Expert Systems with Applications 40 (2013) 892–898

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Solving fuzzy multiple objective generalized assignment problems directly via bees algorithm and fuzzy ranking

Pınar Tapkan [a,*], Lale Özbakır [a], Adil Baykasoğlu [b]

[a] Erciyes University, Faculty of Engineering, Department of Industrial Engineering, Kayseri, Turkey
[b] Dokuz Eylül University, Faculty of Engineering, Department of Industrial Engineering, İzmir, Turkey

ARTICLE INFO

Keywords:
Fuzzy generalized assignment problem
Bees algorithm
Fuzzy ranking
Compromise programming

ABSTRACT

In this paper, a direct solution approach for solving fuzzy multiple objective generalized assignment problems is proposed. In the problem, the coefficients and right hand side values of the constraints and the objective function coefficients are defined as fuzzy numbers. The addressed problem also has a multiple objective structure where the goals are determined so as to minimize the total cost and the imbalance between the workload of the agents. The direct solution approach utilizes the fuzzy ranking methods to rank the objective function values and to determine the feasibility of the constraints within a metaheuristic search algorithm, known as bees algorithm. Different fuzzy ranking methods, namely signed distance, integral value and area based approach are used in bees algorithm. For the computational study, the effects of these fuzzy ranking methods on the quality of the solutions are also analyzed.
© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Most of real life problems and models contain linguistic and/or imprecise variables and constraints. This can be due to different causes; usually, decision makers can state parameters on a system in terms of linguistic variables more easily and properly. Generally, collecting precise data is very difficult, because the environment of the system is unstable or collecting precise data requires high information costs. In addition, decision maker might not be able to express his/her goals or constraints precisely but rather in a fuzzy sense. For modeling systems which are imprecise by nature or which cannot be defined precisely, fuzzy mathematical programming that is based on the fuzzy set theory is generally employed.

In a mathematical programming problem, the fuzziness may appear in many different ways; the aspiration values of the objective(s), the limit values of resources (the right hand value of the constraints), the coefficients of the objective(s), and the coefficients of the constraints can be stated as fuzzy numbers. In this respect, fuzzy mathematical programming is suggested to solve problems which could be formulated as mathematical programming models, the parameter of which are fuzzy rather than crisp numbers (Zimmermann, 1983).

Fuzzy decision model as defined by Zimmermann (1976) is the origin of the fuzzy optimization, which is constituted on the symmetrical approach. However, if the other parameters of the model (objective coefficients, coefficients or right hand side values of constraints) are also fuzzy, the symmetrical approach cannot be used easily. Therefore, for fuzzy mathematical programming models with various fuzzy parameters, different optimization algorithms are proposed. There are various studies in which different fuzzy ranking procedures are used for the solution of fuzzy mathematical programming models in the literature (Baykasoğlu & Göçken, 2007; Cadenas & Verdegay, 2000; Campos & Verdegay, 1989; Fang, Hu, Wang, & Wu, 1999; Iskander, 2002; Jimenez, Arenas, Bilbao, & Rodriguez, 2007; Jimenez, Rodriguez, Arenas, & Bilbao, 2000; Nakahara, 1998; Tanaka, Hichihashi, & Asai, 1984). The main idea in most of the proposed approaches is to transform the fuzzy model into a crisp model, and to solve the crisp model by using a conventional method according to the form of the resultant model (linear or nonlinear). However, in transformation process some information can be lost or the number of constraints may increase (Baykasoğlu & Göçken, 2012).

On the other hand by using a direct solution method (DSM) a fuzzy mathematical programming problem can be solved without transformation into its crisp equivalent. In order to rank the objective function values and to determine the feasibility of the constraints, a ranking method for fuzzy numbers needs to be used and a meta-heuristic algorithm is generally utilized to carry out search to find an efficient solution in DSM (Baykasoğlu & Göçken, 2010, 2012).

In this study, a DSM is used to solve fuzzy multiple objective Generalized Assignment Problem (GAP) where the coefficients and right hand side values of the constraints and objective function coefficients are defined as fuzzy numbers. Swarm intelligence based Bees Algorithm (BA) and signed distance, integral value

* Corresponding author. Tel.: +90 352 2076666.
E-mail address: pinartan@erciyes.edu.tr (P. Tapkan).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.05.045

and area based fuzzy ranking methods are implemented in an integrated manner in the proposed DSM.

The remainder of this paper is structured as follows: Section 2 defines the basic GAP and Section 3 gives details about fuzzy ranking methods that are combined into DSM. Section 4 describes the proposed BA for solving fuzzy multiple objective GAP. Section 5 analyzes the effects of different fuzzy ranking methods on the quality of the solutions. Finally, paper concludes with Section 6.

2. Generalized assignment problem

GAP aims to assign a set of tasks to a set of agents with a minimum total cost where each agent represents a single resource with limited capacity, each task must be assigned to only one agent and each task requires a certain amount of the resource of the agent. There are many application areas of GAP, such as computer and communication networks, location problems, vehicle routing, group technology and scheduling. Since GAP has an NP-hard structure (Fisher, Jaikumar, & Van Wassenhove, 1986), several exact algorithms (Nauss, 2003; Savelsberg, 1997) and heuristic algorithms (Alfandari, Plateau, & Tolla, 2004; Diaz & Fernandez, 2001; Feltl & Raidl, 2004; Yagiura, Ibaraki, & Glover, 2004; Yagiura, Ibaraki, & Glover, 2006; Özbakır, Baykasoğlu, & Tapkan, 2010) have been proposed to solve GAP in recent years. An extensive literature survey on GAP and its possible applications are presented by Öncan (2007).

The non-deterministic case of GAP is studied rarely compared with the deterministic case. However, in real life there may be several uncertainties, for example parameters may change during the assignment process, or historical data to estimate the values of these parameters may be unavailable. Toktas, Yen, and Zabinsky (2006) proposed some heuristics based on approximation models for GAP with stochastic capacities. Majumdar and Bhunia (2007) developed a solution procedure using elitist genetic algorithm for GAP with imprecise cost where impreciseness of cost has been represented by interval valued numbers.

Mingbiao, Xinmeng, Feng, and Zhijian (2007) introduce the idea of Grid Resource Supermarket to organize and manage the grid resources in virtual organization, present a posted-price model based on Grid Resource Supermarket and optimize the allocation combining the fuzzy theory with Hungary algorithm. Bai, Liu, and Shen (2009) solve the fuzzy generalized assignment problem with credibility constraints where the cost and the resource amount consumed are uncertain and characterized to be fuzzy variables with known possibility distributions. Bai, Zhang, and Liu (2010) solve the fuzzy generalized assignment problem, where the resource amounts consumed are uncertain and assumed to be characterized by fuzzy variables with known possibility distributions, by a hybrid algorithm integrating the approximation approach and Particle Swarm Optimization. Based on this consideration, the fuzzy set theory is employed to model the uncertainty in the GAP in this paper. In this manner, the mathematical programming model for the fuzzy multiple objective GAP is formulated as follows:

$$
\begin{aligned}
&\tilde{z}_1 = \min \sum_{i=1}^{n}\sum_{j=1}^{m}\tilde{c}_{ij}x_{ij}\\
&\tilde{z}_2 = \min \sum_{j=1}^{m}\left(\tilde{r} - \sum_{i=1}^{n}\tilde{a}_{ij}x_{ij}\right)\\
&s.t.\\
&\tilde{r} \geqslant \sum_{i=1}^{n}\tilde{a}_{ij}x_{ij} \quad \forall j\\
&\sum_{i=1}^{n}\tilde{a}_{ij}x_{ij}\tilde{b}_j \quad 1 \leqslant j \leqslant m \quad \forall j\\
&\sum_{j=1}^{m}x_{ij} = 1 \quad 1 \leqslant i \leqslant n \quad \forall i\\
&x_{ij} \in \{0,1\} \quad 1 \leqslant i \leqslant n \quad \forall i, 1 \leqslant j \leqslant m \quad \forall j
\end{aligned}
\tag{1}
$$

where I is the set of tasks $(i = 1,\ldots,n)$; J is the set of agents $(j = 1,\ldots,m)$; $\tilde{b}_j$ is the resource capacity of agent j; $\tilde{a}_{ij}$ is the resource required if task i is assigned to agent j; $\tilde{c}_{ij}$ is the cost of task i if assigned to agent j. The decision variables of the mathematical programming model are x_{ij} and $\tilde{r}$ where x_{ij} takes the value of 1 if task i is assigned to agent j and 0 otherwise and $\tilde{r}$ represents the maximum workload of agents. The second constraint set is related to the resource capacity of agents and the third constraint set ensures that each task is assigned to only one agent. As can be seen from the model the agent capacities, resource requirements and costs are considered as fuzzy numbers. Based on the extensive classification of fuzzy mathematical programs which was proposed by Baykasoğlu and Göçken (2008) the present model is a type 15 fuzzy model (fully fuzzy model).

3. Ranking of fuzzy numbers

Since the study of fuzzy ranking began, various ranking methods have been proposed, but no best method has been agreed upon. The proposed ranking methods are based on extracting various features from fuzzy sets (numbers) and consequently all the proposed ranking methods have both advantages and disadvantages. Because different ranking methods order fuzzy sets according to different features, normally the obtained ranking order for the same sample of fuzzy sets (numbers) can be different (Prodanovic, 2001).

In addition, in the literature, the fuzzy parameters are generally defined as triangular or trapezoidal fuzzy numbers, since the employment of ranking methods to triangular or trapezoidal fuzzy numbers is simple. Conversely, if the fuzzy numbers are defined in different shapes, it is not easy to use ranking methods in transformation process mathematically. In the present paper, the fuzzy parameters are defined as triangular fuzzy numbers. A fuzzy number $\widetilde{A}$ is called a triangular fuzzy number $\widetilde{A} = (a_1, a_2, a_3), a_1 \leqslant a_2 \leqslant a_3$ if its membership function $\mu_A(x): R \to [0,1]$ is equal to

$$
\mu_A(x) = \begin{cases} \left(\frac{x-a_1}{a_2-a_1}\right) & x \in (a_1, a_2] \\ \left(\frac{a_3-x}{a_3-a_2}\right) & x \in [a_2, a_3) \\ 0 & others \end{cases} \tag{2}
$$

In this study, three ranking methods are selected according to the ease of computation, and their accomplishment and consistency in ranking of fuzzy numbers. The ranking methods combined into the proposed DSM are clarified as follows:

3.1. Integral value method (Liou & Wang, 1992)

Some ranking methods assume that the membership functions are normal. In the method of ranking fuzzy numbers with integral value, the assumption of normality of membership functions is not required. Ranking fuzzy numbers with integral value is relatively simple in computation, especially in ranking of triangular and trapezoidal fuzzy numbers, and can be used to rank more than two fuzzy numbers simultaneously. The definition of integral values for the triangular fuzzy number $\widetilde{A}$ is written as follows

$$
\begin{aligned}
I(\widetilde{A}) &= (1-\alpha)\int_0^1 g^L_{\widetilde{A}}(u)du + \alpha\int_0^1 g^R_{\widetilde{A}}(u)du \\
&= \frac{1-\alpha}{2}a_1 + \frac{1}{2}a_2 + \frac{\alpha}{2}a_3
\end{aligned}
\tag{3}
$$

where $0 \leqslant \alpha \leqslant 1$. The index of optimism α represents the degree of optimism for a person, where a larger α indicates a higher degree of optimism. The fuzzy numbers are ranked according to their integral values; the fuzzy number with the larger integral value is the bigger fuzzy number. In this research, the performance of different values of α (0.6, 0.7, 0.8) is also analyzed.

3.2. The signed distance method (Yao & Wu, 2000)

The signed distance used for fuzzy numbers has similar properties to those induced by the signed distance in real numbers. The signed distance of a triangular fuzzy number $\widetilde{A}$ is defined as:

$$d(\widetilde{A},0) = \frac{1}{2}\int_0^1 [a_1 + (a_2 - a_1)\alpha + a_3 - (a_3 - a_2)\alpha]d\alpha$$
$$= \frac{1}{4}(2a_2 + a_1 + a_3) \tag{4}$$

Let $\widetilde{A}$ and $\widetilde{B}$ are two triangular fuzzy numbers, their ranking relation is defined as:

$$\widetilde{A} \leqslant \widetilde{B} \leftrightarrow d(\widetilde{A},0) \leqslant d(\widetilde{B},0) \tag{5}$$

3.3. The area-based ranking method (Kahraman & Tolga, 2009)

This method has some advantages with respect to the other methods in both graphical representations and calculations. An index that measures the possibility of one fuzzy number being greater than another is determined ($I(\omega)$). This index can be determined as in the following standard form:

$$I(\omega) = \begin{cases} 0 & b_1 \geqslant a_3 \\ \dfrac{\frac{(a_3-b_1)^2}{(b_2-b_1-a_2+a_3)}}{(a_3-a_1)+(b_3-b_1)} & b_2 \geqslant a_2, b_1 < a_3 \\ \dfrac{(a_3+a_2-b_2-b_1)-\frac{(a_2-b_2)^2}{(b_3-b_2+a_2-a_1)}}{(a_3-a_1)+(b_3-b_1)} & b_2 < a_2, b_3 > a_1 \\ 1 & b_3 \geqslant a_1 \end{cases} \tag{6}$$

And the fuzzy preference relation (P_{KT}) of the fuzzy numbers is determined as following:

$$P_{KT}(\widetilde{A},\widetilde{B}) = \begin{cases} \widetilde{A} > \widetilde{B} & \text{if } \mathrm{I}(\omega) \in (0.5, 1] \\ \widetilde{A} = \widetilde{B} & \text{if } \mathrm{I}(\omega) = 0.5 \\ \widetilde{A} < \widetilde{B} & \text{if } \mathrm{I}(\omega) \in [0, 0.5) \end{cases} \tag{7}$$

4. Bees algorithm for fuzzy multiple objective generalized assignment problem

BA is a general purpose swarm based metaheuristic optimization algorithm, inspired by the foraging behavior of honey bees. In real life, communication among bees about the quality of food sources is achieved in the dancing area by performing the waggle dance. By performing this dance, successful foragers share the information about direction and distance to patches of flowers and the amount of nectar within this flower with their hive mates. This is a successful mechanism which foragers can recruit other bees in their colony to productive locations to collect various resources (Pham et al., 2006). The detailed pseudo code of BA for fuzzy multiple objective GAP is presented in Table 1 with the following notation.

S is the number of scout bees ($s = 1,\ldots,S$); P is the number of employed bees ($p = 1,\ldots,P$); e is the number of best employed bees; *nep* is the number of onlooker bees for each e employed bees; *nsp* is the number of onlooker bees for each *P-e* employed bees (*nsp* < *nep*); σ^s is the solution of sth scout bee; σ^{ls} is the A neighbor solution found by local search, ls = {*shift, doubleshift, ejection chain*}; σ^{best} is the best solution; $fit(\sigma^s)$ is the fitness function value of sth scout bee solution and *MaxIter* is the iteration limit (stopping criteria).

The algorithm starts with parameter initialization (*S, P, e, nep, nsp, EC_Length, MaxIter*) and continues by generating S number of initial solutions by scout bees. P number of good solutions within the set of generated solutions is determined as employed bees. e number of solutions is selected from the set P as the best solutions. *nep* number of onlooker bees are directed to these best solutions in order to carry out a more detailed neighborhood search. Fewer onlooker bees are directed to the remaining *P-e* solutions. Shift, double shift and ejection chain neighborhood mechanisms are applied to each employed bee respectively for local search. The employed bee solutions are compared with the best solution, if it is better than the previous one, the best solution is updated. For global search, *S-P* number of scout bee solution is generated and, in this way, at the end of each iteration, new population is constructed by the representatives of each chosen site and scout bees searching randomly. The important sub-steps of BA for fuzzy multiple objective GAP are detailed at the following sub-sections.

4.1. Bee colony initialization

The initial bee colony is constructed by using the following heuristic procedure.

Find the task and agent that have the minimum cost assignment (i,j).
Check whether there is another assignment of task i with the same minimum cost.
Select an agent from this set randomly (j').
If the assignment is feasible according to capacity constraint, then task i is assigned to agent j', otherwise try the other agents that have minimum assignment cost.
Until all the tasks are assigned.

4.2. Fitness function

In order to handle the fuzzy goals fuzzy compromise programming is used in the present study. Classical compromise programming (Zeleny, 1973) is a mathematical programming technique that ranks a discrete set of alternatives according to their distance from an ideal solution. The closeness of each alternative to the ideal solution is determined by some measure of distance, referred to as a distance metric. Once all distance metrics are obtained, they are sorted from the smallest to largest, where the smallest distance metric value determines the best compromise alternative. However, fuzzy compromise programming (Bender & Simonovic, 1996; Bender & Simonovic, 2000) uses fuzzy numbers in the compromise programming distance metric equation instead of crisp numbers, applies fuzzy arithmetic instead of classical arithmetic and fuzzy set ranking methods instead of simply sorting distance metrics.

The fuzzified compromise programming equation used to compute fuzzy distance metric values is given by

$$\widetilde{L}_j = \left\{ \sum_{i=1}^{n} \left[w_i^p \left(\frac{\tilde{f}_i^* - \tilde{f}_i}{\tilde{f}_i^* - \tilde{f}_i^-} \right)^p \right] \right\}^{1/p} \tag{8}$$

where i represents objectives; j represents alternatives ($j = 1,\ldots,m$); L_j is the distance metric of alternative j; w_i corresponds to a weight of a particular objective; $\tilde{f}_i^*$ and $\tilde{f}_i^-$ are the best and the worst values, respectively, for objective i under all alternatives; and $\tilde{f}_i$ is the actual value of objective i under alternative j. The parameter p ($p = 1,2,\ldots,\infty$) is used to represent the importance of the maximal deviation from the ideal point. If $p = 1$, all deviations are weighted equally; and if $p = 2$, the deviations are weighted in proportion to their magnitude. Typically, as p increases, so also does the weighting of the deviations (Prodanovic & Simonovic, 2002).

Table 1
The pseudo code of BA for fuzzy multiple objective GAP.

```
1. Parameter initialization
2. Initialize S number of scout bees
3. Evaluate scout bees' fitness function
   min fit(σ^s) = { Σ_{i=1}^{2} [ w_i^p ( (f̃_i^* − f̃_i) / (f̃_i^* − f̃_i^−) )^p ] }^{1/p}
4. I = 0
5. Do
   Sort_{s=1...S} μ_cost(σ^s) in decreasing order and determine best P solutions as employed bees
   Select best e employed bees
   Assign nep number of onlooker bees to each of best e employed bees
   Assign nsp number of onlooker bees to each of remaining P-e employed bees
   k = 0
   Do
     t = 0
     Do
       Shift
         If μ_cost(σ^shift) > μ_cost(σ^p) then σ^p = σ^shift
         If μ_cost(σ^shift) = μ_cost(σ^p) and f(σ^shift) < f(σ^p) then σ^p = σ^shift
       Double shift
         If μ_cost(σ^doubleshift) > μ_cost(σ^p) then σ^p = σ^doubleshift
         If μ_cost(σ^doubleshift) = μ_cost(σ^p) and f(σ^doubleshift) < f(σ^p) then σ^p = σ^doubleshift
       Ejection chain
         If μ_cost(σ^ejectionchain) > μ_cost(σ^p) then σ^p = σ^ejectionchain
         If μ_cost(σ^ejectionchain) = μ_cost(σ^p) and f(σ^ejectionchain) < f(σ^p) then σ^p = σ^ejectionchain
     t = t + 1
     While (t < nep)
     t = 0
     Do
       Shift
         If fit(σ^shift) > μ_D(σ^p) then σ^p = σ^shift
         If μ_cost(σ^shift) = μ_cost(σ^p) and f(σ^shift) < f(σ^p) then σ^p = σ^shift
       Double shift
         If fit(σ^doubleshift) > μ_D(σ^p) then σ^p = σ^doubleshift
         If μ_cost(σ^doubleshift) = μ_cost(σ^p) and f(σ^doubleshift) < f(σ^p) then σ^p = σ^doubleshift
       Ejection chain
         If fit(σ^ejectionchain) > μ_D(σ^p) then σ^p = σ^ejectionchain
         If μ_cost(σ^ejectionchain) = μ_cost(σ^p) and f(σ^ejectionchain) < f(σ^p) then σ^p = σ^ejectionchain
     t = t + 1
     While (t < nsp)
   k = k + 1
   While (k < P)
   Update the best solution
     If max_{p=1,...,P} μ_cost(σ^p) > μ_cost(σ^best) then σ^best = σ^p
     If max_{p=1,...,P} μ_cost(σ^p) = μ_cost(σ^best) and f(σ^p) < f(σ^best) then σ^best = σ^p
   Generate S-P number of scout bees
   I = I + 1
While (I = MaxIter)
```

The objective function of the fuzzy multiple objective GAP is stated as a single objective as follows:

$$\widetilde{L} = \left\{ w_{\cos t}^{p} \left(\frac{\tilde{f}_{\cos t}^{*} - \tilde{f}_{\cos t}}{\tilde{f}_{\cos t}^{*} - \tilde{f}_{\cos t}^{-}} \right)^{p} + w_{workload}^{p} \left(\frac{\tilde{f}_{workload}^{*} - \tilde{f}_{workload}}{\tilde{f}_{workload}^{*} - \tilde{f}_{workload}^{-}} \right)^{p} \right\}^{1/p} \tag{9}$$

As mentioned above, classical arithmetic operations cannot be used in fuzzy compromise programming. On the other hand, since the results of multiplication or division are not triangular fuzzy numbers, fuzzy interval arithmetic is utilized for all operations to obtain the approximate values of the result of the multiplication and division operations. The crisp interval of a triangular fuzzy number is obtained by α-cut operation which is calculated as follows:

$$A_{\alpha} = [a_1^{(\alpha)}, a_3^{(\alpha)}] = [(a_2 - a_1)\alpha + a_1, -(a_3 - a_2)\alpha + a_3] \tag{10}$$

Let $\widetilde{A}$ and $\widetilde{B}$ are two triangular fuzzy numbers, the fuzzy arithmetic operations including addition, subtraction, multiplication and division (excluding the case $b_1^{(\alpha)} = 0$ or $b_3^{(\alpha)} = 0$) are defined as follows (Lee, 2005):

$$\widetilde{A}(+)\widetilde{B} = [a_1^{(\alpha)}, a_3^{(\alpha)}](+)[b_1^{(\alpha)}, b_3^{(\alpha)}] = [a_1^{(\alpha)} + b_1^{(\alpha)}, a_3^{(\alpha)} + b_3^{(\alpha)}] \tag{11}$$

$$\widetilde{A}(-)\widetilde{B} = [a_1^{(\alpha)}, a_3^{(\alpha)}](-)[b_1^{(\alpha)}, b_3^{(\alpha)}] = [a_1^{(\alpha)} - b_3^{(\alpha)}, a_3^{(\alpha)} - b_1^{(\alpha)}] \tag{12}$$

$$\begin{aligned} \widetilde{A}(*)\widetilde{B} &= [a_1^{(\alpha)}, a_3^{(\alpha)}](*)[b_1^{(\alpha)}, b_3^{(\alpha)}] \\ &= [a_1^{(\alpha)*} b_1^{(\alpha)} \wedge a_1^{(\alpha)*} b_3^{(\alpha)} \wedge a_3^{(\alpha)*} b_1^{(\alpha)} \wedge a_3^{(\alpha)*} b_3^{(\alpha)}, a_1^{(\alpha)*} b_1^{(\alpha)} \\ &\quad \vee a_1^{(\alpha)*} b_3^{(\alpha)} \vee a_3^{(\alpha)*} b_1^{(\alpha)} \vee a_3^{(\alpha)*} b_3^{(\alpha)}] \end{aligned} \tag{13}$$

$$\begin{aligned} \widetilde{A}(/)\widetilde{B} &= [a_1^{(\alpha)}, a_3^{(\alpha)}](/)[b_1^{(\alpha)}, b_3^{(\alpha)}] \\ &= [a_1^{(\alpha)}/b_1^{(\alpha)} \wedge a_1^{(\alpha)}/b_3^{(\alpha)} \wedge a_3^{(\alpha)}/b_1^{(\alpha)} \wedge a_3^{(\alpha)}/b_3^{(\alpha)}, a_1^{(\alpha)}/b_1^{(\alpha)} \\ &\quad \vee a_1^{(\alpha)}/b_3^{(\alpha)} \vee a_3^{(\alpha)}/b_1^{(\alpha)} \vee a_3^{(\alpha)}/b_3^{(\alpha)}] \end{aligned} \tag{14}$$

4.3. Neighborhood structures

The neighborhood structures incorporated into the algorithm are briefly explained as following:

Shift: This type of neighbor solution is obtained from the original solution by changing the agent assignment of one task.
Double Shift: This neighborhood structure contains two consecutive shift moves.

Ejection Chain: A neighbor solution is obtained by performing the multiple shift moves whose length is specified as chain length (*EC_Length*). This neighborhood structure is more powerful but more complicated than the other ones.

Readers may refer to Özbakır et al. (2010) for detailed information about these neighborhood structures.

5. Computational study

The BA for fuzzy multiple objective GAP is coded in C# and tested on benchmark problems on an Intel Core 2 Duo PC with 3.20 GHz CPU and 2.00 GB RAM. The benchmark problems contain gap-a and gap-b problems that range from 5 agents – 100 tasks to 20 agents – 200 tasks. These test problems are obtained from the OR-Library (http://mscmga.ms.ic.ac.uk/jeb/orlib/gapinfo.html). In order to determine the values of parameters, six different parameter combinations are constructed as $\{S, P, e, nep, nsp\}$: {100, 40, 20, 10, 5}, {100, 40, 20, 5, 2},{50, 20, 10, 10, 5}, {50,20, 10,5,2}, {25,15, 10,10,5} and {25,15,10,5,2}. After the evaluation of different parameter combinations, the one which has the best performance is determined for the parameter setting of the algorithm. The parameters of the proposed BA are defined as in Table 2.

The *EC_Length* parameter does not have a constant value and is changed at the onward stages of the search based on the current iteration number by Eq. (15). This structure provides a global search in the early stages of algorithm while satisfying a more focused search in the latest iterations.

$$(EC\_Length_{final} - EC\_Length_{initial})\frac{Current\ iteration\ number}{MaxIter} + EC\_Length_{initial} \tag{15}$$

where initial value of ejection chain length ($EC\_Length_{initial}$) is 50 and the final value of ejection chain length (EC_Length_{final}) is 5.

The cost objective is assumed to be more important than the workload objective and hence the weight values are determined as $w_1 = 5, w_2 = 1$. It is also assumed that all deviations are weighted equally and so the parameter p is selected as 1. Additionally, the best value of the cost objective ($\tilde{f}^*_{\cos t}$) is determined to be a triangular fuzzy number obtained from the best known crisp solution, where the best value of the workload objective ($\tilde{f}^*_{workload}$) is determined as 0. On the other hand, the worst values of the objectives are obtained from the first generated bee solution. The fitness function of the direct solution approach for fuzzy multiple objective GAP is stated as follows:

$$\tilde{L} = 5\left(\frac{\tilde{f}^*_{\cos t} - \tilde{f}_{\cos t}}{\tilde{f}^*_{\cos t} - \tilde{f}^-_{\cos t}}\right) + \left(\frac{\tilde{f}^*_{workload} - \tilde{f}_{workload}}{\tilde{f}^*_{workload} - \tilde{f}^-_{workload}}\right) \tag{16}$$

The proposed algorithm is reconstructed for each ranking method and the performance of the five different versions of the algorithm is analyzed for each problem type within 10 runs. At the evaluation stage of the computational study results Fuzzy TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) proposed by Olson (2004) is utilized within two phases.

Table 2
Parameter setting.

Parameters	Value
S	25
P	15
e	10
nep	10
nsp	5
MaxIter	500

Table 3
The computational study results according to the cost objective.

Problem type	Integral value method ($\alpha = 0.6$)		Integral value method ($\alpha = 0.7$)		Integral value method ($\alpha = 0.8$)		Area-based method		Signed distance method	
	Cost objective	CPU	Cost objective	CPU	Cost objective	CPU	Cost objective	CPU	Cost objective	CPU
gap-a1	(2088.9,2321,2553.1)	135.918	(2186.1,2429,2671.9)	133.055	(2223.9,2471,2718.1)	106.750	(2082.6,2314,2545.4)	131.843	(2037.6,2264,2490.4)	101.009
gap-a2	(4120.2,4578,5035.8)	213.689	(4604.4,5116,5627.6)	245.023	(4309.2,4788,5266.8)	174.880	(4158,4620,5082)	226.889	(3961.8,4402,4842.2)	190.032
gap-a3	(1650.6,1834,2017.4)	162.288	(1768.5,1965,2161.5)	168.129	(1637.1,1819,2000.9)	142.834	(1710,1900,2090)	121.666	(1704.6,1894,2083.4)	142.826
gap-a4	(3412.8,3792,4171.2)	257.950	(3468.6,3854,4239.4)	291.027	(3396.6,3774,4151.4)	134.146	(3798,4220,4642)	191.693	(3360.6,3734,4107.4)	327.125
gap-a5	(1515.6,1684,1852.4)	208.512	(1550.7,1723,1895.3)	102.296	(1496.7,1663,1829.3)	136.378	(1496.7,1663,1829.3)	194.267	(1506.6,1674,1841.4)	158.630
gap-a6	(3132.9,3481,3829.1)	361.798	(3137.4,3486,3834.6)	357.030	(3098.7,3443,3787.3)	357.281	(3073.5,3415,3756.5)	331.718	(3305.7,3673,4040.3)	318.000
gap-b1	(1750.5,1945,2139.5)	99.262	(1726.2,1918,2109.8)	58.140	(1755,1950,2145)	69.437	(1864.8,2072,2279.2)	20.701	(1922.4,2136,2349.6)	36.344
gap-b2	(4119.3,4577,5034.7)	156.033	(4259.7,4733,5206.3)	195.231	(4374.9,4861,5347.1)	52.048	(4023,4470,4917)	146.257	(4005,4450,4895)	377.056
gap-b3	(1520.1,1689,1857.9)	45.212	(1589.4,1766,1942.6)	38.251	(1579.5,1755,1930.5)	91.903	(1530,1700,1870)	75.985	(1574.1,1749,1923.9)	152.655
gap-b4	(3366,3740,4114)	436.180	(3933,4370,4807)	93.359	(3687.3,4097,4506.7)	321.765	(3503.7,3893,4282.3)	324.934	(3460.5,3845,4229.5)	204.609
gap-b5	(1468.8,1632,1795.2)	44.566	(1399.5,1555,1710.5)	58.615	(1464.3,1627,1789.7)	23.828	(1350,1500,1650)	735.765	(1381.5,1535,1688.5)	122.210
gap-b6	(3241.8,3602,3962.2)	175.163	(3419.1,3799,4178.9)	123.287	(3393,3770,4147)	324.671	(3041.1,3379,3716.9)	670.375	(3149.1,3499,3848.9)	358.526
Avg. CPU		158.719		155.287		161.327		264.341		207.419

Table 4
The computational study results according to the workload objective.

Problem type	Workload objective				
	Integral value method ($\alpha = 0.6$)	Integral value method ($\alpha = 0.7$)	Integral value method ($\alpha = 0.8$)	Area-based method	Signed distance method
gap-a1	(−166.3,7180.3)	(−110.6,6122.6)	(−116.2,6128.2)	(−247.7,9265.7)	(−120.4,8136.4)
gap-a2	(−112,6124)	(−232.4,4240.4)	(−235.3,9253.3)	(−237.4,2241.4)	(−343.3,7357.3)
gap-a3	(−199,58,315)	(−109.3,37,183.3)	(−146,84,314)	(−154.7,49,252.7)	(−228.9,57,342.9)
gap-a4	(−356.2,58,472.2)	(−472.7,89,650.7)	(−474.5,71,616.5)	(−500.9,29,558.9)	(−370.8,60,490.8)
gap-a5	(−67.9225,517.9)	(−67,214,495)	(12,286,560)	(−143.3191,525.3)	(−105.8182,469.8)
gap-a6	(−286.7237,760.7)	(−299.4202,703.4)	(−278.5277,832.5)	(−340.8274,888.8)	(−305.2188,681.2)
gap-b1	(−123.7,7137.7)	(−180.8,16,212.8)	(−202.2,14,230.2)	(−116.6,8132.6)	(−146.2,14,174.2)
gap-b2	(−114.5,1116.5)	(−231.3,5241.3)	(−229.2,8245.2)	(−181.8,6193.8)	(−342.1,9360.1)
gap-b3	(−205.6,52,309.6)	(−144.7,67,278.7)	(−162.1,65,292.1)	(−176.4,54,284.4)	(−182.4,42,266.4)
gap-b4	(−320.8,28,376.8)	(−312.1,25,362.1)	(−361.1,62,485.6)	(−286.1,39,364.1)	(−267.3,21,309.3)
gap-b5	(−117.5,95,307.5)	(−104.8188,480.8)	(−109.4110,329.4)	(−81.6136,353.6)	(−106.2126,258.2)
gap-b6	(−407.5120.647.6)	(−405.3,89,583.3)	(−286,188,662)	(−349.7,87,523.7)	(−364.9,97,558.9)

Table 5
The ranking scores of the ranking methods.

Problem type	Integral value method ($\alpha = 0.6$)	Integral value method ($\alpha = 0.7$)	Integral value method ($\alpha = 0.8$)	Area-based method	Signed distance method
gap-a1	0.698 (1)	0.535 (3)	0.472 (4)	0.450 (5)	0.637 (2)
gap-a2	0.531 (2)	0.488 (4)	0.230 (5)	0.840 (1)	0.512 (3)
gap-a3	0.615 (3)	0.662 (2)	0.338 (5)	0.671 (1)	0.556 (4)
gap-a4	0.622 (2)	0.344 (5)	0.504 (4)	0.586 (3)	0.624 (1)
gap-a5	0.595 (3)	0.578 (4)	0.286 (5)	0.920 (2)	0.935 (1)
gap-a6	0.594 (2)	0.776 (1)	0.479 (5)	0.518 (3)	0.491 (4)
gap-b1	0.926 (1)	0.439 (4)	0.472 (3)	0.620 (2)	0.168 (5)
gap-b2	0.899 (1)	0.477 (2)	0.116 (5)	0.453 (3)	0.273 (4)
gap-b3	0.667 (2)	0.000 (5)	0.097 (4)	0.587 (3)	0.709 (1)
gap-b4	0.869 (2)	0.539 (4)	0.233 (5)	0.629 (3)	0.903 (1)
gap-b5	0.618 (3)	0.259 (5)	0.576 (4)	0.654 (2)	0.855 (1)
gap-b6	0.401 (4)	0.405 (3)	0.030 (5)	0.647 (2)	0.945 (1)
Avg.ranking scores	0.669 (2.166)	0.458 (3.500)	0.319 (4.500)	0.631 (2.500)	0.634 (2.333)

(1) Initially, the best solution within 10 runs for each problem type and version of the algorithm is determined by using Fuzzy TOPSIS. In Table 3, the cost objective values of the corresponding best solutions are presented with CPU times (seconds), whereas the workload objective values of the same best solutions are presented in Table 4.
(2) Afterwards, the best ranking method for each problem type is again determined by Fuzzy TOPSIS. The ranking scores of different ranking methods are presented in Table 5 for each problem type.

Fuzzy TOPSIS has been widely used for solving practical decision problems due to its simplicity and comprehensibility. The main idea behind Fuzzy TOPSIS is to choose an alternative that has the shortest distance from the positive ideal solution and the farthest distance from the negative ideal solution (Amiri, 2010; Rouhani, Ghazanfari, & Jafari, 2012).

When the ranking methods are examined in terms of CPU times, it is observed that the average CPU times of the integral value method is almost identical, regardless of the α value. On the other hand, the area based and signed distance methods come out with larger average CPU times compared with the integral value method. Having combined these average CPU time values with the average distances and rankings of each method, the integral value with $\alpha = 0.6$ can be concluded as the best performer method for this type of problems. Although the performances of integral value with $\alpha = 0.6$, signed distance and area-based methods in terms of average distances to negative ideal and corresponding average rankings are almost the same, integral value method with $\alpha = 0.6$ outperfomed the other methods in terms of CPU times. Among these methods, integral value method with $\alpha = 0.7$ and 0.8 result the worst performances.

6. Conclusions

In this paper, a direct solution method is utilized to solve fuzzy multiple objective generalized assignment problem by taking the advantage of not requiring any transformation process, since through the transformation process some information can be lost or the number of constraints may increase. The mentioned problem mainly differs from the basic generalized assignment problem on defining the coefficients and right hand side values of the constraints and the objective function coefficients as fuzzy numbers. Bees algorithm and different ranking methods are used in an integrated manner in the proposed direct solution approach successfully. The rationale behind the paper is also to analyze the effects of different ranking methods on the quality of the solution. According to the computational study results of handled generalized assignment problems, integral value ranking method with $\alpha = 0.6$ is the best performer one, signed distance and area-based ranking methods are the second best performers and integral value method with $\alpha = 0.7$ and $\alpha = 0.8$ are the worst performer ranking methods with regard to the quality of the solution and the CPU time requirement.

References

Alfandari, L., Plateau, A., & Tolla, P. (2004). A path relinking algorithm for the generalized assignment problem. In M. G. C. Resende & J. D. Sousa (Eds.), *Metaheuristics: Computer decision-making* (pp. 1–17). Boston: Kluwer Academic Publishers.
Amiri, M. P. (2010). Project selection for oil-elds development by using the AHP and fuzzy TOPSIS methods. *Expert Systems with Applications, 37*, 6218–6224.
Bai, X., Liu, Y.K., & Shen, S.Y. (2009). Fuzzy generalized assignment problem with credibility constraints. In *Proceedings of the eight international conference on machine learning and cybernetics* (pp. 657–662).

Bai, X., Zhang, Y., & Liu, F. (2010). Particle swarm optimization for two-stage fuzzy generalized assignment problem. *Lecture Notes in Computer Science, 6215*(2010), 158–165.

Baykasoğlu, A., & Göçken, T. (2007). Solution of a fully fuzzy multi-item economic order quantity problem by using fuzzy ranking functions. *Engineering Optimization, 39*, 919–939.

Baykasoğlu, A., & Göçken, T. (2008). A review and classification of fuzzy mathematical programs. *Journal of Intelligent and Fuzzy Systems, 19*(3), 205–229.

Baykasoğlu, A., & Göçken, T. (2010). Multi-objective aggregate production planning with fuzzy parameters. *Advances in Engineering Software, 41*(9), 1124–1131.

Baykasoğlu, A., & Göçken, T. (2012). A direct solution approach to fuzzy mathematical programs with fuzzy decision variables. *Expert Systems with Applications, 39*(2), 1972–1978.

Bender, M., & Simonovic, S. (1996). *Fuzzy compromise programming. Water resources research report, 034*. Winnipeg, Man: University of Manitoba.

Bender, M., & Simonovic, S. P. (2000). A fuzzy compromise approach to water resource systems planning under uncertainty. *Fuzzy Sets and Systems, 115*, 35–44.

Cadenas, J. M., & Verdegay, J. L. (2000). Using ranking functions in multiobjective fuzzy linear programming. *Fuzzy Sets and Systems, 111*, 47–53.

Campos, L., & Verdegay, J. L. (1989). Linear programming problems and ranking of fuzzy numbers. *Fuzzy Sets and Systems, 32*, 1–11.

Diaz, J. A., & Fernandez, E. (2001). A tabu search heuristic for the generalized assignment problem. *European Journal of Operational Research, 132*, 22–38.

Fang, S. C., Hu, C. F., Wang, H. F., & Wu, S. Y. (1999). Linear programming with fuzzy coefficients in constraints. *Computers and Mathematics with Applications, 37*, 63–76.

Feltl H., & Raidl, G.R. (2004). An improved hybrid genetic algorithm for the generalized assignment problem. In *Proceedings of the symposium on applied computing* (pp. 990–995), Nicosia, Cyprus.

Fisher, M. L., Jaikumar, R., & Van Wassenhove, L. N. (1986). A multiplier adjustment method for the generalized assignment problem. *Management Science, 32*(9), 1095–1103.

Iskander, M. G. (2002). Comparison of fuzzy numbers using possibility programming: Comments and new concepts. *Computers and Mathematics with Applications, 43*, 833–840.

Jimenez, L. M., Rodriguez, M. V., Arenas, M., & Bilbao, A. (2000). Solving a possibilistic linear program through compromise programming. *Mathware and Soft Computing, 7*, 175–184.

Jimenez, M., Arenas, M., Bilbao, A., & Rodriguez, M. V. (2007). Linear programming with fuzzy parameters: An interactive method resolution. *European Journal of Operational Research, 177*, 1599–1609.

Kahraman, C., & Tolga, A. C. (2009). An alternative ranking approach and its usage in multi-criteria decision-making. *International Journal of Computational Intelligence Systems, 2*(3), 219–235.

Lee, K. H. (2005). *First course on fuzzy theory and applications*. Berlin Heidelberg: Springer-Verlag. Chapter 5.

Liou, T. S., & Wang, M. J. (1992). Ranking fuzzy numbers with integral value. *Fuzzy Sets and Systems, 50*, 247–255.

Majumdar, J., & Bhunia, A. K. (2007). Elitist genetic algorithm for assignment problem with imprecise goal. *European Journal of Operations Research, 177*(2), 684–692.

Mingbiao, L., Xinmeng, C., Feng, J., & Zhijian, W. (2007). Optimization of grid resource allocation combining fuzzy theory with generalized assignment problem. *In Proceedings of the sixth international conference on grid and cooperative computing*.

Nakahara, Y. (1998). User oriented ranking criteria and its application to fuzzy mathematical programming problems. *Fuzzy Sets and Systems, 94*, 275–286.

Nauss, R. M. (2003). Solving the generalized assignment problem: An optimizing and heuristic approach. *Informs Journal of Computing, 15*(3), 249–266.

Olson, D. L. (2004). Comparison of weights in TOPSIS model. *Mathematical and Computer Modeling, 40*(7), 721–727.

Öncan, T. (2007). A survey of the generalized assignment problem and its applications. *Information Systems and Operational Research, 45*(3), 123–141.

Özbakır, L., Baykasoğlu, A., & Tapkan, P. (2010). Bees algorithm for generalized assignment problem. *Applied Mathematics and Computation, 215*, 3782–3795.

Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., Zaidi, M. (2006). The bees algorithm – A novel tool for complex optimisation problems. In *Proceedings of innovative production machines and systems virtual conference* (pp. 454–461).

Prodanovic, P. (2001). *Fuzzy set ranking methods and multiple expert decision making*. Technical Report. The University of Western Ontario, Faculty of Engineering Science.

Prodanovic, P., & Simonovic, S. P. (2002). Comparison of fuzzy set ranking methods for implementation in water resources decision-making. *Canadian Journal of Civil Engineering, 29*(5), 692–701.

Rouhani, S., Ghazanfari, M., & Jafari, M. (2012). Evaluation model of business intelligence for enterprise systems using fuzzy TOPSIS. *Expert Systems with Applications, 39*(3), 3764–3771.

Savelsberg, M. (1997). A branch-and-price algorithm for the generalized assignment problem. *Operations Research, 45*, 831–841.

Tanaka, H., Hichihashi, H., & Asai, K. (1984). A formulation of fuzzy linear programming problems based on comparison of fuzzy numbers. *Control and Cybernetics, 13*, 185–194.

Toktas, B., Yen, J. W., & Zabinsky, Z. B. (2006). Addressing capacity uncertainty in resource-constrained assignment problem. *Computers and Operations Research, 33*(3), 724–745.

Yagiura, M., Ibaraki, T., & Glover, F. (2004). An ejection chain approach for the generalized assignment problem. *Informs Journal of Computing, 16*(2), 131–151.

Yagiura, M., Ibaraki, T., & Glover, F. (2006). A path relinking approach with ejection chains for the generalized assignment problem. *European Journal of Operational Research, 169*, 548–569.

Yao, J. S., & Wu, K. (2000). Ranking fuzzy numbers based on decomposition principle and signed distance. *Fuzzy Sets and Systems, 116*, 275–288.

Zeleny, M. (1973). Compromise programming. In J. Cockrane & M. Zeleny (Eds.), *Multiple criteria decision making* (pp. 262–301). Columbia: University of South Carolina Press.

Zimmermann, H. J. (1976). Description and optimization of fuzzy systems. *International Journal of General Systems, 2*, 209–215.

Zimmermann, H. J. (1983). Fuzzy mathematical programming. *Computers and Operations Research, 10*, 291–298.