the following steps:

1. set initial parameters i.e., $T^{(1)}$, $T^{(f)}$, n_c, n_p, and $Gen = 1$
2. $P_t = 1$ and $Q_t = \frac{n_c}{n_p}$, $t = 1, 2, ..., n_p$: Parent and offsprings for the first generation
3. $R_t = P_t \cup Q_t$: Families of the first generation
4. $\mathscr{F}_t$ = tradeoff points of R_t
   $\mathscr{B}_t$ = Remaining members of R_t, who qualify the Boltzmann criterion (See section 4.2)
5. Plot the convex hull of R_t
6. if $\mathscr{F}_t = \phi$, then $\mathscr{F}_t = \min\limits_{\forall k}(d_{ik})$: where d_{ik} is the distance of i^{th} member of family t, from the k^{th} line segment of the convex hull
7. $P'_t = \mathscr{F}_t$: Parents for the next generation
8. $Q'_t = \frac{n_c * (\mathscr{F}_t \cup \mathscr{B}_t)}{\sum_{t=1}^{n_p}(\mathscr{F}_t \cup \mathscr{B}_t)}$: Offsprings for the next generation
9. Until $\sum_{t=1}^{n_p}(|P'_t| + |Q'_t|) \neq |n_p| + |n_c|$ do
   If $\sum_{t=1}^{n_p}(|P'_t| + |Q'_t|) > |n_p| + |n_c|$ then
   $Q'_t = Q'_t - 1$
   $t = t + 1$
   else If $\sum_{t=1}^{n_p}(|P'_t| + |Q'_t|) < |n_p| + |n_c|$ then
   $Q'_t = Q'_t + 1$
   $t = t + 1$
   End
10. Apply Mutation on Q'_t
11. $R'_t = P'_t \cup Q'_t$: Families for the next generation
12. $R_t = R'_t$: Update the current generation families
13. $Gen = Gen + 1$: Increment in the generation
14. Repeat step 4 to 13 until the tradeoff curve remains identical or certain number of iterations has been reached

4.2. Procedure $find\_num$

To decide the number of the points who qualify the Boltzmann criterion, the procedure $find\_num$ is explained in the following steps:

1. Set
   $\mathscr{B}_t = 0$: Number of the points who qualify the Boltzmann criterion
   h = 1 : Iteration count
   $cool\_r = 0.85$: Cooling ratio is decided by performing the exhaustive experiments.
2. $R_t = P_t \cup Q_t$: Family of the generation
3. $\mathscr{R}_t = R_t - \mathscr{F}_t$: Select only those members of this family, which is not on the tradeoff curve
4. For $u = 1$ to $|\mathscr{R}_t|$
   $if(e^{\frac{d_{ik}}{T^{(h)}}} > \rho)$: ρ is a random number between 0 and 1
   $\mathscr{B}_t = \mathscr{B}_t + 1$: Increment in $\mathscr{B}_t$
   $T^{(h+1)} = cool\_r * T^{(h)}$:Update the temperature
   $h = h + 1$: Increment in the iteration

5. Working of Integrated ANN-HMH Approach

Integrated ANN-HMH approach is explained with the following steps:

1. Set the initial parameters: $T^{(1)}$, $T^{(f)}$, n_c, n_p, and mutation rate (p_m).
2. Set the number of activities (n) in the project network and define their precedence relationship and input normal time (NT), normal cost (CT), crash time (CT), and crash cost (CC) for each activity.
3. Generate initial n_p strings (see section 2.2) each of length n. Find the maximum path time (critical path time) and the corresponding cost by summing the cost of each activity along that path, which is provided by ANN. These n_p strings are referred to as 'parents'.
4. Each string generates $Q = \frac{n_c}{n_p}$ number of offsprings by performing crossover and mutation with other strings of parents.
5. Determine the time and cost of each offspring-string as mentioned in step 3.
6. Combine the parent-strings and their corresponding offsprings, which makes n_p families.
7. Plot the tradeoff points and convex hull of these n_p families. The tradeoff points of each family become the 'parents' of the family for the next

generation. If a family has no tradeoff points, the nearest point of the family is made the parent of the family for the next generation.

8. To decide the number of the offsprings for the next generation, *simulated annealing* has been incorporated (see section 4.2).
9. Generate the required number of offsprings for each parent by performing crossover and mutation with the other strings of parents.
10. Update the temperature.
11. Repeat steps 6 to 10 until the tradeoff curve remains identical or for a pre-specified number of iterations.

6. Result of the Benchmark Problems Using HMH Technique

In this study, three Benchmark problems involving convex Pareto front given in [19] are successfully attempted using HMH technique to demonstrate the capability of HMH technique for finding the solutions. To solve these test problems using HMH technique, genetic parameters – initial population (n_p), the ratio (n_c/n_p), and mutation rate (p_m) are selected as 20, 9 and 0.02 respectively and SA parameters – $T^{(1)}$, $T^{(f)}$, and $cool\_r$ are chosen as 100, 0.1 and 0.85 respectively. HMH technique is run for a maximum of 2000 iterations. It has also been shown that HMH is superior to multiobjective GA in terms of convergence to known analytic results, as well as from diversity view point [27]. In the following subsections, each of these problems is described and the performance of the HMH techniques on these problems is investigated.

6.1. *Schaffer's Two Objective Problem*

This problem has two objectives, which are to be minimized:

$$SCH : \begin{cases} f_1(x) & = x^2 \\ f_2(x) & = (x-2)^2 \\ & -A \leqslant x \leqslant A \end{cases}$$

This problem has Pareto-optimal solutions $x^* \in [0,2]$ and the Pareto-optimal set is a convex set: $f_2^* = (\sqrt{f_1^*}-2)^2$ in the range $0 \leqslant f_2^* \leqslant 4$. The values of the bound-parameter A are taken as $[-10,10]$ for this study.

The non-dominated solutions obtained from HMH technique lie on the Pareto-optimal front and the solutions are well distributed in solution space (Fig.4). This shows that HMH technique converges to true optimal front for this Benchmark problem.

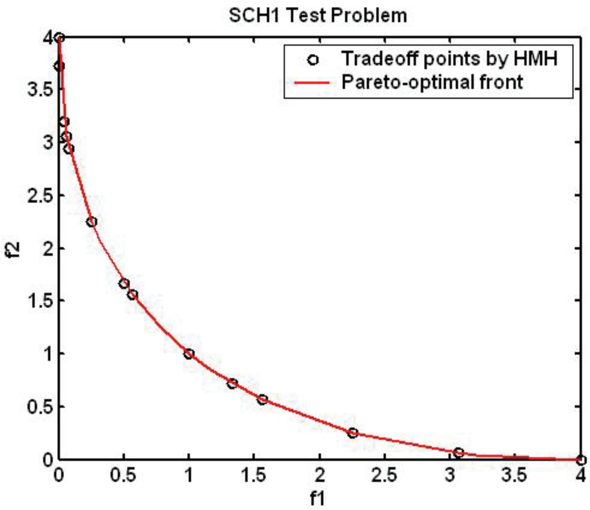

Fig. 4. Tradeoff points with HMH technique on SCH1

6.2. *Zitzler-Deb-Thiele's 1^{st} (ZDT1) Problem*

The ZDT1 benchmark problem has two objectives which are to be minimized is illustrated below:

$$ZDT1 : \begin{cases} f_1(x) & = x_1 \\ f_2(x) & = g(x)[1-\sqrt{\frac{x_1}{g(x)}}] \\ g(x) & = 1+\frac{9}{n-1}\sum_{i=2}^{n} x_i \end{cases}$$

This problem has 30 variables which lie in the range [0, 1] and convex Pareto-optimal solutions lie in the range $0 \leqslant x_1^* \leqslant 1$ and $x_i^* = 0$ for i = 2, 3,..., 30.

The non-dominated solutions obtained from HMH technique fairly matches with Pareto-optimal front and the solutions are well distributed in solution space as shown in Fig.5. Therefore, HMH technique demonstrates the ability in converging to true front and in finding the diverse solutions for large size problems.

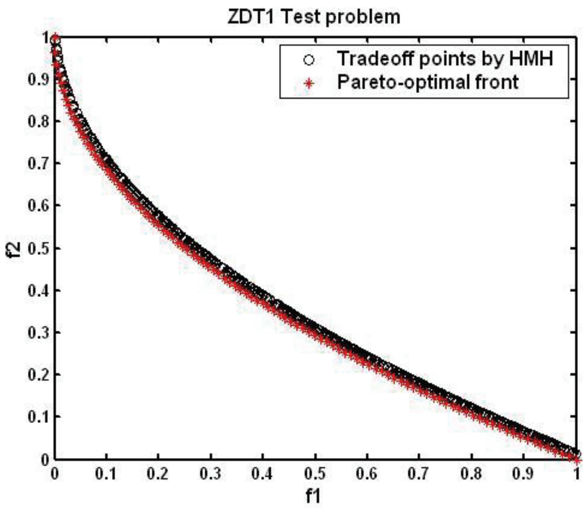

Fig. 5. Tradeoff points with HMH technique on ZDT1

6.3. *Zitzler-Deb-Thiele's* 3^{rd} *(ZDT3) Problem*

The ZDT3 benchmark problem includes a discreteness feature to the front. Its Pareto-optimal front consists of several noncontiguous convex parts. This test problem has two objectives which are to be minimized:

$$ZDT3: \begin{cases} f_1(x) & = x_1 \\ f_2(x) & = g(x)[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)} \sin(10\Pi x_1)] \\ g(x) & = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \end{cases}$$

This problem also has 30 variables which lie in the range [0, 1]. It has a discontinuous convex Pareto front in the range $0 \leqslant x_1^* \leqslant 1$ and $x_i^* = 0$ for i = 2, 3,..., 30.

The non-dominated solutions obtained from HMH technique matches with Pareto-optimal front as shown in Fig.6. This figure clearly demonstrates the ability of HMH technique in converging to true front and in finding the diverse solutions.

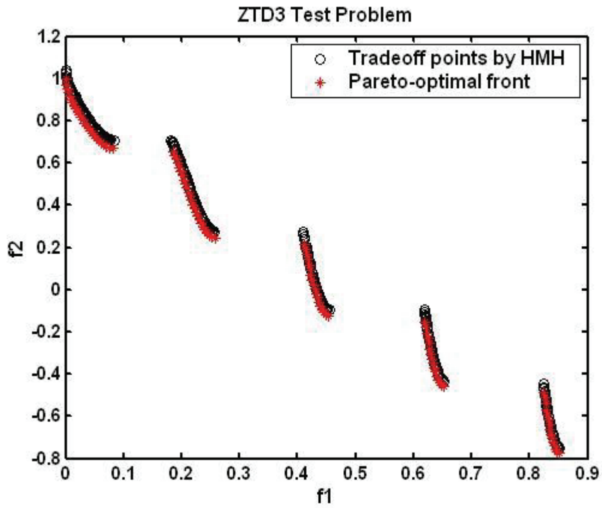

Fig. 6. Tradeoff points with HMH technique on ZDT3

7. ANN-HMH Approach for the Case Studies

To illustrate the concept and the effectiveness of the proposed ANN-HMH technique, three case studies are taken from literature with suitable modifications.The first, second and third case study are taken from [28], [6] and [29] and involve 7, 18 and 89 activities respectively. Fig.7, Fig.8 and Fig.9 show the precedence relationship of each activity of project network for each of the case studies respectively. The different options for the time (in days) and cost (in thousands) of each activity of these networks are shown in Table 1 , Table 2 and Table 3 respectively.

To solve these test problems using ANN-HMH, experiments are performed to select genetic and SA parameters. Genetic parameters – initial population (n_p), the ratio (n_c/n_p) and mutation rate (p_m) are selected as 60, 8 and 0.02 respectively. To decide the parameter n_p, experiments are done with different values of n_p, ranging from 20 to 100. For each value of n_p, 50 trials are conducted by keeping other parameters constant. The average time to converge to the final tradeoff profile is fastest for $n_p = 60$. Similar experiments are conducted to decide the SA parameters – $T^{(1)}$, $T^{(f)}$, and *cool_r*; which ensure the faster convergence to the final tradeoff curve.

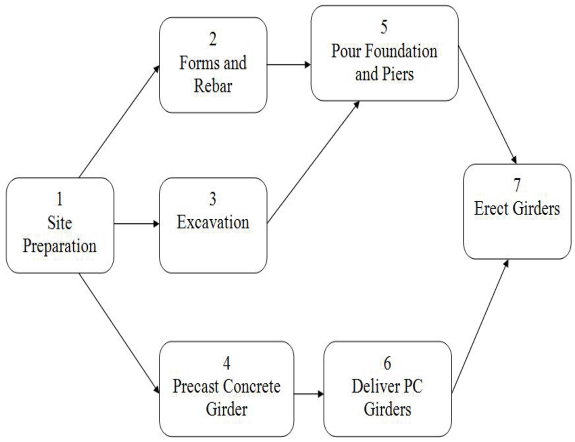

Fig. 7. Network of first case study

The SA parameters – $T^{(1)}$, $T^{(f)}$ and *cool_r* are chosen as 100, 0.1 and 0.85 respectively. In addition, the search is set to terminate when the tradeoff profile does not change in five consecutive iterations.

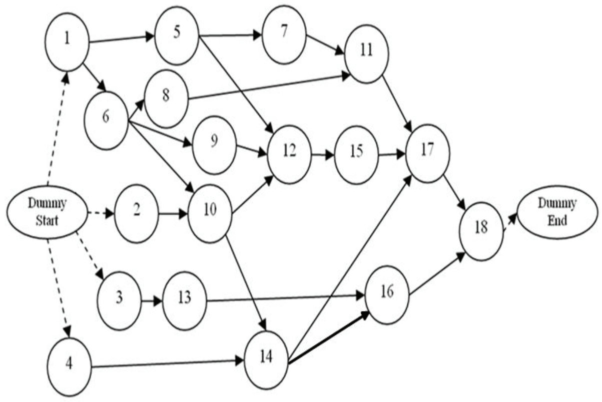

Fig. 8. Network of second case study

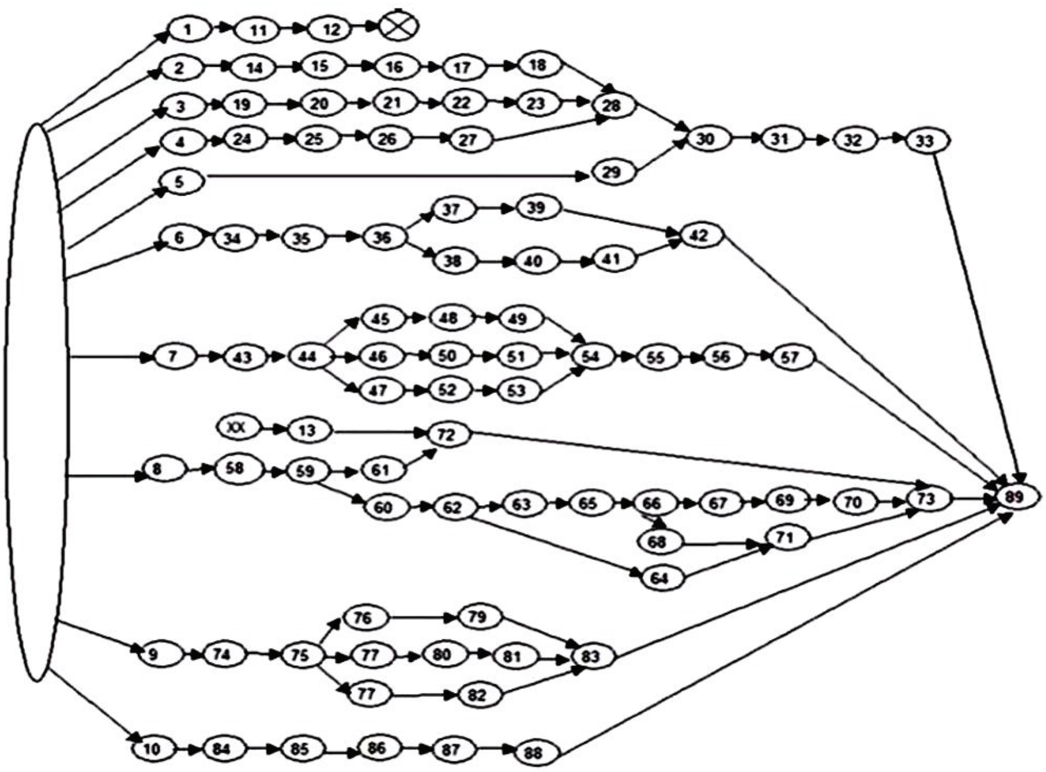

Fig. 9. Network of third case study

Table 1. Network data of first case study

Activity number	Options I (Time, Cost)	II (Time, Cost)	III (Time, Cost)	IV (Time, Cost)	V (Time, Cost)	VI (Time, Cost)	VII (Time, Cost)
1	(14, 23)	(15, 21.67)	(18, 19)	(20, 18)	(22, 16.1)	(23, 13)	(24, 12)
2	(15, 03)	(17, 2.64)	(18, 2.4)	(20, 1.8)	(23, 1.5)	(24, 1.35)	(25, 1)
3	(15, 4.5)	(17, 4.257)	(21, 4.16)	(22, 4)	(25, 3.692)	(30, 3.489)	(33, 3.2)
4	(12, 45)	(14, 40.127)	(15, 37.9)	(16, 35)	(17, 33)	(18, 31.995)	(20, 30)
5	(22, 20)	(24, 17.5)	(25, 16.8)	(26, 16.25)	(28, 15)	(29, 12.875)	(30, 10)
6	(14, 40)	(15, 38)	(17, 34.5)	(18, 32)	(20, 27.533)	(22, 22.307)	(24, 18)
7	(09, 30)	(11, 27.340)	(12, 26.99)	(14, 25.098)	(15, 24)	(17, 23.667)	(18, 22)

Table 2. Network data of second case study

Activity number	Options I (Time, Cost)	II (Time, Cost)	III (Time, Cost)	IV (Time, Cost)	V (Time, Cost)	VI (Time, Cost)	VII (Time, Cost)
1	(14, 2.4)	(15, 2.15)	(16, 1.9)	(18, 1.75)	(21, 1.5)	(23, 1.34)	(24, 1.2)
2	(15, 3.0)	(17, 2.63)	(18, 2.4)	(20, 1.8)	(21, 1.72)	(23, 1.5)	(25, 1)
3	(15, 4.5)	(17, 4.415)	(19, 4.22)	(22, 4)	(25, 3.73)	(30, 3.375)	(33, 3.2)
4	(12, 45)	(13, 44.3)	(15, 38.45)	(16, 35)	(18, 33.7)	(19, 32.4)	(20, 30)
5	(22, 20)	(24, 17.5)	(25, 16.4)	(26, 15.9)	(27, 15.7)	(28, 15)	(30, 10)
6	(14, 40)	(16, 39.2)	(17, 34.5)	(18, 32)	(20, 27.7)	(22, 20.3)	(24, 18)
7	(09, 30)	(11, 27.2)	(13, 26.1)	(14, 25.6)	(15, 24)	(17, 22.3)	(18, 22)
8	(14, 0.22)	(15, 0.215)	(16, 0.2)	(17, 0.19)	(21, 0.167)	(23, 0.15)	(24, 0.12)
9	(15, 0.3)	(18, 0.24)	(20, 0.18)	(23, 0.15)	(24, 0.13)	(25, 0.11)	(25, 0.1)
10	(15, 0.45)	(22, 0.4)	(23, 0.39)	(27, 0.345)	(28, 0.33)	(30, 0.325)	(33, 0.32)
11	(12, 0.45)	(13, 0.42)	(14, 0.37)	(16, 0.35)	(17, 0.33)	(19, 0.305)	(20, 0.3)
12	(22, 2)	(24, 1.75)	(25, 1.69)	(27, 1.525)	(28, 1.5)	(29, 1.2)	(30, 1)
13	(14, 4)	(15, 3.795)	(16, 3.5)	(18, 3.2)	(21, 2.75)	(23, 2.155)	(24, 1.8)
14	(09, 3)	(10, 2.93)	(12, 2.825)	(14, 2.605)	(15, 2.4)	(17, 2.295)	(18, 2.2)
15	(10, 6.525)	(13, 5.99)	(14, 4.5)	(16, 3.5)	(17, 3.355)	(18, 2.6)	(20, 1.93)
16	(20, 3)	(22, 2)	(24, 1.75)	(26, 1.685)	(28, 1.5)	(29, 1.385)	(30, 1)
17	(14, 4)	(16, 3.7)	(17, 3.455)	(18, 3.2)	(21, 2.78)	(23, 2.335)	(25, 1.8)
18	(09, 3)	(10, 2.9)	(12, 2.79)	(14, 2.565)	(15, 2.4)	(16, 2.315)	(18, 2.2)

Table 3. Network data of third case study

Activity number	Options I (Time, Cost)	II (Time, Cost)	III (Time, Cost)	IV (Time, Cost)	V (Time, Cost)	VI (Time, Cost)	VII (Time, Cost)
1	(91, 080)	(95, 076)	(98, 073)	(102, 68)	(105, 64)	(109, 58)	(112, 50)
2	(56, 145)	(61, 139)	(65, 135)	(70, 131)	(075, 126)	(079, 123)	(084, 116)
3	(21, 024)	(22, 23.5)	(23, 023)	(25, 22.25)	(026, 21.66)	(027, 20.5)	(028, 20)
4	(84, 018)	(88, 17.5)	(91, 16.863)	(95, 16.222)	(098, 15.598)	(102, 14.963)	(105, 14.4)
5	(28, 9.40)	(30, 9.10)	(33, 08.60)	(35, 8.10)	(37, 07.90)	(40, 7.70)	(42,7)
6	(21, 18.3)	(22, 17.85)	(23, 17.25)	(25, 16.9)	(26, 16.25)	(27, 15.6)	(28, 15)
7	(105, 6.25)	(107, 6.05)	(110, 5.89)	(112, 5.7)	(114, 5.4)	(117, 5.21)	(119, 5)

Table 3. (*Continued*)

Activity number	Options I (Time, Cost)	II (Time, Cost)	III (Time, Cost)	IV (Time, Cost)	V (Time, Cost)	VI (Time, Cost)	VII (Time, Cost)
8	(140, 50)	(142, 49.25)	(145, 48.495)	(147, 47.71)	(149, 46.975)	(152, 46.22)	(154, 45.455)
9	(42, 35)	(46, 33.1)	(49, 31.01)	(53, 29.187)	(56, 27.22)	(60, 25.271)	(63, 23.333)
10	(49, 15)	(51, 14.45)	(54, 13.89)	(56, 13.5)	(58, 12.65)	(61, 12.25)	(63, 11.667)
11	(91, 35)	(95, 33.5)	(98, 32)	(102, 29.9)	(105, 28.5)	(109, 26.5)	(112, 25)
12	(105, 983)	(107, 971.25)	(110, 960.75)	(112, 946.25)	(114, 935.5)	(117, 921.25)	(119, 910)
13	(14, 15)	(18, 13.87)	(21, 12.17)	(25, 10.37)	(28, 9.37)	(32, 7.47)	(35, 6)
14	(84, 42)	(89, 41)	(93, 40.5)	(98, 40.1)	(103, 39.5)	(107, 38.6)	(112, 38)
15	(126, 1477)	(130, 1442.1)	(133, 1405.2)	(137, 1362.5)	(140, 1345.5)	(144, 1302.2)	(147, 1266)
16	(7, 90)	(9, 84.2)	(12, 76.5)	(14, 69.87)	(16, 63.5)	(19, 57)	(21, 50)
17	(35, 69)	(37, 66.3)	(40, 64.2)	(42, 62)	(44, 59.4)	(47, 56.2)	(49, 54.429)
18	(84, 240)	(86, 236)	(89, 232.5)	(91, 227.5)	(93, 224.67)	(96, 220.1)	(98, 216.5)
19	(84, 12)	(88, 11.3)	(91, 10.5)	(95, 10.1)	(98, 9.38)	(102, 8.57)	(105, 8)
20	(273, 258)	(278, 254.5)	(282, 49.5)	(287, 246.3)	(292, 242.4)	(296, 238.8)	(301, 234)
21	(7, 15)	(8, 14.47)	(9, 14.07)	(11, 13.545)	(12, 13.146)	(13, 12.347)	(14, 12)
22	(14, 9)	(15, 8.657)	(16, 8.223)	(18, 7.997)	(19, 7.457)	(20, 7.456)	(21, 7)
23	(42, 40)	(44, 38.5)	(47, 37.75)	(49, 36.37)	(51, 35.433)	(54, 34.1)	(56, 33)
24	(70, 9)	(72, 8.654)	(75, 8.346)	(77, 8.123)	(79, 7.52)	(82, 7.321)	(84, 7)
25	(21, 12)	(22, 11.617)	(23, 11.34)	(25, 11.1)	(26, 10.6)	(27, 10.39)	(28, 10)
26	(196, 965)	(201, 958.5)	(205, 950.5)	(210, 946.4)	(215, 937.9)	(219, 931.4)	(224, 925)
27	(21, 103)	(22, 99.1)	(23, 94.3)	(25, 90.2)	(26, 85.5)	(27, 81.6)	(28, 77.25)
28	(28, 367)	(29, 358.7)	(30, 350.43)	(32, 342.23)	(33, 332.67)	(34, 325.57)	(35, 317)
29	(70, 3.8)	(75, 3.6)	(79, 3.5)	(84, 3.4)	(89, 32.8)	(93, 31.1)	(98, 3)
30	(28, 70)	(29, 68.43)	(30, 66.267)	(32, 65.1)	(33, 63.47)	(34, 62.07)	(35, 60)
31	(7, 6.5)	(8, 6.29)	(9, 5.9)	(11, 5.71)	(12, 5.491)	(13, 5.150)	(14, 5)
32	(14, 8)	(16, 7.236)	(19, 6.714)	(21, 6.204)	(23, 5.384)	(26, 4.554)	(28, 4)
33	(14, 2.4)	(15, 2.31)	(16, 2.222)	(18, 2.124)	(19, 1.999)	(20, 1.902)	(21, 1.8)
34	(84, 7)	(88, 6.813)	(91, 6.633)	(95, 6.295)	(98, 5.907)	(102, 5.881)	(105, 5.6)
35	(105, 3.6)	(107, 3.51)	(110, 3.41)	(112, 3.29)	(114, 3.19)	(117, 3.14)	(119, 3)
36	(133, 100)	(138, 96.9)	(142, 94.1)	(147, 91.405)	(152, 88.446)	(156, 85.498)	(161, 82.609)
37	(7, 14)	(8, 13.25)	(9, 12.55)	(11, 12.07)	(12, 11.43)	(13, 10.57)	(14, 10)
38	(133, 100)	(137, 97.54)	(140, 95.44)	(144, 92.9)	(147, 91)	(151, 88.4)	(154, 86.364)
39	(14, 25)	(15, 23.687)	(16, 22.522)	(18, 20.934)	(19, 19.455)	(20, 18.156)	(21, 16.667)
40	(7, 14)	(8, 13.45)	(9, 12.57)	(11, 12.001)	(12, 11.003)	(13, 10.546)	(14, 10)
41	(14, 10)	(18, 9.1)	(21, 7.9)	(25, 7.2)	(28, 5.8)	(32, 4.95)	(35, 4)
42	(42, 8)	(44, 7.597)	(47, 7.231)	(49, 7.111)	(51, 6.597)	(54, 6.313)	(56, 6)
43	(84, 2.5)	(89, 2.376)	(93, 2.295)	(98, 2.168)	(103, 2.053)	(107, 1.977)	(112, 1.875)
44	(28, 4)	(30, 3.753)	(33, 3.566)	(35, 3.304)	(37, 3.124)	(40, 2.869)	(42, 2.667)
45	(133, 15.2)	(137, 14.957)	(140, 14.491)	(144, 14.210)	(147, 13.878)	(151, 13.413)	(154, 13.127)
46	(140, 40)	(146, 38.623)	(152, 37.162)	(158, 35.570)	(163, 34.347)	(169, 32.915)	(175, 31.5)
47	(133, 95)	(137, 93.99)	(140, 93.119)	(144, 92.991)	(147, 91.087)	(151, 89.987)	(154, 89)
48	(7, 1.4)	(8, 1.346)	(9, 1.254)	(11, 1.209)	(12, 1.126)	(13, 1.074)	(14, 1)
49	(07, 1.15)	(08, 1.095)	(09, 1.029)	(11, 0.979)	(12, 0.907)	(13, 0.852)	(14, 0.8)
50	(07, 3.30)	(08, 3.084)	(09, 2.877)	(11, 2.657)	(12, 2.427)	(13, 2.209)	(14, 2.0)
51	(07, 2.70)	(08, 2.510)	(09, 2.290)	(11, 2.098)	(12, 1.905)	(13, 1.696)	(14, 1.5)
52	(07, 3.00)	(08, 2.780)	(09, 2.529)	(11, 2.267)	(12, 2.000)	(13, 1.789)	(14, 1.5)
53	(14, 0.72)	(15, 0.690)	(16, 0.642)	(18, 0.613)	(19, 0.569)	(20, 0.542)	(21, 0.5)
54	(70, 22.2)	(75, 21.60)	(79, 20.77)	(84, 20.115)	(89, 19.4)	(93, 18.727)	(98, 18)

Table 3. (*Continued*)

Activity number	Options I (Time, Cost)	II (Time, Cost)	III (Time, Cost)	IV (Time, Cost)	V (Time, Cost)	VI (Time, Cost)	VII (Time, Cost)
55	(14, 2.0)	(15, 1.891)	(16, 1.772)	(18, 1.674)	(19, 1.552)	(20, 1.434)	(21, 1.333)
56	(07, 4.0)	(08, 3.719)	(09, 3.514)	(11, 3.257)	(12, 2.997)	(13, 2.752)	(14, 2.5)
57	(21, 16.4)	(22, 15.687)	(23, 14.943)	(25, 14.195)	(26, 13.497)	(27, 12.737)	(28, 12)
58	(84, 35)	(86, 34.2)	(89, 33.1)	(91, 32.6)	(93, 30.66)	(96, 30.744)	(98, 30)
59	(77, 495)	(86, 459.267)	(96, 423.443)	(105, 387.526)	(114, 351.866)	(124, 315.633)	(133, 280)
60	(14, 20)	(16, 18.29)	(19, 16.5)	(21, 15.2)	(23, 13.44)	(26, 11.687)	(28, 10)
61	(07, 3.0)	(09, 2.5)	(12, 2.34)	(14, 2.1)	(16, 1.567)	(19, 1.343)	(21, 1.0)
62	(14, 12)	(15, 11.46)	(16, 10.56)	(18, 10.013)	(19, 9.256)	(20, 8.7)	(21, 8)
63	(63, 399)	(70, 379.268)	(77, 360.121)	(84, 338.415)	(91, 319.5)	(98, 299.8)	(105, 280)
64	(91, 79)	(98, 72.61)	(105, 66.12)	(112, 59.41)	(119, 52.98)	(126, 46.52)	(133, 40)
65	(14, 20)	(16, 18.2)	(19, 16.7)	(21, 15)	(23, 13.35)	(26, 11.87)	(28, 10)
66	(14, 12)	(15, 11.3)	(16, 10.8)	(18, 9.9)	(19, 9.43)	(20, 8.54)	(21, 8)
67	(63, 392)	(68, 372.13)	(72, 355.1)	(77, 337)	(82, 317.583)	(86, 298.54)	(91, 280)
68	(21, 4)	(22, 3.85)	(23, 3.65)	(25, 3.49)	(26, 3.4)	(27, 3.16)	(28, 3)
69	(14, 12)	(16, 11.01)	(19, 9.99)	(21, 8.8)	(23, 8.001)	(26, 6.99)	(28, 6)
70	(14, 20)	(18, 18.05)	(21, 16.05)	(25, 14.1)	(28, 11.98)	(32, 9.98)	(35, 8)
71	(70, 85)	(74, 76.5)	(77, 68.5)	(81, 59.5)	(84, 51.445)	(88, 43.222)	(91, 35)
72	(231, 690)	(235, 676)	(238, 660.5)	(242, 644.95)	(245, 628.5)	(249, 616.2)	(252,600)
73	(28, 63)	(30, 55.6)	(33, 48.8)	(35, 41.7)	(37, 34.666)	(40, 27.06)	(42, 20)
74	(119, 85)	(123, 82.615)	(126, 80.875)	(130, 77.625)	(133, 76.514)	(137, 74.575)	(140, 72.250)
75	(84, 35)	(85, 34.451)	(86, 34.163)	(88, 33.145)	(89, 33.245)	(90, 32.856)	(91, 32.308)
76	(133, 440)	(137, 434)	(140, 426.5)	(144, 419.9)	(147, 413.2)	(151, 406.8)	(154, 400)
77	(84, 840)	(91, 832)	(98, 826.5)	(105, 819.9)	(112, 813.5)	(119, 807)	(126, 800)
78	(126, 111.5)	(128, 109.68)	(131, 106.5)	(133, 105.75)	(135, 103.75)	(138, 101.8)	(140, 100)
79	(7, 21.2)	(8, 20.4)	(9, 20)	(11, 19.7)	(12, 19)	(13, 18.7)	(14, 18)
80	(14, 25)	(16, 22.8)	(19, 20.9)	(21, 18.6)	(23, 16.7)	(26, 14.6)	(28, 12.5)
81	(49, 47.8)	(50, 46.3)	(51, 45.234)	(53, 43.857)	(54, 42.666)	(55, 41.314)	(56, 40)
82	(7, 5.3)	(8, 5.157)	(9, 5.04)	(11, 4.9)	(12, 4.67)	(13, 4.61)	(14, 4.5)
83	(140, 105)	(144, 101.8)	(147, 98.2)	(151, 95.121)	(154, 91.432)	(158, 88.444)	(161, 85)
84	(28, 2.8)	(30, 2.76)	(33, 2.69)	(35, 2.639)	(37, 2.61)	(40, 2.54)	(42, 2.5)
85	(28, 2.8)	(29, 2.761)	(32, 2.689)	(36, 2.629)	(38, 2.611)	(40, 2.53)	(42, 2.5)
86	(140, 280)	(144, 278.2)	(147, 276.8)	(151, 275.1)	(154, 273.5)	(158, 272)	(161, 270)
87	(14, 7.5)	(16, 7.259)	(19, 7.01)	(21, 6.739)	(23, 6.489)	(26, 6.251)	(28, 6)
88	(21, 3)	(22, 2.865)	(23, 2.76)	(25, 2.645)	(26, 2.49)	(27, 2.372)	(28, 2.25)
89	(21, 10)	(23, 9.2)	(26, 8.555)	(28, 8.111)	(30, 7.42)	(33, 6.7)	(35, 6)

8. Computational Result and Discussion

Feed forward back propagation neural networks with LM rule for all activities are trained with time-cost data for the three case studies considered in this paper as represented in Table 1, Table 2 and Table 3 respectively. There are total seven time-cost options available for each activity for each of the case studies. Training data for ANN is prepared by picking up first and last time-cost options and by randomly selecting three more options. Remaining two options of each activity are used as testing data for the neural network. A three-layer neural network, as shown in Fig. 3, with one input-'activity time' and one output-'activity cost' is used. The training effort is very less with LM learning rule; it takes 4 to 6 iterations only (Fig.10). One network is trained for each activity, thus a total of 7, 18 and 89 ANNs are employed for the three case studies respectively. An error goal of 10^{-5} is specified. The modeling power of ANNs is validated using the testing data set. The activity cost is evaluated using ANNs, and is compared with known cost data and it is found that cost obtained using neural network is very close to known data. It clearly enumerates accuracy of the model.

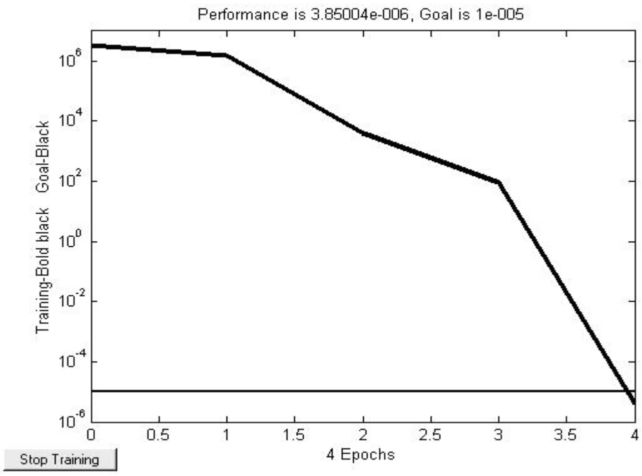

Fig. 10. ANN training of I activity of first case study

8.1. Computational Results of First Case Study

An initial generation of the first case study is shown in Fig.11.

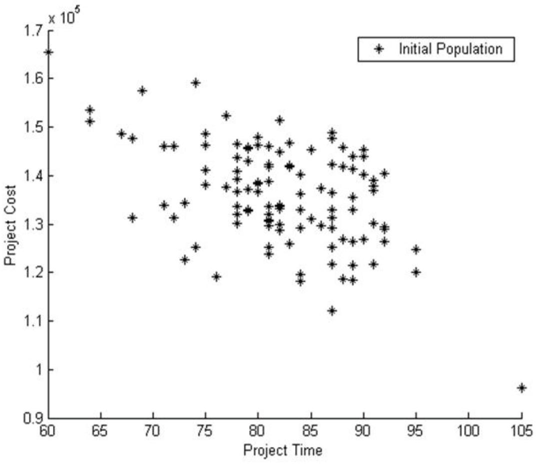

Fig. 11. The Initial generation : First case study

It can be seen that the initial generation is distributed over the solution space and does not gather in one region. Fig.12 depicts the best achieved tradeoff points and its convex hull. Since the tradeoff profile does not improve further, therefore this profile is concluded to be best TCT profile as searched by ANN-HMH.

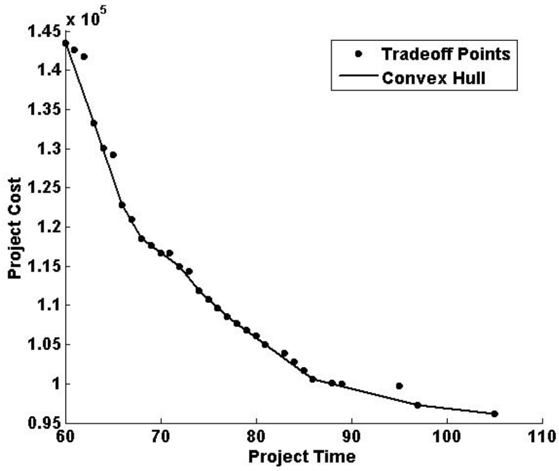

Fig. 12. Tradeoff Points and Convex hull of the final generation population : First case study

8.2. Computational Results of Second Case Study

The best achieved tradeoff points of the second case study is shown in Fig.13 and it is evident that the tradeoff points are distributed over the solution space and do not gather in one region.

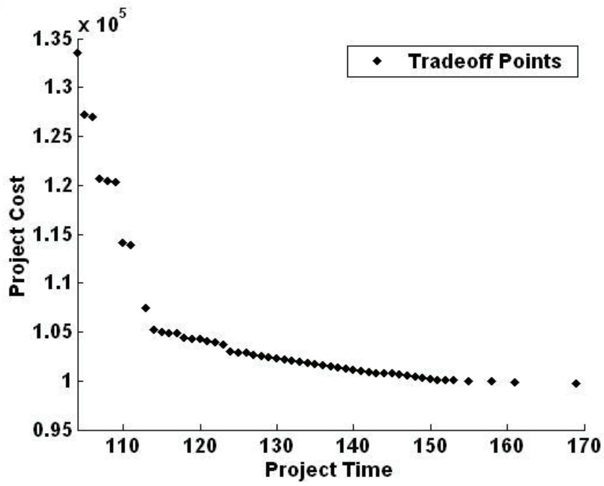

Fig. 13. Tradeoff Points of the final generation population : Second case study

Table 4. Statistical analysis for tradeoff points

Statistics	First case study		Second case study		Third case study	
	Time	Cost	Time	Cost	Time	Cost
Minimum	60	96200	104	99740	588	8459184
Maximum	105	143500	169	133520	805	9601800
Mean	76.2258	113930	131.2453	105110	709.9944	8641300
Median	75	110800	131	102195	715.5	8506400
Standard deviation	11.2538	13584	16.4140	7847.8	59.3998	263150
Range	45	47300	65	33780	217	1142600

8.3. Computational Results of Third Case Study

The best achieved tradeoff points for the third case study involving a large problem size with 89 activities are also well distributed over the solution space as shown in Fig.14. A very wide range of TCT points is obtained. The project managers is equipped with flexibility while carrying out project expediting as per the requirements.

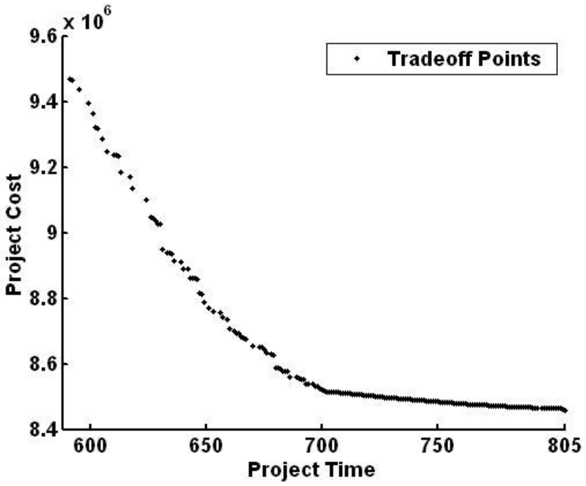

Fig. 14. Tradeoff Points of the final generation population : Third case study

8.4. Statistical Analysis of Case Studies

Graphical results for first, second and third case study are shown in Fig.12, Fig.13 and Fig.14 respectively and their statistical analysis is given in Table 4. The results are self explanatory.

9. Comparison of ANN-HMH approach with NNEMOGA

Techniques to solve the nonlinear TCT problems, where time-cost relationships of each activity in project networks is modeled by a function approximation capability of ANNs using Back Propagation Neural Network (BPNN) with Levenberg-Marquardt (LM) learning rule are scarce in the literature. The second case study has been solved using Neural Network Embedded Multiobjective Genetic Algorithm (NNEMOGA) [9]. As such we show the comparison of ANN-HMH with NNEMOGA for the second case study and it is found that ANN-HMH has better performance results in terms of convergence and diversity points of view (Fig.15).

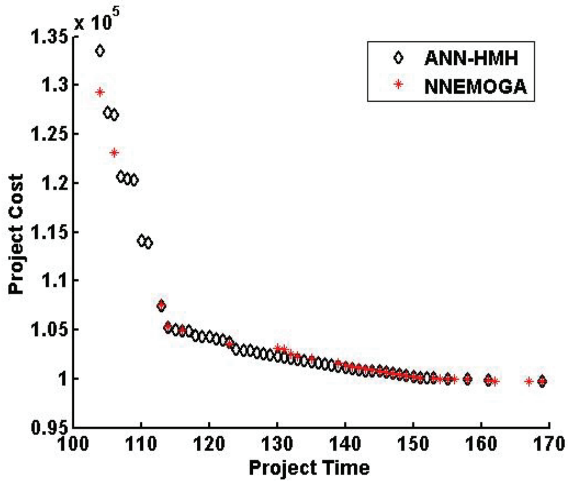

Fig. 15. Tradeoff Points of the final generation population by ANN-HMH and NNEMOGA : Second case study

For comparison purpose convergence criteria and genetic parameters are kept same for both ANN-HMH and NNEMOGA. Genetic Parameters– total initial population for the first generation (Parent + offsprings) and mutation rate (p_m) are selected as 540 (60+480) and 0.02 respectively for both approaches. It should be noted that in the case of ANN-HMH the total initial population, which is 540, comprises 60 parents and 480 offsprings where as in the case of NNEMOGA the total initial population (540) comprises parents only. ANN-HMH and NNEMOGA are run 10 times within same computational environments (IntelCore2Duo CPU, 1.8 GHz with 2 GB RAM) and found that ANN-HMH gives the best tradeoff points in 37 generations (average) while NNEMOGA gives the best tradeoff points in 81 generations (average). The average time to converge to the final tradeoff profile is 331.13 seconds for ANN-HMH and 481.36 seconds for NNEMOGA, which clearly illustrates the superiority of the proposed ANN-HMH technique.

10. Conclusions

The integrated ANN-HMH approach successfully demonstrates the realistic nature of TCT analysis and it does not place any restrictions on time-cost relationship of project activities. Three case studies have been studied to understand the feasibility and accuracy of the ANN-HMH approach. Since the nonlinear problems considered in this paper closely represent real world problems and the solutions are distributed providing more options in the solution space, the approach provides a comprehensive tool to project managers to analyze their time-cost optimization decisions in a more flexible and realistic manner and can help to choose the best alternative over the TCT profiles to execute the real-life projects.

Acknowledgments

The authors gratefully acknowledge the help provided by Dr. Kamal Srivastava (Dayalbagh Educational Institute, Dayalbagh, Agra) and Dr. Mohammed Usman (Jaypee University of Information Technology, Waknaghat, Solan).

References

1. M. Ehrgott and X. Gandibleux, "A survey and annotated bibliography of multiobjective combinatorial optimization,"*OR Spectrum*,**22**, 425–460 (2000).
2. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, **6**, 182–197 (2002).
3. J.H.Holland, "Adaptation in natural selection and artificial systems," *Univ of Mechigan: Press Ann Arbor Mich*, (1975).
4. D. E. Goldberg, "Genetic algorithms in search optimization and machine learning," *Reading MA: Addison-Wesley*,(1989).
5. R. Sivaraj and T. Ravichandran, "A review of selection methods in genetic algorithm," *International Journal of Engineering Science and Technology*, **3(5)**, 3792–3797 (2011).
6. C.-W. Feng, L. Liu, and S. A. Burns, "Using genetic algorithms to solve construction time-cost trade-off problems," *Journal of Computing in Civil Engineering*, **11**, 184–189 (1997).
7. B. K. Pathak and S. Srivastava, "MOGA based time-cost tradeoffs: responsiveness for project uncertainties," *Proc. IEEE Congress on Evolutionary Computation*, **CEC2007**, 3085–3092 (2007).
8. B. K. Pathak, H. K. Singh, and S. Srivastava, "Multi-resource-constrained discrete time-cost tradeoff with MOGA based hybrid method," *Proc. IEEE Congress on Evolutionary Computation*, **CEC2007**, 4425-4432 (2007).
9. B. K. Pathak, S. Srivastava, and K. Srivastava, "Neural network embedded multiobjective genetic algorithm to solve non-linear time-cost tradeoff problems of project scheduling," *Journal of Science and Industrial Research*, **67(2)**, 124–131 (2008).
10. D. Xundi, L. Heng, Z. Saixing, W. Vivian, and G. Hongling, "A pareto multi-objective optimization approach for solving time-cost-quality tradeoff problems," *Technological and Economic Development of Economy*, **17(1)**, 22-41 (2011).
11. B. Zahraie and M. Tavakolan, "Stochastic time-cost-resource utilization optimization using non-dominated sorting genetic algorithm and discrete fuzzy sets," *Journal of Construction Engineering and Management*, **135(11)**, 1162–1173 (2009).
12. S. T. Ng and Y. Zhang, "Optimizing construction time and cost using ant colony optimization approach," *Journal of Construction Engineering and Management*, **134(9)**, 721–728 (2008).
13. D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource- constrained project scheduling," *IEEE Journal of Transportations on Evolutionary Computation*, **6(4)**, 333–346 (2002).
14. S. Christodoulou, "Scheduling resource-constrained Projects with ant colony optimization artificial agents," *Journal of Computing in Civil Engineering*, **24(1)**, 45–55 (2010).
15. M. Vanhoucke, "New computational results for the discrete time/cost trade-off problem with time-switch constraints," *European Journal of Operational Research*, **165**, 359–374 (2005).
16. B. K. Pathak and S. Srivastava, "An Intelligent

approach for resource-constrained nonlinear multi-objective time-cost tradeoff," *DEI Journal of Science and Engineering Research*,**14(1&2)**, 153–163 (2007).

17. P. De, E. D. James, B. Ghosh, and E. Wells, "The discrete time-cost tradeoff problem revisited," *European Journal of Operational Research*, **81**, 225–238 (1995).
18. S. Srivastava, B. K. Pathak, and K. Srivastava, "Project scheduling: time-cost tradeoff problems," *Computational Intelligence in Optimization, Springer-Verlag Berlin Heidelberg (Y Tenne and C-K Goh ed)*, **7**, 325–357 (2010).
19. K. Deb, "Multi-objective optimization using evolutionary algorithms," *NewYork: JohnWiley and Sons*, (2001).
20. C. M. Fonseca and P. J. Fleming,"Genetic algorithm for multiobjective optimization: formulation discussion and generalization," *Proc. of fourth International Conference on Genetic Algorithms*, 416–423 (1993).
21. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, **220**, 671–680 (1983).
22. L. Fausett, "Fundamentals of neural networks," *NJ: Prentice Hall Englewood Cliffs*, (1994).
23. K. H. Raj, R. S. Sharma, S. Srivastava, and C. Patvardhan, "Modelling of manufacturing process with ANNs for intelligent manufacturing," *International Journal of Machine Tools and Manufacture*, **40**, 851–868 (1999).
24. S. Srivastava, K. Srivastava, R. S. Sharma, K. H. Raj, and S. N. Dwivedi, "Multi-objective process optimization of hot closed die forging for intelligent manufacturing," *International Journal of Agile Manufacturing*, **1**, 61–69 (2004).
25. S. Srivastava, K. Srivastava, R. S. Sharma, and K. H. Raj, "Modelling of hot closed die forging of an automotive piston with ANN for intelligent manufacturing," *Journal of Science and Industrial Research*, **63**, 997–1005 (2004).
26. M. V. Arias and C.A.C. Coello, "Asymptotic convergence of metaheurisitcs for multiobjective optimization problems," *Soft Computing*, **10**, 1001-1005 (2005).
27. K. Srivastava, S. Srivastava, B. K. Pathak, and K. Deb, "Discrete time-cost tradeoff with a novel hybrid meta heuristics," *Lecture Notes in Economics and Mathematical System, Springer Publication*, **634**, 177–188 (2010).
28. L. Liu, S. Burns, and C. Feng, "Construction time-cost trade-off analysis using LP/IP hybrid method," *Journal of construction engineering and management*, **121(4)**, 446–454 (1995).
29. B. K. Pathak, "Project scheduling using metaheuristics," *Ph. D. Thesis, Dayalbagh Educational Institute, Dayalbagh, Agra*, (2007).

Co-published by Atlantis Press and Taylor & Francis