EFFECTS OF PROJECT UNCERTAINTIES ON NONLINEAR TIME-COST TRADEOFF PROFILE

B. KUMAR PATHAK AND S. SRIVASTAVA

Abstract. This study presents the effects of project uncertainties on nonlinear time-cost tradeoff (TCT) profile of real life engineering projects by the fusion of fuzzy logic and artificial neural network (ANN) models with hybrid meta-heuristic (HMH) technique, abridged as Fuzzy-ANN-HMH. Nonlinear time-cost relationship of project activities is dealt with ANN models. ANN models are then integrated with HMH technique to search for Pareto-optimal nonlinear TCT profile. HMH technique incorporates simulated annealing in the selection process of multiobjective genetic algorithm. Moreover, in real life engineering projects, uncertainties like management experience, labor skills, and weather conditions are commonly involved, which affect the duration and cost of the project activities. Fuzzy-ANN-HMH analyses responsiveness of nonlinear TCT profile with respect to these uncertainties. A comparison of Fuzzy-ANN-HMH is made with another method in literature to solve nonlinear TCT problem and the superiority of Fuzzy-ANN-HMH is demonstrated by results. The study gives project planners to carry out the best plan that optimizes time and cost to complete a project under uncertain environment.

1. Introduction

Time-cost tradeoff (TCT) analysis is the probably the most important aspect of engineering projects. The objective of the TCT analysis is to reduce the project duration to meet a specific deadline with the least cost. In general, there is a tradeoff between the time and cost to complete an activity. In this study we are considering nonlinear TCT problems wherein time-cost relationship of each project activity in a project network is nonlinear. ANN models are capable to capture nonlinear time-cost relationship of each activity. In real life projects, uncertain variables like management experience (ME), labor skills (LS), weather conditions (WC) are commonly present, and during project implementation phase these dynamically affect the project schedule.

1.1. Literature Review.

While scheduling the projects, uncertainties can be handled by the fuzzy set theory. Many researchers have focused on modeling the effects of uncertainties on the time of project activities while ignoring the effects on the cost of activities. An integrated fuzzy simulation and genetic algorithm method for describing the uncertainty of activity duration times is proposed by Ke et al.

Received: February 2014; Revised: April 2015; Accepted: June 2015

Key words and phrases: Fuzzy logic, Artificial Neural Network, Hybrid Meta-Heuristic, Time-Cost Tradeoff.

[7]. Leu et al. [9] have used fuzzy set theory to model the variations in the duration of activities due to changing environmental factors. The effects of project uncertainties on linear TCT profiles using integrated fuzzy genetic algorithm (GA) approach have been dealt in [14]. Eshtehardian et al. [4] have used fuzzy numbers to address the uncertainties in the activities execution times and costs. TCT analysis of a project network in fuzzy environments is dealt in [2], where the membership function of the fuzzy minimum total crash cost is constructed. A hybrid technique combining simulated annealing algorithms and fuzzy logic is proposed to model the managers' behavior in predicting time and cost of TCT problem [18]. Pathak et al. [15] have solved nonlinear TCT problems of project scheduling by neural network embedded multiobjective genetic algorithm (MOGA). Srivastava et al. [16] deal with a variety of TCT problems encountered in engineering projects using paradigms of computational intelligence.

A fusion of fuzzy logic and ANN models with HMH technique is presented in this work to analyze the responsiveness of nonlinear TCT profiles with respect to ME, LS and WC. ANN models facilitate the evaluation of fitness function of HMH technique. HMH technique incorporates simulated annealing in the selection process of MOGA for deciding offspring for the next generation population. It is important to note that the working of HMH (fitness function evaluation etc.) technique used is quite unconventional in comparison to other multiobjective evolutionary algorithms (MOEA) [3]. A fuzzy logic framework is used for modeling the uncertainties of time and cost of project activities. Elitism in HMH is incorporated by keeping the individuals on the tradeoff profile for the next generation, which helps in converging to the true tradeoff profile. It may be noted that fusions like Fuzzy-ANN-HMH are rarely present in the literature to solve nonlinear TCT problems, however, Fuzzy-ANN-HMH is compared with another method, called integrated fuzzy-ANN-GA (IFAG), in [16] and it is demonstrated that Fuzzy-ANN-HMH has better performance.

1.2. **Nonlinear Time-Cost Tradeoff Problem: A Mathematical Description and Problem Formulation.** Reducing completion time of a project activity, in general, increases its cost. The minimum cost to complete an activity is called the normal cost, and the corresponding duration is known as the normal time. Whereas the minimum possible duration to complete an activity is the crash time, and the corresponding cost is the crash cost. A nonlinear relationship of a project activity is shown in Figure 1. Mathematical description of a nonlinear TCT problem is as follows.

Let the set Ω represents the space of all feasible instances θ of the network where an instance $\theta = \{< t_i, c_i > : \; CT_i \leq t_i \leq NT_i, \;\; i = 1, 2, ..., n\}$ 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 $c_{\theta^*} = \min_{\theta \in \phi}\{c_\theta \quad : t_\theta \leq d\}$ where d is the given project deadline.
(b) Find θ^* such that $t_{\theta^*} = \min_{\theta \in \phi}\{t_\theta \quad : c_\theta \leq b\}$ 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 $(t_\theta \leq t_{\theta^*}) \wedge (c_\theta \leq c_{\theta^*})\}$ 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.

There is a set of options for carrying out each activity, and project managers can choose an option for each activity to minimize project cost and project duration simultaneously.

Suppose that a project network consists of n activities. Each activity i can be performed with θ_i options with a corresponding cost c_i and time duration t_i. Let x_i be the option chosen for i^{th} activity for the TCT problem, therefore the vector of decision variables is taken as $X = x_1, x_2, ..., x_n$. Let ES_i be the earliest start time of the i^{th} activity calculated according to the precedence relationships for the project network.

The multiobjective TCT problem is defined as:

$$Minimize\ Cost\ C(X) = \sum_{i=1}^{n} c_i(x_i) \tag{1}$$

$$Minimize\ Time\ T = max\{ES_i + t_i(x_i) : i = 1, 2, ..., n \tag{2}$$

Subject to

$$1 \leq x_i \leq \theta_i \tag{3}$$

The project duration T is determined by computing the maximum path time (critical path method).

1.3. **Modeling of Time-Cost Relationship with ANNs.** In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network. In more practical terms neural networks are nonlinear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs. In this work, time-cost relationship of each project activity is modeled using back-propagation neural network (BPNN) with Levenberg-Marquardt (LM) learning rule (Figure 2). BPNN is a multiple layer network with an input layer, an output layer and some hidden layers between the input and output layers [5]. LM learning rule is relatively faster in modeling input-output relationships of complex processes [17]. 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,

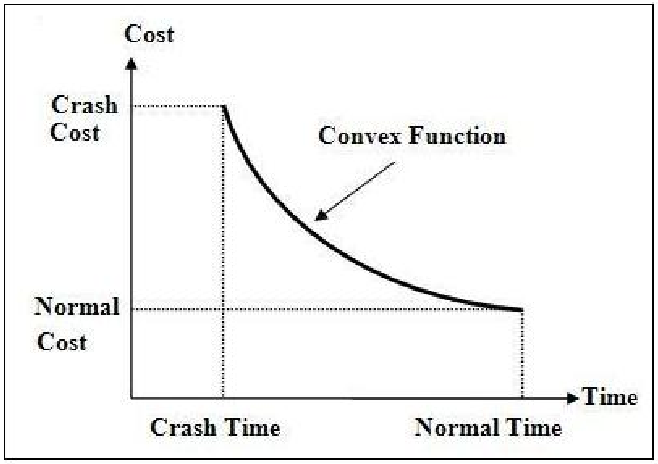

FIGURE 1. Nonlinear Time-cost Tradeoff Relationship for an Activity

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 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.

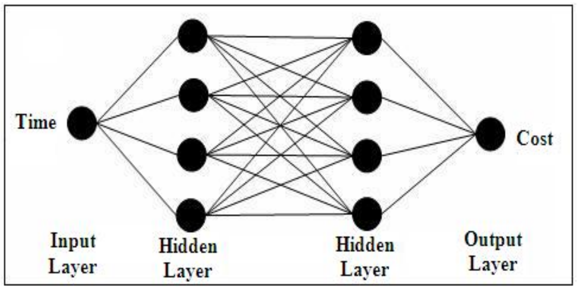

FIGURE 2. Neural Network Architecture for Nonlinear TCT Problem

1.4. **Fuzzy Inference System.** Fuzzy set theory was first proposed by Zadeh in 1965 [19] and after that used by Mamdani [11] in control. This is based on fuzzy logic and can be used in a wide range of domains in which information is incomplete or imprecise. Fuzzy set theory is a very powerful tool to represent imprecise concepts that uses linguistic variables rather than quantitative variables. Linguistic variable is a variable whose values are words or sentences in a given language. Fuzzy inference system (FIS) is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made. The process of fuzzy inference involves membership functions, fuzzy logic operators, and 'if-then' rules. In this study, a Mamdani-type FIS [12] is employed using MATLAB's fuzzy logic toolbox. Mamdani's effort was based on Lotfi Zadeh's paper on fuzzy algorithms for complex systems and decision processes [20].

1.5. **Simulated Annealing.** Simulated annealing (SA) is a stochastic optimization method for searching the global optimum in the entire search space [8]. 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. SA chooses a random move to the neighbourhood of the original solution. If the move is better than its current position then SA will always take it. If the move is worse (i.e. of 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). 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, 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. $T^{(1)}$ is gradually decreased using the cooling ratio ($cool\_r$), and it comes down to almost zero in the final stage of iterations.

This paper is divided to four sections. Section 1 is Introduction with five subsections. Section 1.1 describes past studies in solving different versions of TCT problems. Mathematical description and problem formulation are presented in section 1.2. ANN, FIS and SA are introduced in sections 1.3, 1.4 and 1.5 respectively. Section 2 describes Methodology with three subsections: 2.1, 2.2 and 2.3. Section 2.1 details out HMH technique: section 2.1.1 to section 2.1.7 covers fundamentals of HMH, section 2.1.8 describes working of HMH, and section 2.1.9 and section 2.1.10 illustrate pseudo code of HMH technique and Boltzmann criterion respectively. Section 2.2 explicates FIS: input and output variables are defined in section 2.2.1, membership function curves for input and output variables are identified in section 2.2.2 and 2.2.3 respectively, section 2.2.4 describes fuzzy rule sets employed to deal with uncertain environment, defuzzyfication is illustrated in section 2.2.5, and section 2.2.6 furnishes pseudo-code for designing FIS. Section 2.3 explains step-wise procedure of Fuzzy-ANN-HMH approach. Section 3 elucidates results and discussion on two case studies: section 3.1 focuses on ANN training, section 3.2 illustrates effects of uncertainties, and section 3.3 brings out the comparison of Fuzzy-ANN-HMH with IFAG. Conclusions are mentioned in the last section, i.e. section 4.

2. **Methodology**

The methodology is detailed in the following subsections.

2.1. **Hybrid Meta-Heuristic Technique.**

HMH technique [16], [13] along with its fundamentals is described below for its ready reference.

2.1.1. *Structure of a Solution.* The solution is a string (Figure 3) representing an instance θ of the 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 cost for each activity respectively. The cost data of each activity is intelligently determined by the corresponding trained ANN.

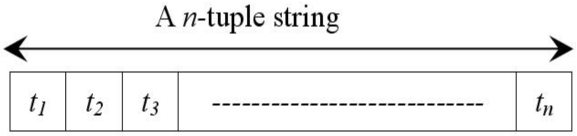

Figure 3. An Instance of Project Schedule

2.1.2. *Initial Population.* Initial population of HMH 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 technique while searching for Pareto-optimal TCT profile. These solutions are referred to as ‘parents’.

2.1.3. *Tradeoff Profile.* Let θ_1 and θ_2 be two strings in a population $\mathbb{F}$. θ_1 dominates θ_2 if $t_{\theta_1} \leq t_{\theta_2}$ and $c_{\theta_1} \leq c_{\theta_2}$ 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 $\mathbb{F}$. These non-dominating solutions are referred to as tradeoff points (Figure 4) and curve joining these points is called TCT profile in the context of project management literature. For more clarity Algorithm 1 is defined for tradeoff profile.

2.1.4. *Convex Hull.* We define a convex hull merely as a boundary (set of points) that encloses all members of a population with smallest convex set. This boundary is in the form of straight line segments (Figure 4). The purpose of drawing a convex hull for each population of HMH technique 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 solutions as illustrated in Figure 4 and Algorithm 2.

Algorithm 1: Tradeoff Profile (Tradeoff Points)

```
Let population of size $n_p$ contain 'time' and 'cost' in an array '$pop$' of an order of
$n_p \times 2$;
$tradeoffpoints = \phi,\ sortpop = \phi,\ temp\_pop = \phi,\ index = 1$ : Initialization of
variables;
for $i = 1$ to $n_p$ do
    for $j = 1$ to $n_p$ do
        if $(i \mathrel{!}= j)$ & $(pop(j,1) <= pop(i,1))$ & $(pop(j,2) < pop(i,2))$ then
            $temp\_pop(i,1) = nan$; Assigning $i^{th}$ position of $temp\_pop(i,1)$ as '$nan'$
            (not a number) i.e. blank
            $temp\_pop(i,2) = nan$; Assigning $i^{th}$ position of $temp\_pop(i,2)$ as '$nan'$
            (not a number) i.e. blank
            break
        else
            $temp\_pop(i,1) = pop(i,1)$; Assigning non-dominated 'time' to
            $temp\_pop(i,1)$ array
            $temp\_pop(i,2) = pop(i,2)$; Assigning non-dominated 'cost' to
            $temp\_pop(i,2)$ array
        end
    end
end
Following steps will store the non-dominating set of points to $tradeoffpoints$ array
from $temp\_pop$ array. for $i = 1$ to $n_p$ do
    if $(temp\_pop(i,1) >= 0$ or $temp\_pop(i,2) < 0)$ then
        $index = index + 1$
        $tradeoffpoints(index,1) = temp\_pop(i,1)$
        $tradeoffpoints(index,2) = temp\_pop(i,2)$
    end
end
```

2.1.5. *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., $d_i = \min_{\forall k}(d_{ik})$ (Figure 4). The solutions with a lower value of distance are fitter than those having larger value of the distance.

2.1.6. *Crossover.* One point crossover is used in this study. In one point crossover, 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 $1 \leq z \leq n$ 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.

2.1.7. *Mutation.* The mutation operator modifies a randomly selected activity of a string with a probability p_m. In this study, $(p_m \times \mid \mathbb{F} \mid)$ strings will undergo for mutation. The mutation operator works on a given string in the following manner.

Algorithm 2: Convex hull

```
Let population of size $n_p$ contain 'time' and 'cost' in an array '$pop$' of an order of
$n_p \times 2$;
$i = 2,\ k = 1,\ numb = 2,\ convexhull = \phi,\ sortpop = \phi$; Initialization of variables
$sortpop = sortrows(pop)$; Sort the points of $pop$ by x-coordinate(time), in case of a
tie, sort by y-coordinate(cost)and store in $sortpop$
$convexhull(1, :) = sortpop(1, :)$; Taking first pair of point i.e. 'time' and 'cost' from
sorted population
while $(i < n_p)$ do
    $flag = 1$; Initialization of $flag$ variable
    while $(flag\ \&\ (i < n_p))$ do
        while $(sortpop(k, 1) == sortpop(i, 1))$ do
            $i = i + 1$; Increment in variable $i$
        end
        $slope1 = \frac{sortpop(k,2)-sortpop(i,2)}{sortpop(k,1)-sortpop(i,1)}$; Finding the slope
        $i = i + 1$; Increment in variable $i$
        while $(sortpop(k, 1) == sortpop(i, 1))$ do
            $i = i + 1$; Increment in variable $i$
        end
        $slope2 = \frac{sortpop(k,2)-sortpop(i,2)}{sortpop(k,1)-sortpop(i,1)}$; Finding the slope
        if $(slope2 > slope1)$ then
            $flag = 0$
        end
    end
    if $(!\ flag)$ then
        $convexhull(numb, :) = sortpop(i - 1, :)$
        $numb = numb + 1$; Increment in variable $numb$
        $k = i - 1$; Updating the value of $K$
    end
end
$convexhull(numb, :) = sortpop(n_p, :)$ : Adding last point of $sortpop$ to $convexhull$
for completing the all points of convex hull
```

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

2.1.8. *Working of HMH.* HMH technique incorporates SA in the selection process of MOGA, thus providing the advantages of both MOGA and SA. This begins with generating an initial population (refer section 2.1.2) of n_p solutions (strings), referred to as 'parents', say; $P_t,\ t = 1, 2, ..., n_p$. Initially each parent P_t produces $|Q_t| = \frac{n_c}{n_p}$ number of offsprings by performing a crossover with a randomly selected string from the remaining population of parents, where n_c is the total number of offsprings produced in each generation is fixed; this number is suitably chosen such that the search space can be extensively scanned for the selection process to follow. Now mutation (refer section 2.1.7) is applied on randomly selected strings of the offsprings population in each generation, in order to introduce random changes in

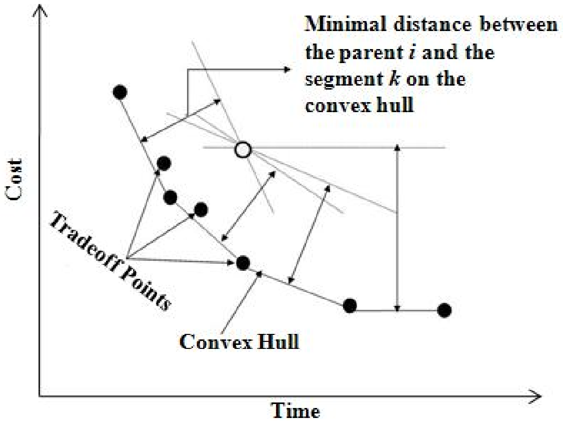
Minimal distance between the parent *i* and the segment *k* on the convex hull
Cost
Tradeoff Points
Convex Hull
Time

FIGURE 4. Fitness Evaluation of each Member of the Population

subsequent generations. A parent, P_t with its offsprings, Q_t constitute a family, say; R_t. Thus n_p families exist in the initial generation. Each family, R_t, has a single parent i.e. $|P_t| = 1$ in the initial generation. The tradeoff curve of the families is determined which represents the non-dominated set of solutions (refer section 2.1.3). Thereafter, the convex hull is drawn (refer section 2.1.4). The basic idea for drawing the convex hull is that if 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 (Figure 4). For each family R_t, its members on the tradeoff curve are counted, say; $|\mathcal{F}_t|$. These $|\mathcal{F}_t|$ members become the parents for the t^{th} family for the next generation, say; 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 (refer section 2.1.5). The importance of the number $|\mathcal{F}_t|$ 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 [1]. Importantly the presence of elites enhances the probability of creating better offspring [16]. 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 (Figure 4) which measures the 'nearness' of each member of a family to the tradeoff curve. To select the members of offspring (Q'_t), the process of SA has been incorporated into the selection process using the procedure *Boltzmann*() (refer section 2.1.10). The number of offspring $|Q'_t|$ 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 number of tradeoff points ($|\mathcal{F}_t|$) plus the number of family members who qualify the Boltzman criterion (refer section 1.5), say; $|\mathcal{B}_t|$. This is obvious as these $|\mathcal{F}_t|$ 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 $|\mathcal{F}_t| \geq 1$ 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.

2.1.9. *Pseudo-code of HMH technique.* HMH technique is elucidated in the following steps:

```
 1. set initial parameters i.e., $T^{(1)}$, $T^{(f)}$, $n_c$, $n_p$, and $Gen = 1$
 2. Each parent $P_t$ produces $|Q_t| = \frac{n_c}{n_p}$, $t = 1, 2, ..., n_p$, number of offsprings
    by performing crossover
 3. Apply Mutation on $Q_t$ offsprings (refer section 2.1.7)
 4. $R_t = P_t \cup Q_t$   : Families of the first generation
 5. Plot the tradeoff points of $R_t$ (refer section 2.1.3)
 6. Plot the convex hull of $R_t$ (refer section 2.1.4)
    Following steps decide the parents, offsprings and families for the next generation.

 7. Select the tradeoff points of $R_t$ ; say $\mathcal{F}_t$
 8. If $\mathcal{F}_t = \phi$, then select a member of this family which is nearest to the
    tradeoff curve and say it is $\mathcal{F}_t$ for this family. This proximity is measured
    by a fitness function (refer section 2.1.5)
 9. $P'_t = \mathcal{F}_t$   : Parents for the next generation
10. $\mathcal{M}_t = R_t \cap \mathcal{F}_t$ : Members of $R_t$ excluding $\mathcal{F}_t$
    $\mathcal{B}_t$ = Member of $\mathcal{M}_t$, who qualify the Boltzmann criterion (refer section
    2.1.10)
11. $|Q'_t| = \frac{n_c * (|\mathcal{P}'_t| + |\mathcal{B}_t|)}{\sum_{t=1}^{n_p} (|\mathcal{P}'_t| + |\mathcal{B}_t|)}$   : This formula distributes the total number of
    offsprings $n_c$ into each family in proportion to the number of parent plus
    those number of members who qualify the Boltzmann criterion for this
    generation.
12. 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
    $|Q'_t| = |Q'_t| - 1$
    $t = t + 1$
    if $t = n_p$ then set $t = 1$
    else If $\sum_{t=1}^{n_p}(|P'_t| + |Q'_t|) < n_p + n_c$ then
    $|Q'_t| = |Q'_t| + 1$
    $t = t + 1$
    if $t = n_p$ then set $t = 1$
    End
    Here step 12 keeps the number of solutions same for each generation.
13. Each parent $P'_t$ produces $|Q'_t|$ number of offsprings by performing crossover
14. Apply Mutation on $Q'_t$ offsprings
15. $R'_t = P'_t \cup Q'_t$    : Families for the next generation
16. $R_t = R'_t$    : Update the current generation families
17. $Gen = Gen + 1$    : Increment in the generation
18. Repeat step 5 to 17 until the tradeoff curve remains identical or certain
    number of iterations has been reached
```

2.1.10. *Procedure Boltzmann().* To decide the number of the points who qualify the Boltzmann criterion, the procedure *Boltzmann*() is explained in the following steps:

```
1. Set
   $\mathcal{B}_t = \phi$ : Members 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. $\mathcal{M}_t = R_t \cap \mathcal{F}_t$    : Select only those members of this family, which is not
   on the tradeoff curve
4. For $u = 1$ to $|\mathcal{M}_t|$
   $if(e^{\frac{d_{ik}}{T^{(h)}}} > \rho)$    : $\rho$ is a random number between 0 and 1
   $\mathcal{B}_t = \mathcal{B}_t \cup \mathcal{M}_t$    : Increment in $\mathcal{B}_t$
   $T^{(h+1)} = cool\_r \times T^{(h)}$    :Update the temperature
   $h = h + 1$    : Increment in the iteration
```

2.2. **Building a Fuzzy Inference System for Project Uncertainties.** This section introduces the graphical user interface (GUI) editors and viewers in Fuzzy Logic Toolbox of MATLAB to define membership functions and fuzzy rules set and analyzes the behavior of Fuzzy Inference System (FIS).

2.2.1. *FIS for linguistic input and output variables.* FIS Editor (Figure 5) is employed for capturing the effects of linguistic variables such as ME, LS and WC on each activity duration and cost of the project network. FIS has been designed with ME, LS and WC as input linguistic variables and 'activity duration' and 'activity cost' as output linguistic variables for each activity of the project network.

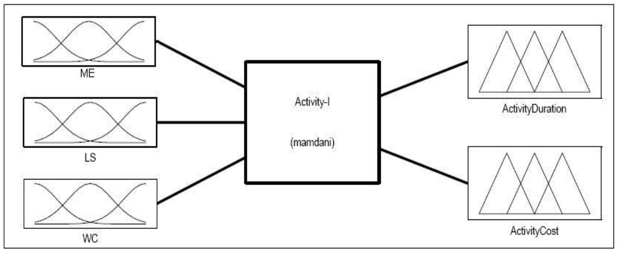

FIGURE 5. Fuzzy Inference System for Linguistic Input and Output Variables

2.2.2. *Membership Function Curves of Input Variables.* Each input linguistic variable namely ME, LS and WC are defined in the range [0, 1] and are modeled using five triangular membership functions such as shown in Figure 6, Figure 7 and Figure 8 respectively, for each activity of the project network.

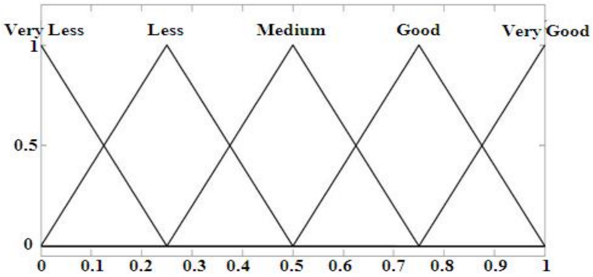

FIGURE 6. Membership Function for Management Experience

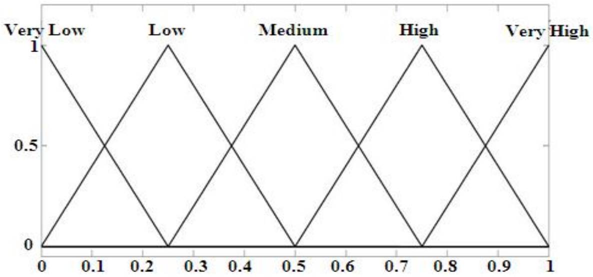

FIGURE 7. Membership Function for Labour Skills

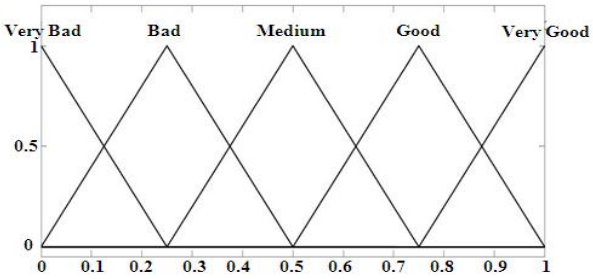

Figure 8. Membership Function for Weather Conditions

2.2.3. *Membership Function Curves for Output Variables.* The output variables 'activity duration' and 'activity cost' have been modeled by 7 membership functions over the universe of discourse (UOD) for each activity of the project network. For one activity, membership function curves are shown in Figure 9 and Figure 10 for 'activity duration' and 'activity cost' respectively. The range of UOD for 'activity duration' and 'activity cost' are assumed between $(D - 0.2 \times D \ to \ D + 0.2 \times D)$ and $(C - 0.2 \times C \ to \ C + 0.2 \times C)$ where D and C takes 'normal duration', 'crash duration' and 'normal cost', 'crash cost' for each activity of the project network.

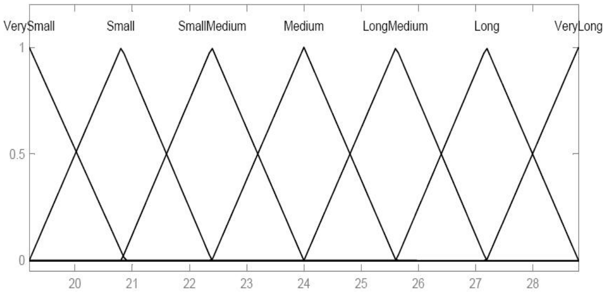

Figure 9. Membership Function for Activity Duration

2.2.4. *Fuzzy rules set.* Since each input linguistic variable is modeled using five membership functions and each output linguistic variable is modeled using 7 membership functions, thus total 6125 $(5 \times 5 \times 5 \times 7 \times 7)$ fuzzy rules sets are possible for each activity of the project network. In this study, 252 fuzzy rules sets are designed for each activity. It is important to note that 252 fuzzy rules sets are decided by performing the experiments on various combinations of fuzzy rules set and found that this number gives the good results. These fuzzy rules sets comprises of 126 fuzzy rules sets for each activity of project network within 'normal time' and 'normal cost' and 126 fuzzy rules sets for each activity of project network within 'crash time' and 'crash cost'. For example, If ME is 'very less', LS is 'very low' and WC is 'bad' then 'activity duration' is 'very long' and 'activity cost' is 'very large' is

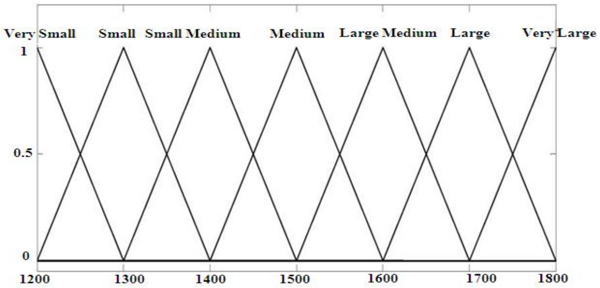

Figure 10. Membership Function for Activity Cost

one such rule. Fuzzy rules sets are constructed by taking different combinations of linguistic input variables (ME, LS, WC) and linguistic output variables ('activity cost', 'activity duration').

2.2.5. *Defuzzification.* Defuzzification is applied to each linguistic time-cost pair corresponding to the normal and crash conditions to transform a new time-cost pair, which reflects the impact of fuzzy conditions. This new time-cost pair is referred to as 'fuzzy time-cost' pair.

2.2.6. *Pseudo-code for designing FIS.* In this section, designing of FIS is explained with step-by-step:

1. Set the number of activities of project network = *Act* (say) and $I = 1$ (I is for increment).
2. Using FIS-Editor, add three input variables namely ME, LS, WC in range [0, 1] and two output variables namely 'activity duration' and 'activity cost' in range $[D - 0.2 \times D,\ D + 0.2 \times D]$ and $[C - 0.2 \times C,\ C + 0.2 \times C]$ respectively as shown in Figure 5.
3. Add five membership functions for each input linguistic variable as shown in Figure 6, Figure 7 and Figure 8 respectively and add seven membership functions for each output linguistic variable as shown in Figure 9 and Figure 10 respectively.
4. Design '252' fuzzy rules sets for I^{th} activity as mentioned in section 2.2.4.
5. Defuzzification is applied to obtained the 'fuzzy time-cost' pair.
6. $I = I + 1$
7. Repeat steps 2 to 5 until $I \neq Act$.

2.3. **Working of Fuzzy-ANN-HMH Approach.** Fusion of Fuzzy and ANN with HMH technique is explained with the following steps:

1. Input project uncertainties (ME, LS, WC) in Fuzzy Logic Framework (refer section 2.2), which generates a set of 'fuzzy time-cost' pair for each activity of a given project.
2. Set the initial parameters: $T^{(1)}$, $T^{(f)}$, n_c, n_p, and mutation rate (p_m).

3. Set the number of activities (n) in the project network and define their precedence relationship and get normal time (NT), normal cost (CT), crash time (CT), and crash cost (CC) for each activity from fuzzy logic framework as mentioned in step 1.
4. Generate initial n_p strings (refer section 2.1.2) each of length n. Find the maximum path time (critical path time) and the corresponding cost by summing the cost of each activity, which is provided by ANN. These n_p strings are referred to as 'parents'.
5. Each string generates $|Q| = \frac{n_c}{n_p}$ number of offsprings by performing crossover and mutation with other strings of parents.
6. Determine the time and cost of each offspring-string as mentioned in step 4.
7. Combine the parent-strings and their corresponding offsprings, which makes n_p families.
8. 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.
9. To decide the number of the offsprings for the next generation, *simulated annealing* has been incorporated (refer section 2.1.10).
10. Generate the required number of offspring for each parent by performing crossover and mutation with the other strings of parents.
11. Update the temperature.
12. Repeat steps 7 to 11 until the tradeoff curve remains identical or for a pre-specified number of iterations.
13. Repeat steps 1 to 12 until the analysis of project uncertainties is not completed.

3. **Results and Discussion**

Nonlinear TCT problem, defined by equations (1) to (3), is solved using the proposed Fuzzy-ANN-HMH approach. To illustrate the concept and the effectiveness of Fuzzy-ANN-HMH approach, two case studies which closely represent real world problems are taken from literature with suitable modifications. The first case study involving 7 activities and second case study involving 18 activities are taken from [10] and [6] respectively. Figure 11 and Figure 12 show the precedence relationship of every activity of project network for each of the case studies respectively. The different options for the time and cost of each activity of these networks are shown in Table 1 and Table 2 respectively. To solve these case studies using Fuzzy-ANN-HMH approach, 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. 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 number of iterations are reached upto 1000 for the both case studies.

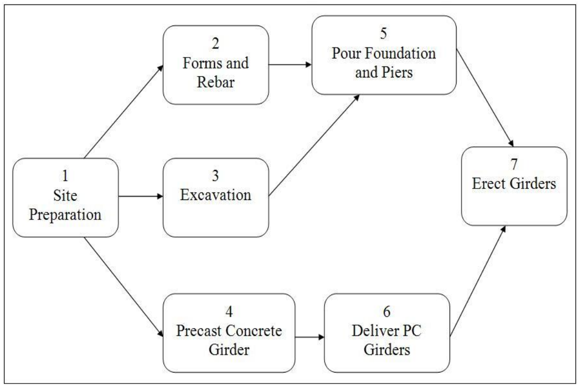

Figure 11. Network of First Case Study

Activity	Options						
number	I	II	III	IV	V	VI	VII
	(Time, Cost)	(Time, Cost)	(Time, Cost)	(Time, Cost)	(Time, Cost)	(Time, Cost)	(Time, Cost)
1	(14, 2400)	(15, 2167)	(18, 1900)	(20, 1800)	(22, 1610)	(23, 1300)	(24, 1200)
2	(15, 3000)	(17, 2640)	(18, 2400)	(20, 1800)	(23, 1500)	(24, 1350)	(25, 1000)
3	(15, 4500)	(17, 4257)	(21, 4160)	(22, 4000)	(25, 3692)	(30, 3489)	(33, 3200)
4	(12, 45000)	(14, 40127)	(15, 37900)	(16, 35000)	(17, 33000)	(18, 31995)	(20, 30000)
5	(22, 20000)	(24, 17500)	(25, 16800)	(26, 16250)	(28, 15000)	(29, 12875)	(30, 10000)
6	(14, 40000)	(15, 38000)	(17, 34500)	(18, 32000)	(20, 27533)	(22, 22307)	(24, 18000)
7	(09, 30000)	(11, 27340)	(12, 26990)	(14, 25098)	(15, 24000)	(17, 23667)	(18, 22000)

Table 1. Network Data of First Case Study

3.1. **ANN Training for Activities.** A feed forward back propagation neural networks with LM rule for all the activities are trained on time-cost data of first and second case study respectively. One network is trained for every activity, thus a total of 7 and 18 ANNs are employed for each case study respectively. A neural network architecture as shown in Figure 2, with one input-'activity time' and one output-'activity cost' is used for training the data and error goal of 10^{-5} is specified. Training data for each ANN is prepared by picking up five time-cost options from Table 1 and Table 2 respectively and two time-cost options are obtained from fuzzy logic toolbox (refer section 2.2.6). Thus total seven options are used for training of each ANN. These seven options for each ANN comprises of two options with minimum and maximum time and their corresponding cost and five options are selected at random out of remaining seven options. Remaining two options of each

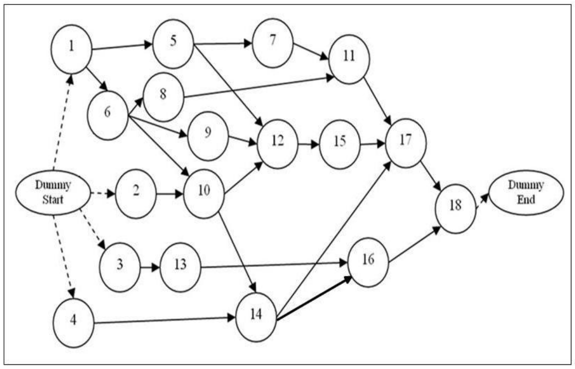

Figure 12. Network of Second Case Study

Activity number	Options						
	I	II	III	IV	V	VI	VII
	(Time, Cost)	(Time, Cost)	(Time, Cost)	(Time, Cost)	(Time, Cost)	(Time, Cost)	(Time, Cost)
1	(14, 2400)	(15, 2150)	(16, 1900)	(18, 1750)	(21, 1500)	(23, 1340)	(24, 1200)
2	(15, 3000)	(17, 2630)	(18, 2400)	(20, 1800)	(21, 1720)	(23, 1500)	(25, 1000)
3	(15, 4500)	(17, 4415)	(19, 4220)	(22, 4000)	(25, 3730)	(30, 3375)	(33, 3200)
4	(12, 45000)	(13, 44300)	(15, 38450)	(16, 35000)	(18, 33700)	(19, 32400)	(20, 30000)
5	(22, 20000)	(24, 17500)	(25, 16400)	(26, 15900)	(27, 15700)	(28, 15000)	(30, 10000)
6	(14, 40000)	(16, 39200)	(17, 34500)	(18, 32000)	(20, 27700)	(22, 20300)	(24, 18000)
7	(09, 30000)	(11, 27200)	(13, 26100)	(14, 25600)	(15, 24000)	(17, 22300)	(18, 22000)
8	(14, 220)	(15, 215)	(16, 200)	(17, 190)	(21, 167)	(23, 150)	(24, 120)
9	(15, 300)	(18, 240)	(20, 180)	(23, 150)	(24, 130)	(25, 110)	(25, 100)
10	(15, 450)	(22, 400)	(23, 390)	(27, 345)	(28, 330)	(30, 325)	(33, 320)
11	(12, 450)	(13, 420)	(14, 370)	(16, 350)	(17, 330)	(19, 305)	(20, 300)
12	(22, 2000)	(24, 1750)	(25, 1690)	(27, 1525)	(28, 1500)	(29, 1200)	(30, 1000)
13	(14, 4000)	(15, 3795)	(16, 3500)	(18, 3200)	(21, 2750)	(23, 2155)	(24, 1800)
14	(09, 3000)	(10, 2930)	(12, 2825)	(14, 2605)	(15, 2400)	(17, 2295)	(18, 2200)
15	(10, 6525)	(13, 5990)	(14, 4500)	(16, 3500)	(17, 3355)	(18, 2600)	(20, 1930)
16	(20, 3000)	(22, 2000)	(24, 1750)	(26, 1685)	(28, 1500)	(29, 1385)	(30, 1000)
17	(14, 4000)	(16, 3700)	(17, 3455)	(18, 3200)	(21, 2780)	(23, 2335)	(25, 1800)
18	(09, 3000)	(10, 2900)	(12, 2790)	(14, 2565)	(15, 2400)	(16, 2315)	(18, 2200)

Table 2. Network Data of Second Case Study

activity are used as testing data for the neural network. The training effort is very less with LM learning rule; it takes 4 to 6 iterations only (Figure 13). 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.

3.2. **Effects of Project Uncertainties on Case Studies.** Fuzzy-ANN-HMH approach starts within normal conditions of the linguistic variables ME, LS and WC. After getting the TCT profile from Fuzzy-ANN-HMH approach within normal conditions, effect analysis is made by changing the values of linguistic variables ME, LS, WC. If the values of these linguistic variables are greater than (0.5, 0.5, 0.5)

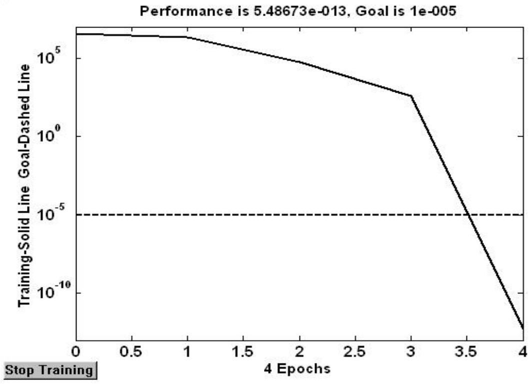

Figure 13. ANN Training for 1^{st} Activity of First Case Study

i.e. better than normal conditions, the profile moves towards the coordinate axes i.e. project duration and cost are reduced and vice versa. The effects of project uncertainties such as ME, LS, WS on 'project time' and 'project cost' for the first and second case study are shown in Figure 14 and Figure 15 respectively and their corresponding statistical analysis has been shown in Table 3 to improve the clarity of the results. These case studies demonstrates that project managers can analyze the effects of real life uncertainties involved in engineering projects and can help to choose the best alternative over the nonlinear TCT profiles.

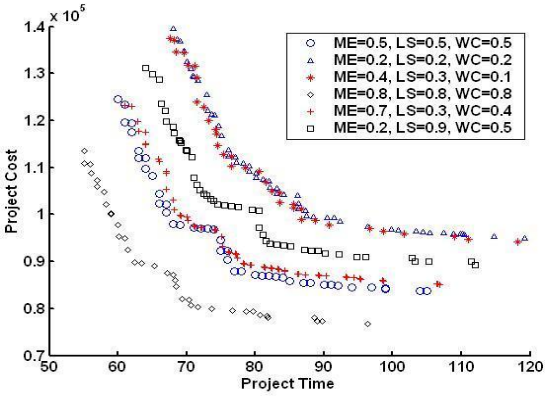

Figure 14. Effects of Linguistic Variables on Tradeoff Profile: First Case Study

3.3. **Comparison of Fuzzy-ANN-HMH with IFAG.** Multiobjective optimization techniques including MOEA techniques such as MOGA or HMH involve determining Pareto-optimal solutions, out of which one or more are selected by a

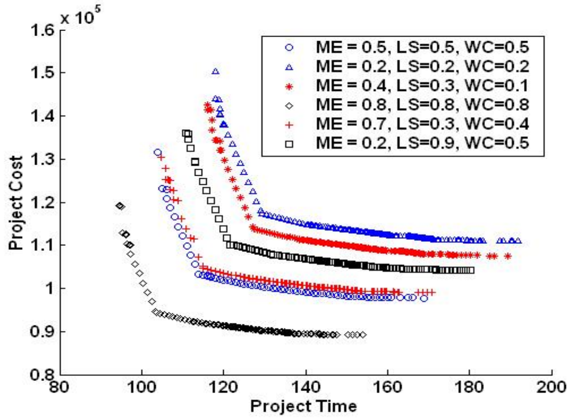

Figure 15. Effects of Linguistic Variables on Tradeoff Profile: Second Case Study

Statistics	First Case Study		Second Case Study		Uncertain Variables
	Project Time	Project Cost	Project Time	Project Cost	
Min	60	83720	104	97761	ME=0.5, LS=0.5, WC=0.5 Normal conditions
Max	105	124600	169	131550	
Mean	77.12	96520	133.65	102600	
Std	12.9	12530	18.47	7257.8	
Min	68.1	95010	118.03	110940	ME=0.2, LS=0.2, WC=0.2 Very bad conditions
Max	119.2	139500	191.82	150260	
Mean	85.75	109800	157.48	114380	
Std	14.09	13460	16.59	6486.6	
Min	67.55	94240	116.09	107570	ME=0.4, LS=0.3, WC=0.1 Bad conditions
Max	118.2	137400	189.27	142660	
Mean	82.49	111800	151.63	112200	
Std	13.05	14000	16.99	8011.9	
Min	55.03	76800	94.40	89210	ME=0.8, LS=0.8, WC=0.8 Very good conditions
Max	96.31	113500	154.01	119270	
Mean	68.74	90190	124.99	92641	
Std	11.38	11780	14.26	6314.9	
Min	61.06	85170	104.84	99066	ME=0.7, LS=0.3, WC=0.4 Combination of good and bad conditions
Max	106.8	123200	170.95	130380	
Mean	78.85	95900	140.09	102530	
Std	11.99	10930	16.10	6404.7	
Min	63.99	89290	110.91	104260	ME=0.2, LS=0.9, WC=0.5 Combination of bad, good and normal conditions
Max	112	131100	180.24	135940	
Mean	79.8	103800	146.25	108410	
Std	12.74	12270	18.11	7160.50	

Table 3. Statistical Analysis of Case Studies

decision-maker to suit decision environment. Therefore it is hard to compare the performance of two or more MOEA. Further, fusions like Fuzzy-ANN-HMH are unique in the literature to solve nonlinear multiobjective optimization problems in the presence of uncertainties. However, for evaluation purpose, Fuzzy-ANN-HMH is compared with another method in literature [16], called integrated fuzzy-ANN-GA (IFAG). The comparison of Fuzzy-ANN-HMH and IFAG is carried out for the both case studies with mean values of uncertainties (i.e. ME=0.5, LS=0.5, WC=0.5). Genetic parameters are kept same i.e. initial population for the first generation (parents + offspring) and mutation rate (p_m) are selected as 200 and 0.02 respectively for both methods. It should be noted that the initial population in Fuzzy-ANN-HMH is 200 comprising 20 parents and 180 offspring together whereas in IFAG the total initial population is 200 comprising entirely parents. Both approaches are run for a maximum of 500 iterations. It is found that results obtained from Fuzzy-ANN-HMH approach are more diversified and also have better convergence (Figure 16 and Figure 17).

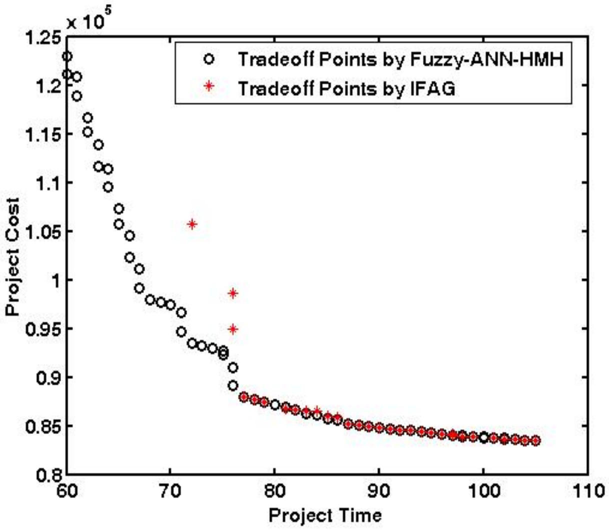

Figure 16. Tradeoff points by Fuzzy-ANN-HMH and IFAG with mean values of uncertainties: First case study

4. Conclusions

Fusion of fuzzy logic and ANN models with HMH technique is presented as an approach to solve nonlinear TCT analysis with commonly present uncertainties, and it does not place any restriction on time-cost relationship of project activities. Two case studies have been attempted to understand the feasibility of the Fuzzy-ANN-HMH and the results are compared with IFAG. It is evident from the visual results that the performance of Fuzzy-ANN-HMH is superior to IFAG in terms of convergence and diversity of solution points on TCT profile. Solutions obtained by Fuzzy-ANN-HMH for nonlinear TCT problems considered are well distributed on TCT profile and provide a large number of time-cost options. Therefore, the

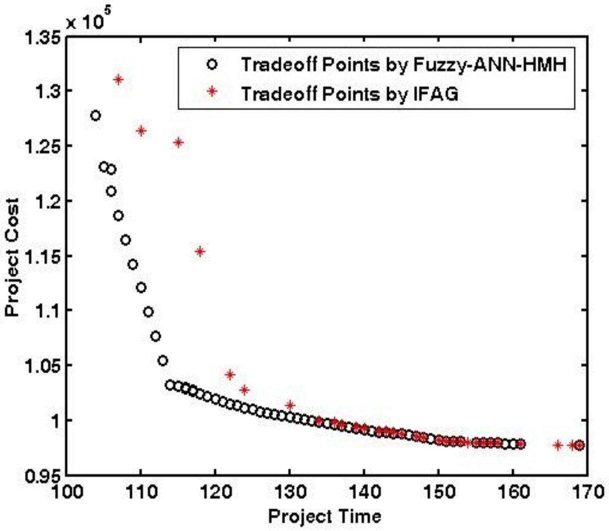

Figure 17. Tradeoff points by Fuzzy-ANN-HMH and IFAG with mean values of uncertainties: Second case study

proposed fusion provides a tool to project managers to analyze the effects of project uncertainties on the nonlinear TCT profile and helps to choose the best alternative over nonlinear TCT profile to execute projects.

References

[1] M. V. Arias and C. A. C. Coello, *Asymptotic convergence of metaheurisitcs for multiobjective optimization problems*, Soft Comput., **10** (2005), 1001–1005.

[2] S. P. Chen, M. J. Tsai, *Time-cost trade-off analysis of project networks in fuzzy environments*, Eur. J. Oper.Res., **212(2)** (2011), 386–397.

[3] K. Deb, *Multi-objective optimization using evolutionary algorithms*, NewYork: JohnWiley and Sons, 2001.

[4] E. Eshtehardian, A. Afshar and R. Abbasnia, *Fuzzy-based MOGA approach to stochastic time cost trade-off problem*, Automation in Construction, **18** (2009), 692–701.

[5] L. Fausett, "Fundamentals of neural networks," *NJ: Prentice Hall Englewood Cliffs*, 1994.

[6] C. W. Feng, L. Liu, and S. A. Burns, *Using genetic algorithms to solve construction time-cost trade-off problems*, J. Comput. Civil Eng., **11** (1997), 184–189.

[7] H. Ke, W. Ma, X. Gao and W. Xu, *New fuzzy models for time-cost trade-off problem*, Fuzzy Optim Decis Making, **9** (2010), 219–231.

[8] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, *Optimization by simulated annealing*, Sci., **220** (1983), 671–680.

[9] S. S. Leu, A. T. Chen and C. H. Yang, *A GA-based fuzzy optimal model for construction time-cost trade-off*, Int. J. Project Manage., **19** (2001), 47–58.

[10] L. Liu, S. Burns and C. Feng, *Construction time-cost trade-off analysis using LP/IP hybrid method*, J. Const. Eng. Manage., **121(4)** (1995), 446–454.

[11] E. H. Mamdani, *Application of fuzzy logic to approximate reasoning using linguistic synthesis*, Trans. Comput., **26(12)** (1977), 1182–1191.

[12] E. H. Mamdani, *Applications of fuzzy algorithms for control of simple dynamic plant*, In: Proc. IEEE, **121(12)** (1974), 1585–1588.

[13] B. K. Pathak and S. Srivastava, *Integrated Fuzzy-HMH for project uncertainties in time-cost tradeoff problem*, Applied Soft Computing, **21** (2014), 320-329.

[14] B. K. Pathak and S. Srivastava, *MOGA based time-cost tradeoffs: responsiveness for project uncertainties*, In: Proc. CEC, (2007), 3085–3092.

[15] B. K. Pathak, S. Srivastava and K. Srivastava, *Neural network embedded multiobjective genetic algorithm to solve nonlinear time-cost tradeoff problems of project scheduling*, J. Sci. Ind. Res., **67(2)** (2008), 124–131.

[16] S. Srivastava, B. Pathak and K. Srivastava, *Project scheduling: time-cost tradeoff problems*, In: Comput. Intell.Optimization, Y. Tenne and C-K Goh Ed., Springer-Verlag Berlin Heidelberg, **7** (2010), 325–357.

[17] 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*, J. Sci. Ind. Res., **63** (2004), 997–1005.

[18] M. O. Suliman, V. S. S. Kumar and W. Abdulal, *Optimization of uncertain construction time-cost trade off problem using simulated annealing algorithm*, World Congr. Inform. Commun. Tech., (2011), 489–494.

[19] L. A. Zadeh, *Fuzzy sets*, Inform. and Control, **8(3)** (1965), 333–353.

[20] L. A. Zadeh, *Outline of a New Approach to the Analysis of a Complex System and Decision Processes*, IEEE Trans. Syst. Man Cybern., **3** (1973), 28–44.

Bhupendra Kumar Pathak *, Department of Mathematics, Jaypee University, Anoopshahr, Bulandshahr, 203390 India
E-mail address: pathak.maths@gmail.com

Sanjay Srivastava, Department of Mechanical Engineering, Dayalbagh Educational Institute (Deemed University), Dayalbagh, Agra, 282005 India
E-mail address: ssrivastava@dei.ac.in

*Corresponding author