International Journal of Applied Operational Research
Vol. 4, No. 2, pp. 53-82, Spring 2014

Journal homepage: www.ijorlu.ir

Solving a tri-objective convergent product network using the Steiner tree

R. Hassanzadeh*, I. Mahdavi

Received: 24 September 2013 ; **Accepted:** 4 February 2014

Abstract Considering convergent product as an important manufacturing technology for digital products, we integrate functions and sub-functions using a comprehensive fuzzy mathematical optimization process. To form the convergent product, a web-based fuzzy network is considered in which a collection of base functions and sub-functions configure the nodes and each arc in the network is to be a link between two nodes. The aim is to find an optimal tree of functionalities in the network adding value to the product in the web environment. First, a purification process is performed in the product network to assign the links among bases and sub-functions. Then, numerical values as benefits and costs are determined for the arc and the node, respectively. Also, a fuzzy customers' value corresponding to the arcs is considered. Next, the Steiner tree methodology is adapted to a multi-objective model of the network to find the optimal tree. A fuzzy multi-objective solution methodology is developed for solving the proposed problem. Finally, an example is worked out to illustrate the proposed approach.

Keywords: Convergent Product, Web-Based (Digital) Network, Fuzzy Multi-Objective Programming, Steiner Tree, Fuzzy Number.

1 Introduction

Convergence in electronics and communications sectors has enabled the addition of disparate new functionalities to existing base functions (e.g., adding mobile television to a cell phone or Internet access to a personal digital assistant, PDA). An important managerial issue for such convergent products (CPs) is determination of new functionalities adding more value to a given base. For example, a manufacturer of PDAs may wonder whether it would be a good idea to add satellite radio to it (i.e., a new functionality incongruent with the base), or whether it would be better to add electronic Yellow Pages (i.e., a new functionality congruent with the functions of a PDA). In addition, determining the significance of the base being primarily associated with utilitarian consumption goals (e.g., a PDA), or with hedonic ones (e.g., an MP3 music player) is important.

A convergent product is similar to product assembly where different parts of a product get together to configure a final product. Thus, a designer (modeler) for assembly, as a convergent product, should be able to specify important features affecting the final product. These features may in turn help optimize the manufacturing process.

---

* **Corresponding Author.** (✉)
E-mail: rhz_1974@yahoo.com (R. Hassanzadeh)

R. Hassanzadeh,
Ph.D. Student, Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran.

I. Mahdavi
Professor, Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran.

For instance, the ability of the assembly modeler to furnish information on interferences and clearances between mating parts is particularly useful. Such information would enable the designer to eliminate interference between two mating parts where it is impractical to provide for an interference based on physical assembly requirements. This activity can be accomplished within the modeling program, thereby averting any loss of productivity that might occur due to interferences on the shop floor. Also, knowledge of mass properties for the entire assembly, particularly the center of gravity, may permit the designer to redesign the assembly based on equilibrium and stability considerations. In the absence of such information, the presence of an elevated center of gravity and the attendant instability would only be detected after physical assembly on the shop floor. Three-dimensional exploded views generated by the assembly modeler can help designers verify whether obvious violations of common design for assembly (DFA) guidelines are present, such as absence of chamfers on mating parts.

Corresponding analyses can be achieved within the framework of the assembly modeler. Additionally, the assembly model may be imported into third-party programs that can perform kinematic, dynamic, or tolerance analysis. Tolerance analysis is quite relevant to the physical assembly process. With the input of the assembly model and other user-supplied information such as individual part tolerances, tolerance analysis programs can check the assembly for the presence of tolerance stacks. Tolerance stacks are undesirable elements in the sense that acceptable tolerances on individual parts are combined to produce an unacceptable dimensional relationship, thereby resulting in a malfunctioning or nonfunctioning assembly. Stacks are usually discovered during physical assembly, at which point any remedial procedure becomes expensive in terms of time and cost. Tolerance analysis programs can help the user eliminate or significantly reduce the likelihood of stacks being present.

Based on the results of the tolerance analysis, assembly designs may be optimized by modifying individual part tolerances. Note, however, that tolerance modifications have cost implications; in general, tighter tolerances increase production costs. Engineering handbooks contain tolerance charts indicating the range of tolerances achieved by manufacturing processes such as turning, milling, and grinding. Designers use these tables as guides for rationally assigning part tolerances and selecting manufacturing processes.

A more effective methodology for optimizing product assembly and convergent product is the tree model, whereas the optimization decision is based on a decision tree. One useful tree for assembly modeling as a multiple optimization tool is the Steiner tree.

The Steiner tree problem (STP) is a much actively investigated problem in graph theory and combinatorial optimization. This core problem poses significant algorithmic challenges and arises in several applications where it serves as a building block for many complex network design problems. Given a connected undirected graph $G=(V,E)$, where V denotes the set of nodes and E is the set of edges, along with a weight C_e associated with each edge $e \in E$, the Steiner tree problem seeks a minimum-weight subtree of G that spans a specified subset $N \subset V$ of *terminal nodes*, optionally using the subset $N=V–N$ of *Steiner nodes*. The Steiner tree problem is NP-hard for most relevant classes of graphs; see [1].

The Steiner problem in graphs was originally formulated by Hakimi [2]. Since then, the problem has received considerable attention in the literature. Several exact algorithms and heuristics have been proposed and discussed. Hakimi remarked that an *Steiner minimal tree* (SMT) for X in a network $G=(V,E)$ can be found by enumerating minimum spanning trees of subgraphs of G induced by supersets of X. Lawler [3] suggested a modification of this algorithm, using the fact that the number of Steiner points is bounded by $|X|-2$, showing that not all subsets of V need to be considered. Restricting NP-hard algorithmic problems regarding arbitrary graphs to a smaller class of graphs will sometimes, yet not always, result in polynomially solvable problems.

Two special cases of the problem, $N = V$ and $N = 2$, can be solved by polynomial time algorithms. When $N = V$, the optimal solution of STP is obviously the spanning tree of G and thus the problem can be solved by polynomial time algorithms such as Prim's algorithm. When $N = 2$, the shortest path between two terminal nodes, which can be found by Dijkstra's algorithm, is exactly the Steiner minimum tree.

A survey of Steiner tree problems was given by Hwang and Richards [4]. Several exact algorithms have been proposed such as the dynamic programming technique by Dreyfuss and Wagner [5], Lagrangean relaxation approach by Beasley [6], and brand-and-cut algorithm by Koch and Martin [7]. Duin and Volgenant [8] presented some techniques to reduce the size of the graphs for the GSP. Another approach for the GSP is using approximation algorithms to find a near-optimal solution in a reasonable time.

Some heuristic algorithms have been developed such as Shortest Path Heuristic (SPH) by Takahashi and Matsuyama [9], Distance Network Heuristic (DNH) by Kou et al. [10], Average Distance Heuristic (ADH) by Rayward-Smith and Clare [11] and Path-Distance Heuristic (PDH) by Winter and MacGregor Smith [12]. Mehlhorn [13] modified DNH to arrive at a more efficient algorithm. Robins and Zelikovsky [14, 15] proposed algorithms improving the performance ratio.

Recently, metaheuristics have been considered to propose better methods for finding near optimal solutions. Examples are Genetic Algorithm (GA) [16, 17], GRASP [18] and Tabu search [19]. Although these algorithms have polynomial time complexities, in general, but they cost enormously on large scale input sets. To deal with the cost issue, some parallel metaheuristic algorithms have been proposed such as parallel GRASP [20], parallel GRASP using hybrid local search [21] and parallel GA [22].

Here, making use of the Steiner tree, a fuzzy multi-objective mathematical model is developed for the convergent product. The remainder of our work is organized as follows. In Section 2, some elementary concepts and related operations of fuzzy set theory are provided. In Section 3, the proposed model is described and some useful network algorithms are given. Section 4 presents the mathematical model and a solution algorithm. Section 5 works out an experimental study to illustrate the proposed algorithm. We conclude in Section 6.

2 Definitions and Preliminaries

Here, we provide some basic definitions of fuzzy sets and fuzzy numbers, adopted from [23-25].

Definition 1. A fuzzy set $\tilde{A}$ in a universe of discourse X is characterized by a membership function $\mu_{\tilde{A}}(x)$ which associates with each element x in X a real number in the interval [0, 1]. The function value $\mu_{\tilde{A}}(x)$ is termed the grade of membership of x in $\tilde{A}$.

Definition 2. A fuzzy number $\tilde{A}$ is a fuzzy convex subset of the real line satisfying the following conditions:

(a) $\mu_{\tilde{A}}(x)$ is piecewise continuous.

(b) $\mu_{\tilde{A}}(x)$ is normalized; that is, there exists $m \in R$ with $\mu_{\tilde{A}}(m) = 1$, where m is called the mean value of $\tilde{A}$.

Definition 3. A triangular fuzzy number $\tilde{a}$ is defined by a triplet (a_1, a_2, a_3), where $a_i \in R, 1 \le i \le 3$, and its conceptual schema and mathematical form are shown by (1) below:

$$\mu_{\tilde{a}}(x) = \begin{cases} 0, & x \le a_1 \\ \dfrac{x - a_1}{a_2 - a_1}, & a_1 \le x \le a_2 \\ \dfrac{a_3 - x}{a_3 - a_2}, & a_2 \le x \le a_3 \\ 0, & a_3 \le x \end{cases} \tag{1}$$

A triangular fuzzy number $\tilde{a}$ in the universe of discourse X that conforms to this definition is shown in Fig. 1.

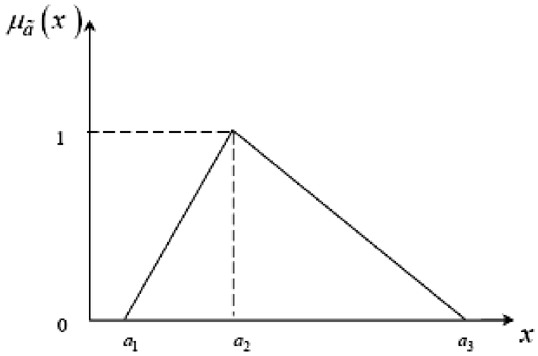

Fig. 1 A triangular fuzzy number $\tilde{a}$.

Definition 4. A trapezoidal fuzzy number $\tilde{a}$ can be defined by a quadruplet (a_1, a_2, a_3, a_4), where $a_i \in R, 1 \le i \le 4$, and its conceptual schema and mathematical form are shown by (2) below:

$$\mu_{\tilde{a}}(x) = \begin{cases} 0, & x \le a_1 \\ \dfrac{x - a_1}{a_2 - a_1}, & a_1 \le x \le a_2 \\ 1, & a_2 \le x \le a_3 \\ \dfrac{a_4 - x}{a_4 - a_3}, & a_3 \le x \le a_4 \\ 0, & a_4 \le x. \end{cases} \tag{2}$$

A trapezoidal fuzzy number $\tilde{a}$ in the universe of discourse X that conforms to this definition is shown in Fig. 2.

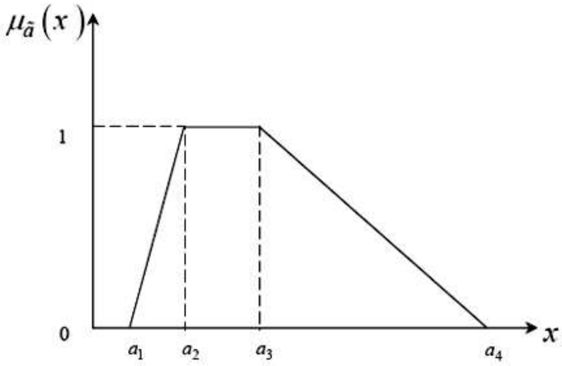

Fig. 2 A trapezoidal fuzzy number $\tilde{a}$.

Note that one can consider a triangular fuzzy number as a special case of a trapezoidal fuzzy number by having $b_2 = b_3$.

Definition 5. For $\tilde{a} = (a_1, a_2, a_3)$, and $\tilde{b} = (b_1, b_2, b_3)$, triangular fuzzy numbers, the fuzzy sum is given by:

$$\tilde{a} + \tilde{b} = (a_1 + b_1, a_2 + b_2, a_3 + b_3), \tag{3}$$

and for $\tilde{a} = (a_1, a_2, a_3, a_4)$, and $\tilde{b} = (b_1, b_2, b_3, b_4)$, trapezoidal fuzzy numbers, the fuzzy sum of the two numbers is given by:

$$\tilde{a} + \tilde{b} = (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4).$$

Definition 6. The α-cut and strong α-cut for a fuzzy set $\tilde{A}$ are shown by $\tilde{A}_\alpha$ and $\tilde{A}_\alpha^+$, respectively, and for $\alpha \in [0,1]$ are defined to be:

$$\tilde{A}_\alpha = \left\{x \,\middle|\, \mu_{\tilde{A}}(x) \geq \alpha, x \in X\right\},$$
$$\tilde{A}_\alpha^+ = \left\{x \,\middle|\, \mu_{\tilde{A}}(x) > \alpha, x \in X\right\},$$

where X is the universal set.

Upper and lower bounds for any α-cut $\tilde{A}_\alpha$ are shown by $\sup \tilde{A}_\alpha$ and $\inf \tilde{A}_\alpha$, respectively. Here, we assume that the upper and lower bounds of α-cuts are finite values and for simplicity we denote $\sup \tilde{A}_\alpha$ by $\tilde{A}_\alpha^R$ and $\inf \tilde{A}_\alpha$ by $\tilde{A}_\alpha^L$.

Needing to find minimal values amongst fuzzy numbers, we should have a method of ranking the fuzzy numbers for comparison purposes. For this reason, ranking or ordering methods of fuzzy quantities have been proposed by several authors. Unfortunately, none of these methods is commonly accepted.

Here, we make use of a new ranking method for fuzzy numbers proposed by Mahdavi et al. [26]. For define the fuzzy min operation similar to the fuzzy addition as follows:

$$MV = MV(\tilde{a}, \tilde{b}) = \min(\tilde{a}, \tilde{b}) = (\min(a_1, b_1), \min(a_2, b_2), \min(a_3, b_3), \min(a_4, b_4)). \tag{4}$$

It is evident that, for non-comparable fuzzy numbers $\tilde{a}$ and $\tilde{b}$, this fuzzy min operation results in a fuzzy number different from both $\tilde{a}$ and $\tilde{b}$. For example, for $\tilde{a} = (5,10,13,19)$ and $\tilde{b} = (6,9,15,20)$, using (4) ,we get $\tilde{MV} = \min(\tilde{a},\tilde{b}) = (5,9,13,19)$, which is different from both $\tilde{a}$ and $\tilde{b}$. To avert this drawback, Mahdavi et al. [26] offered a new approach based on the distance between fuzzy numbers, proposed in [27].

Definition 7. The $D_{p,q}$-distance, indexed by parameters p, $1 < p < \infty$, and q, $0 < q < 1$, between two fuzzy numbers $\tilde{a}$ and $\tilde{b}$ is a nonnegative function given as follows:

$$D_{p,q}(\tilde{a},\tilde{b}) = \begin{cases} [(1-q)\int_0^1 \left|a_\alpha^- - b_\alpha^-\right|^p d\alpha + q\int_0^1 \left|a_\alpha^+ - b_\alpha^+\right|^p d\alpha]^{\frac{1}{p}} & , \ p < \infty \\ (1-q)\sup_{0<\alpha\le 1}(\left|a_\alpha^- - b_\alpha^-\right|) + q \inf_{0<\alpha\le 1}(\left|a_\alpha^+ - b_\alpha^+\right|) & , \ p = \infty \end{cases} \tag{5}$$

The analytical properties of $D_{p,q}$ depend on the first parameter *p*, while the second parameter *q* of $D_{p,q}$ characterizes the subjective weight attributed to the end points of the support of the fuzzy numbers. If there is no reason for distinguishing any side of the fuzzy numbers, $D_{2,\frac{1}{2}}$ is recommended.

For triangular fuzzy numbers $\tilde{a} = (a_1, a_2, a_3)$, and $\tilde{b} = (b_1, b_2, b_3)$, the above distance with $p = 2$ and $q = \frac{1}{2}$ is calculated to be:

$$D_{2,\frac{1}{2}}(\tilde{a},\tilde{b}) = \sqrt{\frac{1}{6}[\sum_{i=1}^{3}(b_i - a_i)^2 + (b_2 - a_2)^2 + \sum_{i\in\{1,2\}}(b_i - a_i)(b_{i+1} - a_{i+1})} . \tag{6}$$

And if $\tilde{a} = (a_1, a_2, a_3, a_4)$, and $\tilde{b} = (b_1, b_2, b_3, b_4)$, are trapezoidal fuzzy numbers, then the distance is calculated as:

$$D_{2,\frac{1}{2}}(\tilde{a},\tilde{b}) = \sqrt{\frac{1}{6}[\sum_{i=1}^{4}(b_i - a_i)^2 + \sum_{i\in\{1,3\}}(b_i - a_i)(b_{i+1} - a_{i+1})} . \tag{7}$$

Now, we make use of the distance function (7) to acquire the distances of $\tilde{a}$ and $\tilde{b}$ to MV, as obtained by (4). We then compute $D_{2,\frac{1}{2}}(\tilde{a}, MV) = 0.1667$ and $D_{2,\frac{1}{2}}(\tilde{b}, MV) = 1.33$. Thus, $\tilde{a}$, having a lower distance from MV, is decided to be smaller than $\tilde{b}$.

3 The Proposed Model

In our proposed product digital network, a group of functionalities are considered for a product. Customers view their opinions for classifying the functionalities into base functions and sub-functions. We make use of this classification in developing our model. The classification procedure is as follows. First, the customer chooses a product in a list of products being produced in a company. The functionalities of the product are viewed in a web page. Then, the customer clicks either function or sub-function for any of the functionalities. Consequently, customer clicks the "classify" button and observes the classified functionalities in a separate web page. This process is shown in Fig. 3.

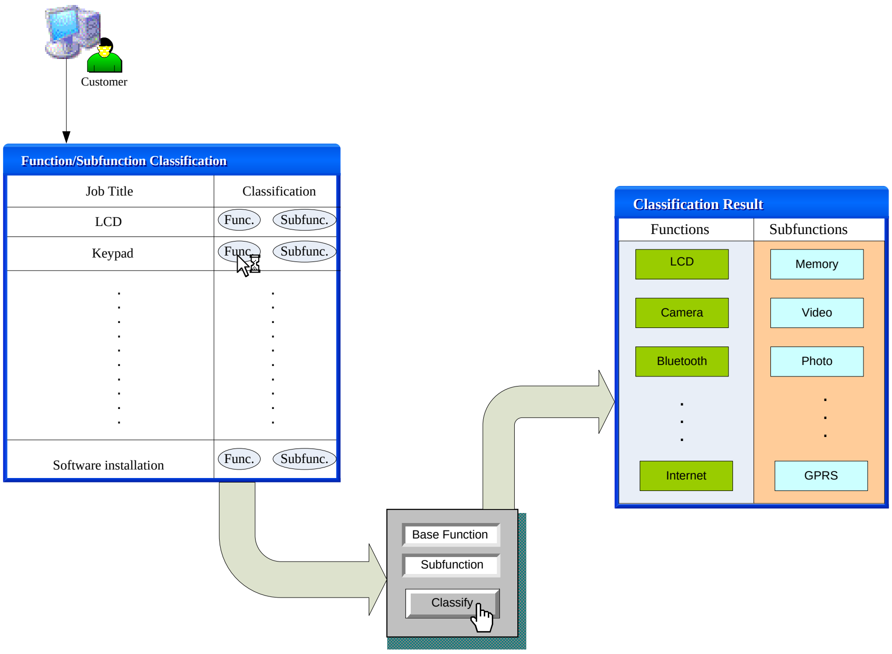

Fig. 3 The classification process

Here, we weigh all functionalities (both base functions and sub-functions) considering different significant attributes affecting the value of a product. Therefore, we consider the following mathematical notations.

Mathematical Notations:

i and j	Index for functions and sub-functions;	i and j=1,…,n+m
k	Index for attributes;	k=1,…,p
$\tilde{F}_{ijk}$	The fuzzy score of triplet comparison of functions (or sub-functions) with functions(or sub-functions) considering different attributes.	

The three dimension comparison fuzzy matrix $\tilde{F}$ is shown in Fig. 4. Note that customers fill in this matrix using triangular fuzzy numbers.

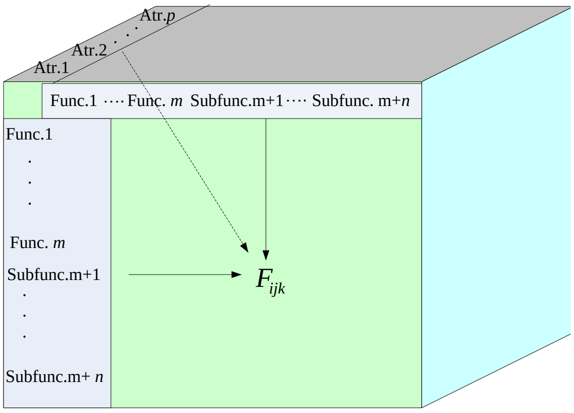

Fig. 4 The three dimensional comparison matrix

This matrix is normalized to remove the scales. The normalized values are shown by $\overset{norm}{\tilde{F}_{ijk}}$. A threshold value of θ is considered in a way that the $D_{2,\frac{1}{2}}(\overset{norm}{\tilde{F}_{ijk}}, MV) \geq \theta$ are chosen to be assigned as links. These links configure a network called purified network as shown in Fig. 5.

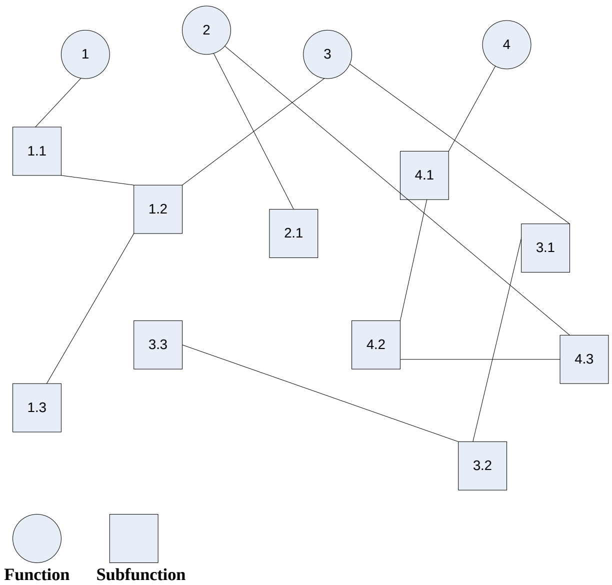

Fig. 5 A purified network

Now, using the purified network, we characterize the arcs. To do this, two processes of leveling and clustering are performed. For leveling, we set the base functions at level zero, sub-functions with one outlet to the previous level at level 1, and so on. Thus, an l level network is configured. The proposed algorithm is given next.

Algorithm 1: Leveling to configure a leveled network.

Step 1: Set the base functions at level 0. Let l=0.
Step 2: **While** sub-functions exist for processing **do**

Find sub-functions with a link to a function (or sub-function) at level l and put them in level l+1. Let l=l+1;

End while.

{l is the number of levels.}

Step 3: **Stop**.

The nodes of the leveled network are associated with given costs. We are looking for the benefit each link provides. Here, a clustering approach is considered. Clusters are formed as follows: at each level, all sub-functions linked to a single parent is grouped in a cluster. Therefore, clusters consisting different nodes are configured. These clusters are being configured as a new network. The leveling and clustering processes are shown schematically in Fig. 6. Later, we apply the Steiner tree methodology to optimize this network.
The proposed algorithm for clustering is given next.

Algorithm 2: Clustering of levels in a leveled network.

Step 1: Set each node at level 0 to be a cluster.

Step 2: **For** i=1 till l **do**

{Form clusters at level i}

Cluster all sub-functions at level i linked to a single parent at level i-1;

Solve a zero/one mathematical program for level l (we will discuss the corresponding mathematical program later on);

Perform purification of benefits and costs at level l (as discussed later on);

End for.

Step 3: **Stop**.

Here, the clustered network is used to configure a tree (the Steiner tree) keeping the base functions and optimizing three objectives of minimal cost, maximal profit and maximize customer’s total value in the convergent product value adding process. In traditional Steiner tree approach, the aim is usually to find a tree having a minimal arc total cost. Here, we extend the approach by looking for a tree having the base functions and minimizing cost, maximizing benefit and maximizing customer’s total value. In fact, the model structure’s closeness to the Steiner tree model justifies modeling the problem with the proposed approach. Next, we formulate our adapted proposed Steiner tree model. In the proposed network, node i (function or sub-function i) have two costs:

c_{i1}: software cost,

c_{i2}: hardware cost.

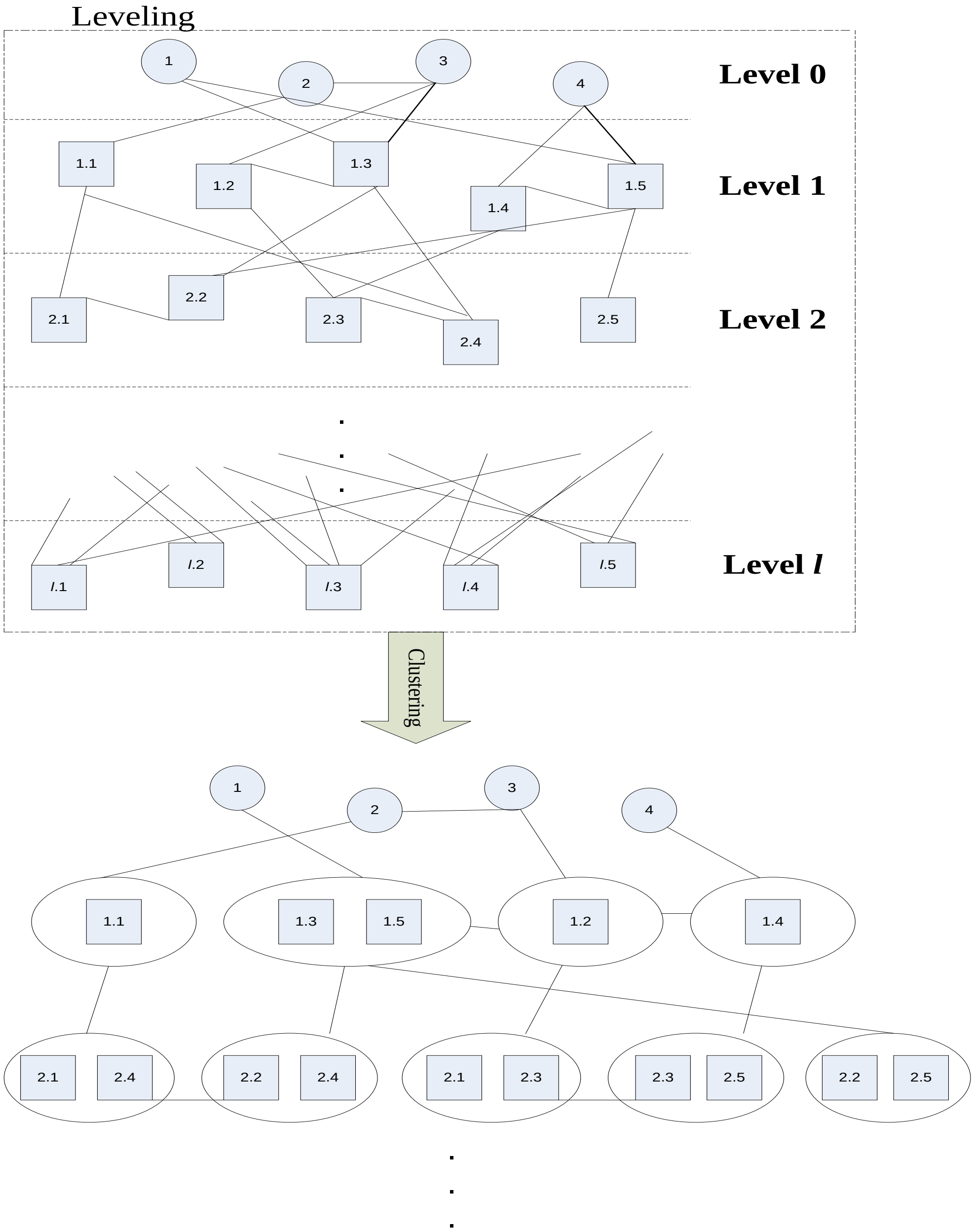

Fig. 6 Leveling and clustering processes

Each arc is accompanied with a benefit $p_{ii'}$, attained by nodes i and i'. Regarding the solution approach and using the Steiner tree in the proposed network and the NP-hardness of the problem, we used leveling and clustering processes to reduce the complexity of the problem. In clustering, it is not acceptable for any node to be included in more than one cluster at any level. To guarantee this, for each level l, a zero/one mathematical program is developed in order to properly appropriate nodes to clusters with the aim of minimizing the total cost.

Next, we give the zero/one mathematical program and the purification procedure for each level.

The zero/one mathematical program for level *l*:

$$Min \quad T = \sum_{i \in n_l} \sum_{j \in m_{li}} (\alpha_{ij} c_{ij1} + \beta_{ij} c_{ij2}) z_{ij}$$

$$s.t.$$

$$\sum_{j \in m_{li}} z_{ij} = 1, \qquad i = 1, \dots, n_l,$$

$$z_{ij} = \begin{cases} 1, & \text{if node } i \text{ is in cluster } j \\ 0, & \text{otherwise} \end{cases}$$

where, $\beta_{ij}, \alpha_{ij} \in [0,1], \forall i, j$, and
m_{li} : set of indices of clusters at level l where node i is included,
n_l : set of indices of different nodes at level l,
c_{ij2} : hardware cost of node i in cluster j ($c_{ij2}= c_{i2}$, for all i and j),
c_{ij1}: software cost of node i in cluster j ($c_{ij1}= c_{i1}$, for all i and j),
α_{ij} : the software reduction cost coefficient of node i in cluster j,
β_{ij} : the hardware reduction cost coefficient of node i in cluster j.

Purifying benefits, costs and customers total value at level *l*:

To determine the cost for each cluster at level l, we use

$$c_{jl} = \sum_{i \in n_l} (\alpha_{ij} c_{ij1} + \beta_{ij} c_{ij2}) - \sum_{\forall i, i'} p_{ii'},$$

with c_{ij1}, c_{ij2}, and n_l as defined above, $\beta_{ij}, \alpha_{ij} \in [0,1], \forall i, j$, and
c_{jl} : cost of cluster j at level l,
α_{ij} : the software reduction cost coefficient of node i in cluster j,
β_{ij} : the hardware reduction cost coefficient of node i in cluster j,
$p_{ii'}$: the benefit of an arc connecting node i in cluster j to node i' in cluster j.

Also, to adjust the combined arc benefits in clusters, the following equation is used:

$$p_{jj'} = (1 + \gamma_{jj'}) \sum_{\forall i, i'} p_{ii'}$$

where, $\forall i, i', j, j'$, and
$p_{jj'}$: the adjusted arc benefit connecting cluster j to cluster j',
$p_{ii'}$: the benefit of an arc connecting node i in cluster j to node i' in cluster j',
$\gamma_{jj'}$: the added value configured from nodes in clusters j and j'.

Also, to adjust the combined arc customers total value in clusters, the following equation is used:

$$\tilde{f}_{jj'} = (1+\gamma_{jj'})\sum_{\forall i,i'} \tilde{f}_{ii'}$$

where, $\forall i, i', j, j'$, and

$\tilde{f}_{jj'}$: the adjusted arc customers total value connecting cluster *j* to cluster j',

$\tilde{f}_{ii'}$: the customers total value of an arc connecting node *i* in cluster *j* to node i' in cluster j',

$\gamma_{jj'}$: the added value configured from nodes in clusters *j* and j'.

Algorithms 1 and 2 are transformed into Algorithm 3 using the aforementioned considerations. Also, each node should be in only one cluster at level *l*. The node having a minimal cost is chosen for the level *l*. Then, instead of using the zero/one mathematical program for level *l*, we can be used in step 3 of Algorithm 3. This leads to a reduction of computations by avoiding the need for using the zero/one programs.

Algorithm 3: leveling and clustering in the network.

Step 1: Set the base functions at level 0. Let *l*=0.
Step 2: **While** sub-functions exist for processing **do**
Find sub-functions with a link to a function (or sub-function) at level *l* and place them at level *l*+1; Let *l*=*l*+1;
End while.
{*l* is the number of levels}
Step 3: Set each node at level 0 to be a cluster.
Step 4: **For** *i*=1 till *l* **do**
{Form clusters at level *i*}
Cluster all sub-functions at level *i* linked to a single parent at level *i*-1;
While $|n_i| > 0$ **do**
Select $k \in n_i$ such that $\alpha_{kp}c_{kp1} + \beta_{kp}c_{kp2} = \min_{j\in m_{ik}}\{\alpha_{kj}c_{kj1} + \beta_{kj}c_{kj2}\}$. Set $z_{kp} = 1$, and $z_{kj} = 0$, $\forall j \in m_{ik}$, $j \neq p$;
set $n_i \leftarrow n_i - \{k\}$.
End while;
For *j*=1 till q_i **do** { q_i is the number of clusters in the level i}

$$c_{ji} = \sum_{i\in n_l}(\alpha_{ij}c_{ij1} + \beta_{ij}c_{ij2}) - \sum_{\forall i,i'} p_{ii'} \ ;$$

End for;
For *j*=1 till q_i **do**
For j'=1 till q_i **do**

$$p_{jj'} = (1+\gamma_{jj'})\sum_{\forall i,i'} p_{ii'} \ ;$$

$$\tilde{f}_{jj'} = (1+\gamma_{jj'})\sum_{\forall i,i'} \tilde{f}_{ii'} \ ;$$

End for;

End for;
End for.
Step 5: **Stop**.

4 Mathematical Formulation and Solution Method

Here, we first propose the mathematical model for the considered problem and then state a solution approach.

4.1 Mathematical formulation

We first recall the undirected Dantzig–Fulkerson–Johnson model for the convergent product Steiner tree problem (CPSTP) proposed in [28]. Let x_{ij} and y_i be binary variables associated with links $(i, j) \in E$ and clusters $i \in V$, respectively. Variable y_i is 1 if cluster i belongs to the solution, and is 0 otherwise. Similarly, variable x_{ij} is 1 if link (i, j) belongs to the solution, and is 0 otherwise. For $S \subseteq V$, define $E(S)$ as the set of links with both end nodes in S. Assume that terminals are the set N. The mathematical model can then be written as:

$$\text{Maximize} \quad \sum_{(i,j)\in E} p_{ij}.x_{ij} \,, \tag{8}$$

$$\text{Minimize} \quad \sum_{i\in V} c_i.y_i \,, \tag{9}$$

$$\text{Maximize} \quad \sum_{(i,j)\in E} \tilde{f}_{ij}.x_{ij} \,, \tag{10}$$

Such that

$$\sum_{(i,j)\in E} x_{ij} = \sum_{i\in V} y_i - 1, \tag{11}$$

$$\sum_{(i,j)\in E(S)} x_{ij} \le \sum_{i\in S-\{k\}} y_i \,, \qquad \forall k \in S \subseteq V,\ \forall S : |S| \ge 2, \tag{12}$$

$$y_h = 1, \qquad \forall h \in N \,, \tag{13}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i, j \in E \,, \tag{14}$$

$$y_i \in \{0,1\}, \qquad \forall i \in V \,. \tag{15}$$

The objectives are to maximize the aggregated benefits, minimize the aggregated costs and maximize the aggregated customers total value. Constraint in (11) guarantees that the number of clusters in a solution is equal to the number of links minus one, and constraints (12) are the connectivity constraints. The number of constraints (12) equals $2^{|V|} - |V| - 1$. As a result, the number of variables and constraints are increased exponentially with respect to the number of clusters. Constraints (13) impose the terminal clusters to exist in the tree. Relations (14) and (15) show the variable types.

4.2 Fuzzy multi-objective solution method

Numerous problems in economics, operations research, decision sciences, engineering and management sciences have mainly been studied from optimization point of view. As decision making is influenced by the disturbances of social and economical circumstances, a direct optimization approach is not always the most effective. This is because under such influences, most problems are ill-structured. In the real-world, as reaching the ideal solution is practically unrealistic, a decision-maker is usually satisfied with feasible solutions closest to the ideal solutions [29]. Thus, a satisfying approach is much better than an optimization one. The literature reveals variants of Multi-Objective Linear Programming (MOLP) models and their use in decision-making. For example, Karsak and Kuzgunkaya [30] proposed a fuzzy MOLP approach as an alternative to the classical mathematical programming formulation. They used triangular fuzzy numbers not considering the compromise approach during evaluation of candidate-alternatives. Further, Gao and Tang [31] proposed a MOLP model for purchasing the raw material in a large-scale steel plant. The ‘Point estimate weight-sums method’ was used in their work to solve the equations. The method converted the MOLP into a general LP problem and the solution was obtained by assigning positive weights only. Their method did not embed a fuzzy technique so as to deal with vagueness of the problem. Further, the efficacy of MOLP was justified by Downing and Ringuest [32]. They used Excel and Visual Basic to implement four different algorithms for MOLP. It was demonstrated that explicit and effective modeling of any decision-making process with MOLP algorithms improved the effectiveness of the process. Interactive ‘‘fuzzy linear programming’’ (FLP) and ‘‘fuzzy MOLP’’ methods were proposed by Liang [33, 34] for solving transportation planning problems considering fuzzy goals, available supply and forecast demand. Petrovic-Lazarevic and Abraham [35] attempted to point out the significance of applying a fuzzy approach to multi-objective decision methods in the process of organizing business activities. The analysis and modeling of the construction industry problem presented in Petrovic-Lazarevic and Abraham [35] are based on both linear objective functions and constraints in a form of linear membership functions.

Several researchers studied the multi-objective formulation of decision models. According to Dyson [36], fuzzy programming models should not be treated as new contributions to multiple objective decision-making methods, but rather as leads to new conventional decision methods. ‘‘Support for this thesis would require examples of new and effective fuzzy inspired multi-criteria methods’’ [36]. Buckley [37] reported a multiobjective ‘‘fully fuzzified LP’’ methodology. They transformed a MOLP problem into a single objective fuzzy LP problem and an evolutionary algorithm was used in order to generate non-dominated set. Triangular fuzzy numbers were used in their model. Another fuzzy approach, a generalization of max-min, averaging and the two-phase method, was delineated by Chen and Chou [38] to solve the MOLP problem. Guu and Wu [39] proposed a two-phase approach with equally weighted coefficients in order to yield an efficient solution for multiple objective programming problems. A revised version of the two-phase approach was described by Wu and Guu [40] in order to generate improved solutions for fuzzy MOLP (FMOLP) problems using min operator, but the model did not appear to be capable of handling enough imprecision arising in large-scale optimization problems.

4.3 The proposed FMOLP algorithm

The proposed FMOLP algorithm is described next.

Algorithm 4: The proposed FMOLP algorithm.

Step 1: Formulate the fuzzy multi objective linear programming (FMOLP) model. The standard FMOLP model is as follows:

$$Maximise\ u = \tilde{c}_u x;\ Maximise\ v = \tilde{c}_v x;\ Maximise\ z = \tilde{c}_z x;\quad s.t.$$
$$Ax \le,\ or = or\ \ge b \quad and \quad x \ge 0.$$

Step 2: Determine the type of the fuzzy number to be chosen for each objective function coefficient $\tilde{c}$. For the triangular fuzzy numbers, the LP model with multiple objectives to minimize the value of fuzzy triangular numbers is:

$$\left.\begin{aligned} &\max\ u = (c_u^- x, c_u x, c_u^+ x) \\ &\max\ v = (c_v^- x, c_v x, c_v^+ x) \\ &\max\ z = (c_z^- x, c_z x, c_z^+ x) \end{aligned}\right\}\ s.t.\quad Ax \le,=,\ge b;\ x \ge 0.$$

Step 3: Transform the problem in Step 2 into the following:

$$\left.\begin{aligned} &\min\ u_1 = (c_u - c_u^-)x,\ \max u_2 = c_u x, \max\ u_3 = (c_u^+ - c_u)x \\ &\min\ v_1 = (c_v - c_v^-)x,\ \max v_2 = c_v x, \max\ v_3 = (c_v^+ - c_v)x \\ &\min\ z_1 = (c_z - c_z^-)x,\ \max z_2 = c_z x, \max\ z_3 = (c_z^+ - c_z)x \end{aligned}\right\}\ s.t.\ \ x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}.$$

Step 4: Determine the following values:

$$u_1^{\max} = \max_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} (c_u - c_u^-)x, \qquad u_1^{\min} = \min_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} (c_u - c_u^-)x$$
$$u_2^{\max} = \max_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} c_u x, \qquad u_2^{\min} = \min_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} c_u x$$
$$u_3^{\max} = \max_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} (c_u^+ - c_u)x, \qquad u_3^{\min} = \min_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} (c_u^+ - c_u)x$$

$$v_1^{\max} = \max_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} (c_v - c_v^-)x, \qquad v_1^{\min} = \min_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} (c_v - c_v^-)x$$
$$v_2^{\max} = \max_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} c_v x, \qquad v_2^{\min} = \min_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} c_v x$$
$$v_3^{\max} = \max_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} (c_v^+ - c_v)x, \qquad v_3^{\min} = \min_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} (c_v^+ - c_v)x$$

$$z_1^{\max} = \max_{x \in X = \{Ax \le,=,\ge b;\, x \ge 0\}} (c_z - c_z^-)x, \qquad z_1^{\min} = \min_{x \in X = \{Ax \le,=,\ge b;\, x \ge 0\}} (c_z - c_z^-)x$$

$$z_2^{\max} = \max_{x \in X = \{Ax \le,=,\ge b;\, x \ge 0\}} c_z x, \qquad z_2^{\min} = \min_{x \in X = \{Ax \le,=,\ge b;\, x \ge 0\}} c_z x$$

$$z_3^{\max} = \max_{x \in X = \{Ax \le,=,\ge b;\, x \ge 0\}} (c_z^+ - c_z)x, \qquad z_3^{\min} = \min_{x \in X = \{Ax \le,=,\ge b;\, x \ge 0\}} (c_z^+ - c_z)x.$$

Step 5: Define the following sets of membership functions (as examples):

$$\mu_{u_1}(x) = \begin{cases} 1, & if \quad (c_u - c_u^-)x \le u_1^{\min} \\ \dfrac{u_1^{\max} - (c_u - c_u^-)x}{u_1^{\max} - u_1^{\min}}, & if \quad u_1^{\min} \le (c_u - c_u^-)x \le u_1^{\max} \\ 0, & if \quad u_1^{\max} \le (c_u - c_u^-)x \end{cases}$$

$$\mu_{u_2}(x) = \begin{cases} 1, & if \quad c_u x \ge u_2^{\max} \\ \dfrac{c_u x - u_2^{\min}}{u_2^{\max} - u_2^{\min}}, & if \quad u_2^{\min} \le c_u x \le u_2^{\max} \\ 0, & if \quad u_2^{\min} \ge c_u x \end{cases}$$

$$\mu_{u_3}(x) = \begin{cases} 1, & if \quad (c_u^+ - c_u)x \ge u_3^{\max} \\ \dfrac{(c_u^+ - c_u)x - u_3^{\min}}{u_3^{\max} - u_3^{\min}}, & if \quad u_3^{\min} \le (c_u^+ - c_u)x \le u_3^{\max} \\ 0, & if \quad u_3^{\min} \ge (c_u^+ - c_u)x. \end{cases}$$

Step 6: Define the following LP problem:

$$\max_{x \in X = \{Ax \le,=,\ge b;\ x \ge 0\}} \{\alpha = \min\{\mu_{u_1}(x), \mu_{u_2}(x), \mu_{u_3}(x), \mu_{v_1}(x), \mu_{v_2}(x), \mu_{v_3}(x), \mu_{z_1}(x), \mu_{z_2}(x), \mu_{z_3}(x)\}\},$$

Or equivalently,

$$
\begin{aligned}
& Max\ \ \alpha \\
& s.t. \\
& \qquad (c_u - c_u^-)x + \alpha(u_1^{\max} - u_1^{\min}) \le u_1^{\max} \\
& \qquad c_u x - \alpha(u_2^{\max} - u_2^{\min}) \ge u_2^{\min} \\
& \qquad (c_u^+ - c_u)x - \alpha(u_3^{\max} - u_3^{\min}) \ge u_3^{\min} \\
& \\
& \qquad (c_v - c_v^-)x + \alpha(v_1^{\max} - v_1^{\min}) \le v_1^{\max} \\
& \qquad c_v x - \alpha(v_2^{\max} - v_2^{\min}) \ge v_2^{\min} \\
& \qquad (c_v^+ - c_v)x - \alpha(v_3^{\max} - v_3^{\min}) \ge v_3^{\min} \\
& \\
& \qquad (c_z - c_z^-)x + \alpha(z_1^{\max} - z_1^{\min}) \le z_1^{\max} \\
& \qquad c_z x - \alpha(z_2^{\max} - z_2^{\min}) \ge z_2^{\min} \\
& \qquad (c_z^+ - c_z)x - \alpha(z_3^{\max} - z_3^{\min}) \ge z_3^{\min} \\
& \\
& \qquad Ax \le,=,\ge b \\
& \qquad 0 \le \alpha \le 1 \\
& \qquad x \ge 0.
\end{aligned}
$$

Step 7: Solve the model in Step 6 to get an optimal solution x^*. Then, calculate the relative membership $\mu_{u_i}(x^*)$, $\mu_{v_i}(x^*)$ and $\mu_{z_i}(x^*)$, for $i = 1,2,3$.

Step8: Solve the following model to get an optimal solution x^{**}:

$$
\begin{aligned}
& Max \quad \alpha = \omega_1\alpha_{u_1} + \omega_2\alpha_{u_2} + \omega_3\alpha_{u_3} + \omega_4\alpha_{v_1} + \omega_5\alpha_{v_2} + \omega_6\alpha_{v_3} + \omega_7\alpha_{z} + \omega_8\alpha_{z_2} + \omega_9\alpha_{z_3} \\
& s.t. \\
& \qquad \mu_{u_i}(x^*) \le \alpha_{u_i} \le \mu_{u_i}(x), \quad i = 1,2,3, \\
& \qquad \mu_{v_i}(x^*) \le \alpha_{v_i} \le \mu_{v_i}(x), \quad i = 1,2,3, \\
& \qquad \mu_{z_i}(x^*) \le \alpha_{z_i} \le \mu_{z_i}(x), \quad i = 1,2,3, \\
& \qquad \sum_{i=1}^{9} \omega_i = 1\ ,\ \omega_i > 0,\ \forall i, \\
& \qquad Ax \le,=,\ge b\ \ ,\ x \ge 0.
\end{aligned}
$$

Step 9: Stop.

In Algorithm 4, steps 1 to 5 are employed from [41] and steps 6 to 8 from [42]. In [42], the authors proved the obtained solutions to be effective. Also, if all objective functions are not fuzzy, it is also possible to use Algorithm 4 to determine an optimal solution. For instance, if max $u = c_u x$ is not fuzzy, then it is enough to compute $u_2^{\min}$ and $u_2^{\max}$ in Step 4 and $\mu_{u_2}(x)$ in Step 5, and then follow the rest of the algorithm as before.

5 Experimental Study

Here, to illustrate the applicability and effectiveness of our proposed multiple optimization process, an experiment is worked out. Consider an undirected graph *G=(i, j)* with the cluster set $V = \{1,...,n\}$ and the link set $E = \{e = (i, j) : i, j \in V, i < j\}$, non-negative profits, p_e, associated with the links and non-negative costs, c_i, associated with the clusters. In this Steiner tree problem, the aim is to find the tree maximizing the revenue, i.e., the sum of the profits of the links in p_e spanned by the solution, and minimizing the sum of the costs of the clusters. On the one hand, we would like to have solution spanning all links avoiding the loss of profit; but this can be too expensive in terms of the cost of the tree-structured network providing service to all clusters. Thus, there is a trade-off between the cost of the clusters being in the solution and the profit of the links obtained by the solution.

The three dimensional fuzzy matrix of functions, sub-functions, and attributes are shown in Table 1. Note that the tables related to all the three attributes are configured and their arithmetic means are shown as the final functions, sub-functions, and attribute comparison matrix. In Table 2, $D_{2,\frac{1}{2}}(\overset{norm}{\tilde{F}_{ijk}}, MV)$ is shown for the comparison of fuzzy matrix for all the attributes and the minimum value is also reported as $MV = (0.27, 0.33, 0.4)$.

Our threshold value is considered to be 0.23 which is the mean of the data given in Table 2. Therefore, the thresholded matrix is shown in Table 3, and the corresponding network is configured as Fig. 7.

Table 1 The three dimensional comparison fuzzy matrix for all the attributes

Attributes	B_1	B_2	B_3	S_1	S_2	S_3
B_1	0	(0.53,0.59,0.65)	(0.48,0.58,0.68)	(0.51,0.6,0.96)	(0.42,0.53,0.6)	(0.55,0.66,0.75)
B_2	-	0	(0.35,0.46,0.57)	(0.29,0.36,0.45)	(0.38,0.46,0.58)	(0.6,0.63,0.72)
B_3	-	-	0	(0.48,0.59,0.7)	(0.43,0.56,0.65)	(0.44,0.53,0.61)
S_1	-	-	-	0	(0.5,0.58,0.64)	(0.49,0.59,0.67)
S_2	-	-	-	-	0	(0.27,0.36,0.41)
S_3	-	-	-	-	-	0
S_4	-	-	-	-	-	-
S_5	-	-	-	-	-	-
S_6	-	-	-	-	-	-
S_7	-	-	-	-	-	-

Continue of Table 1

Attributes	S_4	S_5	S_6	S_7
B_1	(0.37,0.46,0.57)	(0.45,0.52,0.6)	(0.4,0.48,0.57)	(0.39,0.46,0.57)
B_2	(0.75,0.86,0.94)	(0.51,0.58,0.68)	(0.37,0.43,0.52)	(0.37,0.43,0.49)
B_3	(0.81,0.90.97)	(0.55,0.63,0.71)	(0.27,0.33,0.4)	(0.47,0.53,0.6)
S_1	(0.49,0.6,0.72)	(0.45,0.53,0.62)	(0.52,0.6,0.71)	(0.5,0.6,0.67)
S_2	(0.35,0.43,0.54)	(0.37,0.43,0.5)	(0.62,0.7,0.78)	(0.77,0.83,0.93)
S_3	(0.550.63,0.74)	(0.48,0.6,0.74)	(0.55,0.63,0.72)	(0.5,0.58,0.69)
S_4	0	(0.4,0.46,0.52)	(0.49,0.6,0.7)	(0.33,0.43,0.52)
S_5	-	0	(0.58,0.65,0.72)	(0.53,0.63,0.7)
S_6	-	-	0	(0.56,0.63,0.76)
S_7	-	-	-	0

Table 2 The $D_{2,\frac{1}{2}}(\tilde{F}_{ijk}^{norm}, MV)$ comparison of fuzzy matrix for all the attributes.

Attributes	B_1	B_2	B_3	S_1	S_2	S_3	S_4	S_5	S_6	S_7
B_1	0	0.257	0.248	0.349	0.188	0.323	0.134	0.190	0.150	0.138
B_2	-	0	0.130	0.034	0.139	0.313	0.520	0.255	0.105	0.98
B_3	-	-	0	0.259	0.219	0.195	0.563	0.298	0	0.200
S_1	-	-	-	0	0.243	0.253	0.271	0.200	0.276	0.260
S_2	-	-	-	-	0	0.019	0.106	0.100	0.368	0.508
S_3	-	-	-	-	-	0	0.305	0.275	0.300	0.256
S_4	-	-	-	-	-	-	0	0.127	0.266	0.097
S_5	-	-	-	-	-	-	-	0	0.317	0.290
S_6	-	-	-	-	-	-	-	-	0	0.313
S_7	-	-	-	-	-	-	-	-	-	0

Table 3 The thresholded comparison fuzzy matrix for all the attributes

Attributes	B_1	B_2	B_3	S_1	S_2	S_3	S_4	S_5	S_6	S_7
B_1	0	1	1	1	0	1	0	0	0	0
B_2	-	0	0	0	0	1	1	1	0	0
B_3	-	-	0	1	0	0	1	1	0	0
S_1	-	-	-	0	1	1	1	0	1	1
S_2	-	-	-	-	0	0	0	0	1	1
S_3	-	-	-	-	-	0	1	1	1	1
S_4	-	-	-	-	-	-	0	0	1	0
S_5	-	-	-	-	-	-	-	0	1	1
S_6	-	-	-	-	-	-	-	-	0	1
S_7	-	-	-	-	-	-	-	-	-	0

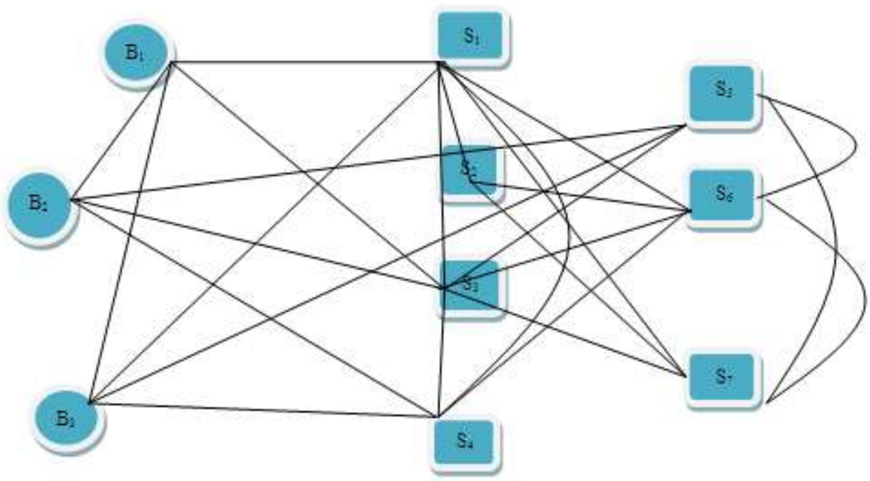

Fig. 7 The configured thresholded network

Then, the leveling process (the first and the second steps of Algorithm 3) is performed and the leveled network is configured as Fig. 8. The clustered network (the third and the forth steps of Algorithm 3) is shown in Fig. 9.

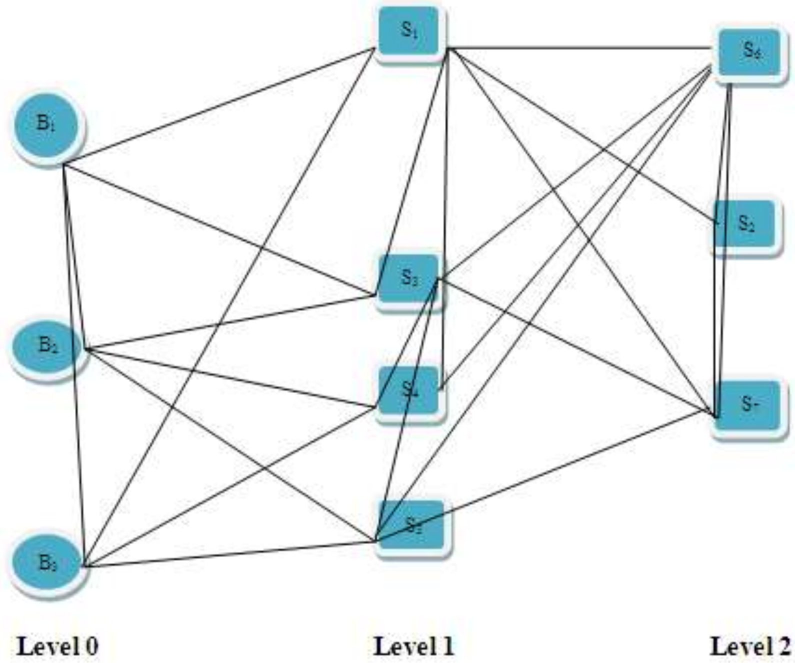

Fig. 8 The configured leveled network

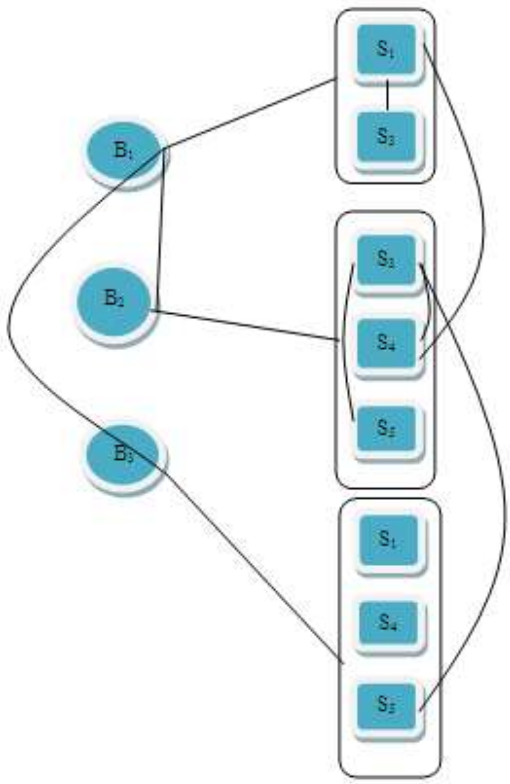

Fig. 9 The configured clustered network (the first step of Algorithm 2)

The cost vectors, the benefit matrix and the matrices $\alpha = [\alpha_{ij}]$, $\beta = [\beta_{ij}]$ are obtained to be:

C_1=(150, 210, 180, 20, 30, 30, 50, 10, 20, 20),
C_2=(300, 450, 600, 70, 80, 50, 40, 20, 20, 20).

$$\alpha = \begin{bmatrix} - & - & - \\ - & - & - \\ - & - & - \\ 0.65 & - & 0.7 \\ - & - & - \\ 1 & 0.8 & - \\ - & 0.6 & 0.9 \\ - & 0.7 & 0.65 \\ - & - & - \\ - & - & - \end{bmatrix} \qquad \beta = \begin{bmatrix} - & - & - \\ - & - & - \\ - & - & - \\ 0.9 & - & 0.9 \\ - & - & - \\ 0.9 & 0.7 & - \\ - & 0.6 & 0.9 \\ - & 0.7 & 0.6 \\ - & - & - \\ - & - & - \end{bmatrix}$$

$$p = \begin{bmatrix} - & 1500 & 1300 & 100 & - & 80 & - & - & - & - \\ - & - & - & - & - & 70 & 90 & 120 & - & - \\ - & - & - & 60 & - & - & 100 & 110 & - & - \\ - & - & - & - & 50 & 90 & 110 & - & 60 & 40 \\ - & - & - & - & - & - & - & - & 70 & 30 \\ - & - & - & - & - & - & 80 & 70 & 40 & 20 \\ - & - & - & - & - & - & - & - & 30 & - \\ - & - & - & - & - & - & - & - & 20 & 10 \\ - & - & - & - & - & - & - & - & - & 20 \\ - & - & - & - & - & - & - & - & - & - \end{bmatrix}.$$

$$f = \begin{bmatrix} - & (0.53,0.59,0.65) & (0.48,0.58,0.68) & (0.51,0.6,0.96) & - & (0.55,0.66,0.75) & - & - & - & - \\ - & - & - & - & - & (0.6,0.63,0.72) & (0.75,0.86,0.94) & (0.51,0.58,0.68) & - & - \\ - & - & - & (0.48,0.59,0.7) & - & - & (0.81,0.9,0.97) & (0.55,0.63,0.71) & - & - \\ - & - & - & - & (0.5,0.58,0.64) & (0.49,0.59,0.67) & (0.49,0.6,0.72) & - & (0.52,0.6,0.71) & (0.5,0.6,0.67) \\ - & - & - & - & - & - & - & - & (0.62,0.7,0.78) & (0.77,0.83,0.93) \\ - & - & - & - & - & - & (0.55,0.63,0.74) & (0.48,0.6,0.74) & (0.55,0.63,0.72) & (0.5,0.58,0.69) \\ - & - & - & - & - & - & - & - & (0.49,0.6,0.7) & - \\ - & - & - & - & - & - & - & - & (0.58,0.65,0.72) & (0.53,0.63,0.7) \\ - & - & - & - & - & - & - & - & - & (0.56,0.63,0.76) \\ - & - & - & - & - & - & - & - & - & - \end{bmatrix}$$

For level 1, using iteration 1 of the while loop in step 4 of Algorithm 3, we obtain:

$(\alpha_{41}c_{411}+\beta_{41}c_{412})=76$
$(\alpha_{43}c_{431}+\beta_{43}c_{432})=77$

$(\alpha_{61}c_{611}+\beta_{61}c_{612})=75$
$(\alpha_{62}c_{621}+\beta_{62}c_{622})=59$

$(\alpha_{72}c_{721}+\beta_{72}c_{722})=54$
$(\alpha_{73}c_{731}+\beta_{73}c_{732})=81$

$(\alpha_{82}c_{821}+\beta_{82}c_{822})=21$
$(\alpha_{83}c_{831}+\beta_{83}c_{832})=18.5.$

Therefore, $z_{41}=1, z_{62}=1, z_{72}=1$ and $z_{83}=1$ with other variables being zero. The configured network up to level 1 is shown in Fig. 10.

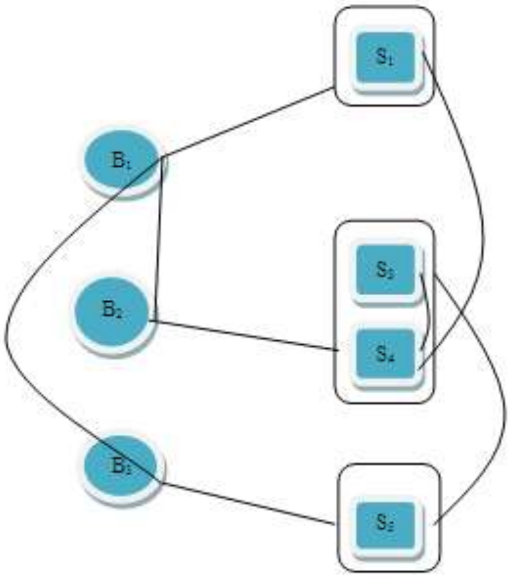

Fig. 10 The configured clustered network for level 1

In Fig. 11, the next iteration of Algorithm 3 for clustering is performed, and the purified network is obtained as in Fig. 12.

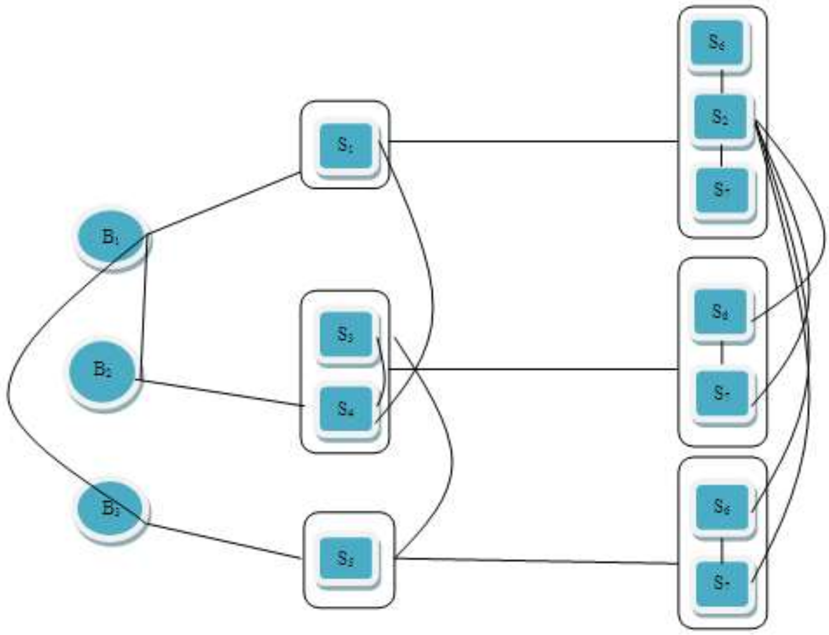

Fig. 11 The configured clustered network

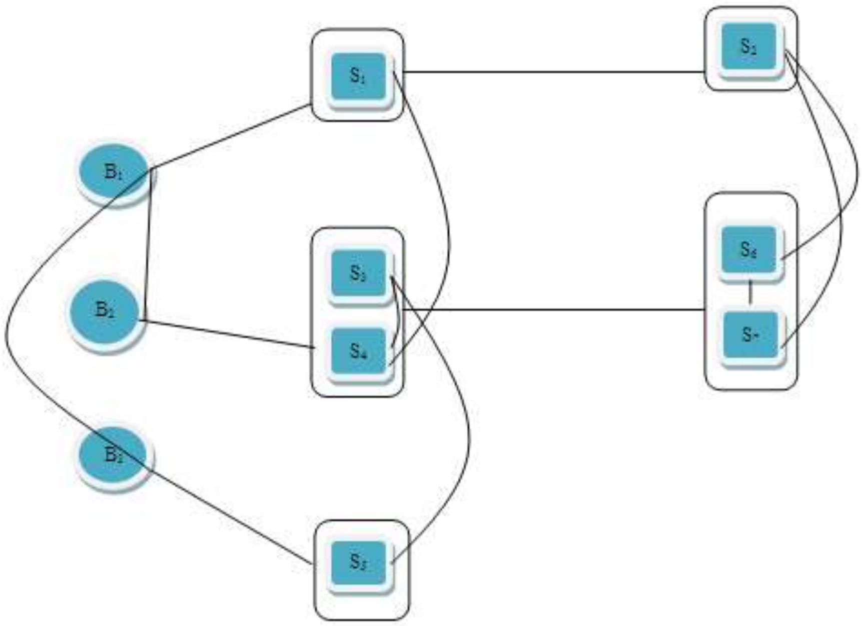

Fig. 12 The configured clustered network

After purifying the benefits, costs and customers total value for level 2, the final cost, benefit and customers total value matrices are formed as follows:

$$\tilde{p} = \begin{bmatrix} - & 1500 & 1300 & 100 & - & - & - & - \\ - & - & - & - & - & 208 & - & - \\ - & - & - & - & - & - & 110 & - \\ - & - & - & - & 50 & 132 & - & - \\ - & - & - & - & - & - & - & 130 \\ - & - & - & - & - & - & 84 & 135 \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{bmatrix}$$

$$\tilde{c} = (450 \quad 660 \quad 780 \quad 90 \quad 110 \quad 33 \quad 30 \quad 34)$$

$$\tilde{f} = \begin{bmatrix} - & (0.53,0.59,0.65) & (0.48,0.58,0.68) & (0.51,0.6,0.96) & - & - & - & - \\ - & - & - & - & - & (1.75,1.94,2.16) & - & - \\ - & - & - & - & - & - & (0.55,0.63,0.71) & - \\ - & - & - & - & (0.5,0.58,0.64) & (0.59,0.72,0.86) & - & - \\ - & - & - & - & - & - & - & (1.8,1.99,2.22) \\ - & - & - & - & - & - & (0.58,0.72,0.89) & (2.31,2.72,3.16) \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{bmatrix}$$

With respect to these matrices, the Steiner tree model is:

$$\max \quad u = 1500x_{12} + 1300x_{13} + 100x_{14} + 208x_{26} + 110x_{37} + 50x_{45} + 132x_{46} + 130x_{58} + 84x_{67} + 135x_{68}$$

$$\begin{aligned} \max \quad v = {} & (0.53,0.59,0.65)x_{12} + (0.48,0.58,0.68)x_{13} + (0.51,0.6,0.96)x_{14} + (1.75,1.94,2.16)x_{26} + \\ & (0.55,0.63,0.71)x_{37} + (0.5,0.58,0.64)x_{45} + (0.59,0.72,0.86)x_{46} + (1.8,1.99,2.22)x_{58} \\ & + (0.58,0.72,0.89)x_{67} + (2.31,2.72,3.16)x_{68} \end{aligned}$$

$$\min \quad z = 450y_1 + 660y_2 + 780y_3 + 90y_4 + 110y_5 + 33y_6 + 30y_7 + 34y_8$$

s.t.

$$x_{12} + x_{13} + x_{14} + x_{26} + x_{37} + x_{45} + x_{46} + x_{58} + x_{67} + x_{68} = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 - 1$$

$$x_{14} \le y_1$$

$$x_{14} \le y_4$$

$$x_{26} \le y_2$$

$$x_{26} \le y_6$$

$$x_{37} \le y_3$$

$$x_{37} \le y_7$$

$$x_{46} \le y_4$$

$$x_{46} \le y_6$$

$$x_{58} \le y_5$$
$$x_{58} \le y_8$$
$$x_{12} \le y_1$$
$$x_{12} \le y_2$$
$$x_{67} \le y_6$$
$$x_{67} \le y_7$$
$$x_{13} \le y_1$$
$$x_{13} \le y_3$$
$$x_{45} \le y_5$$
$$x_{45} \le y_4$$
$$x_{68} \le y_6$$
$$x_{68} \le y_8$$

$$x_{14} + x_{26} + x_{12} + x_{46} \le y_1 + y_2 + y_4$$
$$x_{14} + x_{26} + x_{12} + x_{46} \le y_1 + y_2 + y_6$$
$$x_{14} + x_{26} + x_{12} + x_{46} \le y_1 + y_6 + y_4$$
$$x_{14} + x_{26} + x_{12} + x_{46} \le y_6 + y_2 + y_4$$

$$x_{45} + x_{58} + x_{68} + x_{46} \le y_4 + y_5 + y_6$$
$$x_{45} + x_{58} + x_{68} + x_{46} \le y_4 + y_5 + y_8$$
$$x_{45} + x_{58} + x_{68} + x_{46} \le y_4 + y_8 + y_6$$
$$x_{45} + x_{58} + x_{68} + x_{46} \le y_8 + y_5 + y_6$$

$$x_{12} + x_{26} + x_{67} + x_{37} + x_{13} \le y_1 + y_2 + y_3 + y_6$$
$$x_{12} + x_{26} + x_{67} + x_{37} + x_{13} \le y_1 + y_2 + y_3 + y_7$$
$$x_{12} + x_{26} + x_{67} + x_{37} + x_{13} \le y_1 + y_2 + y_7 + y_6$$
$$x_{12} + x_{26} + x_{67} + x_{37} + x_{13} \le y_1 + y_7 + y_3 + y_6$$
$$x_{12} + x_{26} + x_{67} + x_{37} + x_{13} \le y_7 + y_2 + y_3 + y_6$$

$$x_{14} + x_{46} + x_{67} + x_{37} + x_{13} \le y_1 + y_4 + y_3 + y_6$$
$$x_{14} + x_{46} + x_{67} + x_{37} + x_{13} \le y_1 + y_4 + y_3 + y_7$$
$$x_{14} + x_{46} + x_{67} + x_{37} + x_{13} \le y_1 + y_4 + y_7 + y_6$$
$$x_{14} + x_{46} + x_{67} + x_{37} + x_{13} \le y_1 + y_7 + y_3 + y_6$$
$$x_{14} + x_{46} + x_{67} + x_{37} + x_{13} \le y_7 + y_4 + y_3 + y_6$$

$$x_{14}+x_{45}+x_{58}+x_{26}+x_{12}+x_{68}\le y_1+y_2+y_4+y_5+y_6$$
$$x_{14}+x_{45}+x_{58}+x_{26}+x_{12}+x_{68}\le y_1+y_2+y_4+y_5+y_8$$
$$x_{14}+x_{45}+x_{58}+x_{26}+x_{12}+x_{68}\le y_1+y_2+y_4+y_8+y_6$$
$$x_{14}+x_{45}+x_{58}+x_{26}+x_{12}+x_{68}\le y_1+y_2+y_8+y_5+y_6$$
$$x_{14}+x_{45}+x_{58}+x_{26}+x_{12}+x_{68}\le y_1+y_8+y_4+y_5+y_6$$
$$x_{14}+x_{45}+x_{58}+x_{26}+x_{12}+x_{68}\le y_8+y_2+y_4+y_5+y_6$$

$$x_{14}+x_{45}+x_{58}+x_{68}+x_{37}+x_{67}+x_{13}\le y_1+y_3+y_4+y_5+y_6+y_7$$
$$x_{14}+x_{45}+x_{58}+x_{68}+x_{37}+x_{67}+x_{13}\le y_1+y_3+y_4+y_5+y_6+y_8$$
$$x_{14}+x_{45}+x_{58}+x_{68}+x_{37}+x_{67}+x_{13}\le y_1+y_3+y_4+y_5+y_8+y_7$$
$$x_{14}+x_{45}+x_{58}+x_{68}+x_{37}+x_{67}+x_{13}\le y_1+y_3+y_4+y_8+y_6+y_7$$
$$x_{14}+x_{45}+x_{58}+x_{68}+x_{37}+x_{67}+x_{13}\le y_1+y_3+y_8+y_5+y_6+y_7$$
$$x_{14}+x_{45}+x_{58}+x_{68}+x_{37}+x_{67}+x_{13}\le y_1+y_8+y_4+y_5+y_6+y_7$$
$$x_{14}+x_{45}+x_{58}+x_{68}+x_{37}+x_{67}+x_{13}\le y_8+y_3+y_4+y_5+y_6+y_7$$

$$y_h=1, \qquad \forall h\in\{1,2,3\}$$
$$x_{ij}\in\{0,1\}, \qquad \forall i, j$$
$$y_i\in\{0,1\}, \qquad \forall i.$$

Now, we apply Algorithm 4 for obtaining Pareto optimal solution. After performing steps 2 and 3 of Algorithm 4, we obtain:

$$\max \quad u_2=1500x_{12}+1300x_{13}+100x_{14}+208x_{26}+110x_{37}+50x_{45}+132x_{46}+130x_{58}+84x_{67}+135x_{68}$$
$$x\in X,$$
$$\min \quad v_1=0.06x_{12}+0.1x_{13}+0.09x_{14}+0.19x_{26}+0.08x_{37}+0.08x_{45}+0.13x_{46}+0.19x_{58}+0.14x_{67}+0.41x_{68}$$
$$x\in X,$$
$$\max \quad v_2=0.59x_{12}+0.58x_{13}+0.6x_{14}+1.94x_{26}+0.63x_{37}+0.58x_{45}+0.72x_{46}+1.99x_{58}+0.72x_{67}+2.72x_{68}$$
$$x\in X,$$
$$\max \quad v_3=0.06x_{12}+0.1x_{13}+0.09x_{14}+0.22x_{26}+0.08x_{37}+0.06x_{45}+0.14x_{46}+0.23x_{58}+0.17x_{67}+0.44x_{68}$$
$$x\in X,$$
$$\max \quad z_2=-450y_1-660y_2-780y_3-90y_4-110y_5-33y_6-30y_7-34y_8$$
$$x\in X,$$

where, X is a feasible solution of the model. After solving the programming problems we obtain:

$$\mu_{u_2}(x)=\begin{cases}1, & if \quad u_2\ge 3515\\ \dfrac{u_2-634}{3515-634}, & if \quad 634\le u_2\le 3515\\ 0, & if \quad 634\ge u_2\end{cases}$$

$$\mu_{v_1}(x)=\begin{cases}1, & if \quad v_1 \le 0.24\\ \dfrac{1.25-v_1}{1.25-0.24}, & if \quad 0.24 \le v_1 \le 1.25\\ 0, & if \quad 1.25 \le v_1\end{cases}$$

$$\mu_{v_2}(x)=\begin{cases}1, & if \quad v_2 \ge 9.32\\ \dfrac{v_2-1.8}{9.32-1.8}, & if \quad 1.8 \le v_2 \le 9.32\\ 0, & if \quad 1.8 \ge v_2\end{cases}$$

$$\mu_{v_3}(x)=\begin{cases}1, & if \quad v_3 \ge 1.39\\ \dfrac{v_3-0.24}{1.39-0.24}, & if \quad 0.24 \le v_3 \le 1.39\\ 0, & if \quad 0.24 \ge v_3.\end{cases}$$

$$\mu_{z_2}(x)=\begin{cases}1, & if \quad z_2 \ge -1920\\ \dfrac{z_2+2187}{-1920+2187}, & if \quad -2187 \le z_2 \le -1920\\ 0, & if \quad -2187 \ge z_2\end{cases}$$

The LP problem in step 6 of Algorithm 4 is:

$$\begin{aligned}
&Max \ \ \alpha\\
&s.t.\\
&\qquad u_2-2881\alpha \ge 634\\
&\qquad v_1+1.01\alpha \le 1.25\\
&\qquad v_2-7.52\alpha \ge 1.8\\
&\qquad v_3-1.15\alpha \ge 0.24\\
&\qquad z_2-267\alpha \ge -2187\\
&\qquad 0 \le \alpha \le 1\\
&\qquad x \in X \ .
\end{aligned}$$

After solving this model, we get an optimal solution $\alpha^*=0.4851485$ and, $y_1=1, y_2=1, y_3=1, y_6=1, \ \ y_8=1, x_{12}=1, x_{13}=1, \ x_{26}=1, \ x_{68}=1$, with other variables being zero. At the final step, we solve the following model to obtain the Pareto optimal solutions:

$$
\begin{aligned}
&Max \quad \alpha = \omega_1\alpha_{u_2} + \omega_2\alpha_{v_1} + \omega_3\alpha_{v_2} + \omega_4\alpha_{v_3} + \omega_5\alpha_{z_2} \\
&s.t. \\
&\qquad 0.8708781 \le \alpha_{u_2} \le \mu_{u_2}(x) \\
&\qquad 0.4851485 \le \alpha_{v_1} \le \mu_{v_1}(x), \\
&\qquad 0.5359042 \le \alpha_{v_2} \le \mu_{v_2}(x), \\
&\qquad 0.5043478 \le \alpha_{v_3} \le \mu_{v_3}(x), \\
&\qquad 0.8614232 \le \alpha_{z_2} \le \mu_{z_2}(x) \\
&\qquad \sum_{i=1}^{5} \omega_i = 1 \ , \ \omega_i > 0, \ \forall i, \\
&\qquad x \in X .
\end{aligned}
$$

Given the compensation coefficients, ω_i, $\forall i$, we solve the resulting crisp model. If the decision maker is satisfied with the obtained current efficient compromise solution, then we stop. Otherwise, another efficient solution is provided by changing the value of some controllable parameters. For example, with $\varpi_1 = 0.3, \varpi_2 = 0.05, \varpi_3 = 0.3, \varpi_4 = 0.05, \varpi_5 = 0.3,$ after solving the model we get an optimal solution $\alpha^{**} = 0.7299365$ with $y_1 = 1, y_2 = 1, y_3 = 1, y_6 = 1, y_8 = 1, x_{12} = 1, x_{13} = 1, \quad x_{26} = 1, x_{68} = 1$ with the optimal network as shown in Fig. 13.

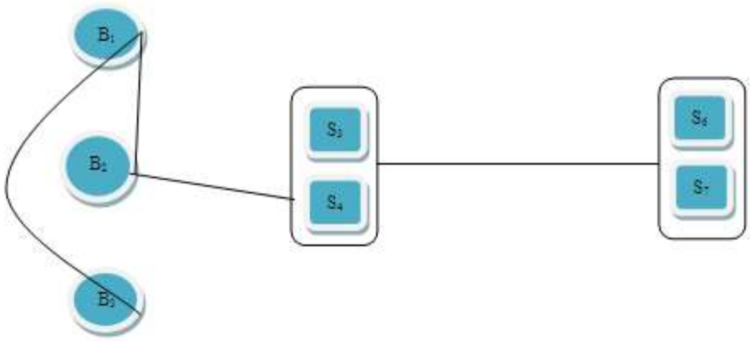

Fig. 13 The Pareto optimal solution

The proposed method provides different products for producers and consumers having different benefits, costs and customer's values. The numerical results imply the configuration of different products having various costs and customer's values being based on customers' views obtained from the web based system. The products themselves are the ones providing maximum benefits for the producers. The significant decision made in the proposed methodology is the trade off between the cost, benefit and customer's value objectives which is based on customers' views on adding features of products and producers views on configuration of beneficial features.

6 Conclusions

Considering convergent product as an important manufacturing technology for digital products, we integrated functions and sub-functions using a comprehensive fuzzy mathematical optimization process. To form the convergent product, a web-based fuzzy network was considered in which a collection of base functions and sub-functions configure the nodes and each arc in the network was to be a link between two nodes. The aim was to find an optimal tree of functionalities in the network adding value to the product in the web environment. First, a purification process was performed in the product network to assign the links among bases and sub-functions. Then, numerical values as benefits and costs were determined for the arc and the node, respectively. Also, a fuzzy customers' value corresponding to the arcs was considered. Next, the Steiner tree methodology was adapted to a multi-objective model of the network to find the optimal tree. A fuzzy multi-objective solution methodology was developed for solving the proposed problem. Finally, an example was worked out to illustrate the proposed approach.

Acknowledgements

The first two authors thank Mazandaran University of Science and Technology and the third author thanks Sharif University of Technology for supporting this work.

References

1. Johnson, D. S., (1985). The NP-completeness column: An ongoing guide. Journal of Algorithms, 6, 434-451.
2. Hakimi, S. B., (1971). Steiner's problem in graphs and its implications. Networks, 1, 113-133.
3. Lawler, E. L., (1976). Combinatorial Optimbation Networks and Matroids. New York, Holt, Rinehart and Winston.
4. Hwang, F. K., Richards, D. S., (1992). Steiner tree problems. Networks, 22, 55-89.
5. Dreyfuss, S. E., Wagner, R. A., (1971). The Steiner problem in graphs. Networks, 1, 195-207.
6. Beasley, J. E., (l989). An SST-based algorithm for the Steiner problem in graphs. Networks, 19, 1-16.
7. Koch, T., Martin, A., (1998). Solving Steiner tree problems in graphs to optimality. Networks, 32, 207-232.
8. Duin, C. W., Volgenant, A., (1989). Reduction tests for the Steiner problem in graphs. Networks, 19, 549-567.
9. Takahashi, H., Matsuyama, A., (1980). An approximate solution for the Steiner problem in graphs. Mathematica Japonica, 24(6), 573-577.
10. Kou, L., Markowsky, G., Berman, L., (1981). A fast algorithm for Steiner trees. Acta Informatica, 15, 141-145.
11. Rayward-Smith, V. J., Clare, A., (1986). On finding Steiner vertices. Networks, 16, 283-294.
12. Winter, P., MacGregor Smith, J., (1992). Path-distance heuristics for the Steiner problem in undirected networks. Algorithmica, 7, 309-327.
13. Mehlhorn, K., (1988). A faster approximation algorithm for the Steiner problem in graphs. Information Processing Letters Archive, 27, 125-128.
14. Robins, G., Zelikovsky, A., (2000). Improved Steiner tree approximation in graphs, in Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM, Philadelphia, ACM, New York, 770-779.
15. Robins, G., Zelikovsky, A., (2005). Tighter bounds for graph Steiner tree approximation. SIAM Journal on Discrete Mathematics, 19 (1), 122-134.
16. Esbensen, H., (1995). Computing near-optimal solutions to the Steiner problem in a graph using a genetic algorithm. Networks, 26, 173-185.
17. Kapsalis, A., Rayward-Smith, V. J., Smith, G. D., (1993). Solving the graphical Steiner tree problem using genetic algorithms. Journal of the Operational Research Society, 44(4), 397-406.

18. Martins, S. L., Pardalos, P., Resende, M. G., Ribeiro, C. C., (1999). Greedy randomized adaptive search procedures for the Steiner problem in graphs. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 43, 133-146.
19. Ribeiro, C. C., Souza, M. C., (2000). Tabu search for the Steiner problem in graphs. Networks, 36, 138-146.
20. Martins, S. L., Ribeiro, C. C., Souza, M. C., (1998). A parallel GRASP for the Steiner problem in graphs, Lecture Notes in Computer Science. Springer-Verlag, 1457, 310-331.
21. Martins, S. L., Resende, M. G. C., Ribeiro, C. C., Pardalos, P. M., (2000). A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy. Journal of Global Optimization, 17, 267-283.
22. Di Fatta, G., Lo Presti, G., Lo Re, G., (2003). A parallel genetic algorithm for the Steiner problem in networks. in proceeding of the 15th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2003), Marina del Rey, CA, USA, 569-573.
23. Bellman, R. E., Zadeh, L. A., (1970). Decision-making in a fuzzy environment. Management Science, 171, 41–164.
24. Buckley, J. J., (1987). The fuzzy mathematics of finance. Fuzzy Sets and Systems, 21, 257–273.
25. Kaufmann, A., Gupta, M. M., (1991). Introduction to Fuzzy Arithmetic: Theory and Applications. New York, Van Nostrand-Reinhold.
26. Mahdavi, I., Nourifar, R., Heidarzade, A., Mahdavi-Amiri, N., (2009). A dynamic programming approach for finding shortest chains in a fuzzy network. Applied Soft Computing, 9(2), 503–511.
27. Sadeghpour Gildeh, B., Gien, D., (2001). La Distance-Dp,q et le Cofficient de Corrélation entre deux Variables Aléatoires floues. Actes de LFA'2001, Monse-Belgium, 97-102.
28. Costa, A. M., Cordeau, J. F., Laporte, G., (2006). Steiner tree problems with profits. INFOR, 44(2), 99-115.
29. Zeleny, M., (1982). Multiple Criteria Decision Making. New York, McGraw-Hill Book Company.
30. Karsak, E. E., Kuzgunkaya, O., (2002). A fuzzy multiple objective programming approach for the selection of a flexible manufacturing system. International Journal of Production Economics, 79, 101–111.
31. Gao, Z., Tang, L., (2003). A multi-objective model for purchasing of bulk raw materials of a large-scale integrated steel plant. International Journal of Production Economics, 83, 325–334.
32. Downing, C. E., Ringuest, J. L., (1998). An experimental evaluation of the efficacy of four multi-objective linear programming algorithms. European Journal of Operational Research, 104, 549–558.
33. Liang, T. F., (2008). Interactive multi-objective transportation planning decisions using fuzzy linear programming. Asia-Pacific Journal of Operational Research, 25(1), 11–31.
34. Liang, T. F., (2006). Distribution planning decisions using interactive fuzzy multiobjective linear programming. Fuzzy Sets and Systems, 157(10), 1303–1316.
35. Petrovic-Lazarevic, S., Abraham, A., (2003). Hybrid fuzzy-linear programming approach for multi criteria decision making problems. International Journal of Neural, Parallel and Scientific Computations, 11, 53–68.
36. Dyson, R. G., (1980). Maxmin programming, fuzzy linear programming and multicriteria decision making. Journal of Operational Research Society, 31, 263–267.
37. Buckley, J. J., Feuring, T., Hayashi, Y., (2001). Multi-objective fully fuzzified linear programming. International Journal of Uncertainty, Fuzziness and Knowledge- Based Systems, 9(5), 605–621.
38. Chen, H. K., Chou, H. W., (1996). Solving multiobjective linear programming problems – a generic approach. Fuzzy Sets and Systems, 82, 35–38.
39. Guu, S. M., Wu, Y. K., (1997). Weighted coefficients in two-phase approach for solving the multiple objective programming problems. Fuzzy Sets and Systems, 85, 45–48.
40. Wu, Y. K., Guu, S. M., (2001). A compromise model for solving fuzzy multiple objective linear programming problems. Journal of the Chinese Institute of Industrial Engineers, 18(5), 87–93.
41. Susanto, S., Bhattacharya, A., (2011). Compromise fuzzy multi-objective linear programming (CFMOLP) heuristic for product-mix determination. Computers & Industrial Engineering, 61, 582–590.
42. Li, X. Q., Zhang, B., Li, H., (2006). Computing efficient solutions to fuzzy multiple objective linear programming problems. Fuzzy Sets and Systems, 157, 1328–1332.