Journal of Optimization in Industrial Engineering 15 (2014) 77-83

A Heuristic Algorithm for Nonlinear Lexicography Goal Programming with an Efficient Initial Solution

Mahdi Bashiri[a,*], Amirhossein Parsa Manesh[b], Hamid Hasanzadeh[b]

[a] *Associate professor, Shahed University, Qom Highway, Tehran, Iran*
[b] *MSc Stusent, IE department, Shahed University, Tehran, Iran*

Received 13 March, 2013; Revised 20 September, 2013; Accepted 4 February, 2014

Abstract

In this paper, a heuristic algorithm is proposed in order to solve a nonlinear lexicography goal programming (NLGP) by using an efficient initial point. Some numerical experiments showed that the search quality by the proposed heuristic in a multiple objectives problem depends on the initial point features, so in the proposed approach the initial point is retrieved by Data Envelopment Analysis to be selected as an efficient solution. There are some weaknesses in classic NLGP algorithm that lead to trapping into the local optimum, so a simulated annealing concept is implemented during the searching stage to increase the diversity of search in the solution space. Some numerical examples with different sizes were generated and comparison of results confirms that the proposed solution heuristic is more efficient than the classic approach. Moreover the proposed approach was extended for cases with ordinal weights of inputs or outputs. The computational experiments for 5 numerical instances and the statistical analysis indicate that the proposed heuristic algorithm is a robust procedure to find better preferred solution comparing to the classic NLGP.

Keywords: Nonlinear goal programming; Simulated Annealing; Data Envelopment Analysis; Heuristic algorithm; Efficient initial solution.

1. Introduction

In many real world problems, there are more than one objective. As a popular method, all objectives can be aggregated to a single one; however' it's not feasible or desirable to reduce all existing objectives of the problem to a single objective but we are interested in solving the problem regarding their respected goals. Goal programming is an extension of linear or nonlinear programming that involves deviation of all objectives from their goals. To deal with goals, we need to have the importance weight of each objective. Sometimes it is difficult to achieve their importance weights and the ordinal ranking of objectives can be used as an alternative. So by applying ordinal ranking, goal programming is applied as lexicographic procedure in which each goal is satisfied according to its importance order. The goal programming general formulation can be shown as following:

$$Min\ (w^{(1)} f(d_1^+, d_1^-), \dots, w^{(m)} f(d_m^+, d_m^-)) \quad (1)$$

$$S.t Z_i + d_i^+ - d_i^- = T_i\ ,\ \forall i = 1,\dots,m \quad (2)$$

Where $w^{(i)}$ denotes priority of each objective, Z_i is objective function, T_i stands for goal of each objective, d_i^+-d_i^- are deviations of objective from its goal.

There is a classic algorithm to solve nonlinear lexicography goal programming which is called NLGP.

This algorithm starts to find a solution with least deviation from the more important objective target. Then it is tried to improve other lower ranked objectives without more violation of the high ranked objectives. In classic NLGP, the initial point is chosen randomly which is very important in algorithm efficiency. Random selection of the initial solution will help us to find better final results during search iterations; however, the solution quality cannot be guaranteed. Employing of a method to find more efficient initial solution for NLGP among all possible solutions can be an alternative instead of using random initial solution.In this paper, a method is proposed to find efficient initial solution of the NLGP to improve its performance. There are some methods for measuring of efficiency of decision making units. For example, Data envelopment analysis (DEA), developed by Charnes, Cooper et al. (1978) is a method for assessing the productive efficiency of decision making unit (DMUs)

* Corresponding author E-mail: bashiri.m@gmail.com

which uses some inputs to produce some outputs. On the other hand, DEA is a method for numerical comparisons of efficiencies of DMUs. Data envelopment analysis is a kind of mathematical technique that measures relative efficiencies of decision making units with multiple input-output. Each of these DMUs consumes varying amounts of m inputs and s different outputs. Efficiencies of DMUs are calculated by ratio of their total weighted outputs to their total weighted inputs. In DEA model, there is a constraint that normalizes efficiencies and forces them to be less than or equal to unity. It's clear that more ratios mean more efficiency of DMUs. Suppose that each DMU consumes m input to produce s outputs. Efficiency of each DMU (j=1, 2… n) is calculated by solving the following linear model:

$$Max \sum_{r=1}^{s} u_r y_{ro} \tag{3}$$

s.t.

$$\sum_{i=1}^{m} w_i x_{io} = 1 \tag{4}$$

$$\sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} w_i x_{ij} \leq 0, \qquad j1,2,\dots,n \tag{5}$$

$$w_i \geq \varepsilon, \qquad i = 1,2,\dots,m \tag{6}$$

$$u_r \geq \varepsilon, \qquad r = 1,2,\dots,m \tag{7}$$

Where x_{ij} *and* y_{rj} are ith input and rth output of DMU $j$respectively and o is the index of selected DMU.$w_i$is weight of ith input and u_r is rth output weight. Also ε is non-Archimedean infinitesimal value for preventing weights to be equal to zero. It's clear that this model should be run for each DMU. From a general point of view, each DMU which can produce more outputs by consuming less inputs will have more efficiency. In this paper, the random generated initial solutions are assumed to be decision making units and we are interested in determining the efficient solutions among them to be used as an efficient initial solution of the NLGP algorithm.

Another deficiency of the classic NLGP is it's trapping into the local optimum in most of nonlinear problems. So in this paper a new heuristic nonlinear lexicographic goal programming is proposed in which the probability of its trapping into the local optimum will be reduced. Classic NLGP is so simple but it should solve all sub problems sequentially and throughout iteration within sub problem it cannot terminate by finding sub problem solution. Classic NLGP uses a method which is not capable of finding optimum solution while objectives are nonlinear. There can be a modification in classic algorithm to increase the probability of finding the optimum solution even if objectives are nonlinear. The schematic comparison of the proposed approach with the classic NLGP is illustrated in Figure 1.

This paper is organized as follows; the literature review is presented in the following section. The proposed heuristic approach is illustrated in section 3. Numerical examples and the analysis of results are discussed in section 4. Finally, the conclusion is presented in the last section.

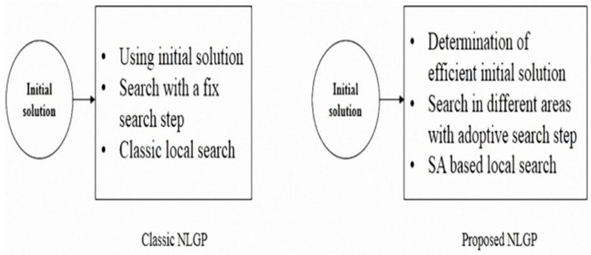

Fig. 1. Schematic comparison of the proposed approach with the classic NLGP

2. Literature Review

Goal programming is used for solving an optimization problem with multiple conflicting goals. The aim of the goal programming is achieving as much goals as possible by minimizing their deviations from their targets. A detailed discussion about different aspects of goal programming (GP) is presented by Ignizio (1978) . As it was mentioned by Zanakis and Gupta (1985), there are some issues that cause the goal programming not to be capable of solving a large spectrum of the real world problems. Then they proposed four suggestions two of which are associated with GP structure. Their first concentration is on the goals priorities and assigning of weight to each goal. The second one is implementation of new algorithm to solve the large scale GP problems. Among various approaches of the GP, lexicographic GP (LGP) with assigning of ordinal weights to objectives is more popular. Tamiz et al.(1995) argue that LGP is the most widely used GP variants and around 64% of the reported applications in the GP literature are related to LGP. According to the research of Köhn (2011), LGP takes places beside the weighted-sum and ε-constraint in a specific category of multi objective techniques that are categorized based on the solution methods. The main difference between the two mentioned methods and LGP is in extracting of the Pareto solutions. The preferred solution will be selected by using of the Lexicographic method instead of extracting the Pareto solutions. From another point of view, Köhn (2011) distinguishes between exact and approximated multi objective optimization solutions due to accuracy of the obtained solution. The exact methods are divided into scalarization and nonscalarization techniques. Scalarization methods convert multi objective optimization into one objective to solve it by conventional techniques. It should be noted that if objectives are not convex, weighted-sum method is incapable of solving a multiple objective problem but lexicographic method does not care that objectives are convex or not.

As mentioned before, lexicographic problems are usually used when there exist conflicting objectives in a decision problem and the objectives should be considered in the hierarchical manner. There are some recent examples which applied lexicographic goal programming techniques to solve real world problems such as Puente et al.(2013), Liberatore et al (2014) and Coshall and Charlesworth (2011). Moreover, some heuristics and metaheuristics have been proposed for solving of multiple objectives problems. According to the research of Jones, Mirrazavi et al. (2002), simulated annealing and Tabu search are two of the most popular meta heuristics used in solving multiple objectives (e.g. Mandow and Pérez de la Cruz, 2001; Suman, 2004; Kulturel-Konak, Smith et al., 2006; Suman, Hoda et al., 2010). Also there are some studies that have used the new algorithm to solve goal programming. Ghoseiri and Ghannadpour (2010) Proposed a new model and solution method for multi objective vehicle routing problem with time windows using goal programming and genetic algorithm. Their model tries to minimize the deviation of objectives from their goals. Modiri et al.(2010) used the mathematic goal programming model in the cement industry using fuzzy and absolute approach to answer which one presents the optimal solution. Du et al.(2014) proposed a multi-objective optimization of reverse osmosis for seawater desalination. Lexicographic optimization and ε-constraint method are proposed to solve the multi-objective optimization problem. Then a fuzzy decision maker is introduced to derive the most efficient solution. Research done by Mandow and Pérez de la Cruz (2001) describes a new general algorithm for graph search problems with additive lexicographic goals. Using lexicographic goals in the formulations helps to provide greater control of solution paths. However, to the best of our knowledge, there is not any research that proposes a new simulated annealing based heuristic used to solve a lexicographic goal programming. Moreover, in this study, a sensitivity analysis has been done for initial point of lexicographic goal programming algorithm, so a method is proposed to find a proper initial solution during the proposed algorithm. Some numerical tests confirm that the proposed algorithm can be used for lots of goal programming problem types such as nonlinear goal functions. The research gap has been depicted in Table 1 by comparing the previous studies with the presented research.

3. The Proposed Heuristic Algorithm

One of the major problems of classic NLGP is its trapping into the local optimum (Saber and Ravindran 1993). Since the classic algorithm tries to search in a special direction with a limited step, so the search in the mentioned direction continues until the objective improvement stops; then, the other direction according to the other variables

Table 1
Comparison of the presented research approach and previous studies

	Year	Search step	Initial solution	Search method	Data
(Mandow and Pérez de la Cruz)	2001	Fix	Random	Classic	Crisp
(Modiri et al.)	2010	Fix	Random	Classic	Fuzzy
(Liao and Kao)	2010	Fix	Random	Classic	Fuzzy
(Arbaiy and Watada)	2011	Fix	Random	Classic	Fuzzy
Coshall and Charlesworth	2011	Fix	Random	Classic	Crisp
(Soliman and Sarker)	2011	Variable	Random	Differential evolution	Fuzzy
(Chen and Xu 2012)	2012	Fix	Random	Classic	Crisp
(Puente et al.)	2013	Fix	Random	Classic	Crisp
(Liberatore et al.)	2014	Fix	Random	Classic	Crisp
(Du et al.)	2014	Variable	Random	ε-constraint	Crisp
This research		Variable	Efficient initial solution by DEA	SA based	Crisp

changes is selected and the algorithm is iterated until the stopping condition is met. However, in the proposed approach the variables change step is not limited and can be more during the search. Moreover, non-improved directions can be accepted by a probability distribution. So the proposed approach has inherited the main characteristic of the simulated annealing in its search algorithm. These features help the algorithm to increase its search diversity among solution space and increase probability of finding the most preferred solution or at least better solution compared with classic NLGP. The pseudo code of the proposed algorithm has been depicted in Figure 2. In each iteration, by choosing the objective with the highest priority, one dimension of the start point is increased in a special direction. Each time that deviation from the goal is decreased, the point is saved as the temporary best one. Moreover, non-improved solutions can be accepted by a probability distribution as illustrated in the pseudo code. Finally, this procedure is stopped by the stopping condition. In the next step, the search is continued in the reverse direction till the stop condition is met. Then, this procedure is resumed by choosing the next dimension and using the last best point. These stages should be repeated for the next lower ranked objectives, considering that violation of high ranked objectives is not acceptable. The selection of efficient initial solution has led to having a more reliable algorithm to solve multiple objectives problems with ordered preferences. More explanation on using the efficient initial solution has been illustrated in the next sections.

4. Numerical Examples and Analysis

It is shown that classic NLGP is dependent to the initial point. It means that by choosing different initial points classic algorithm may spend more or less time to find the preferred solution. In contrast, the proposed heuristic algorithm is less dependent and less sensitive to the initial point. To prove it, we solved several examples using the proposed heuristic and classic NLGP, some of which are reported in Table 2. The reported examples include linear and nonlinear objective functions. Each example is solved using both algorithms by different initial points in 50 runs. Then the final results variations are compared. The analysis show that the proposed algorithm contains less variation in the obtained results comparing to the classic approach as illustrated in Figures 2 and 3. It confirms that the proposed algorithm is more robust and independent of the initial solution comparing to the classic NLGP. The first reason is that the change step is not restricted to a limited size and the second one is related to the nature of accepting non improvement directions during the search in the proposed approach. Feature of accepting the non-improvement direction helps the algorithm to decrease the chance of being trapped in to the local optimum.

Initial x_0 and compute corresponding objective functions values ($z0_l$, $l = 1,2,..,m$)
k: the step size of moving in any dimension
$d_{l-best} = d_{l-gold} = \infty$
$x_{r-best} = x_{r-gold} = x_0$
Repeat
Repeat
If not improve in all dimensions **then** *k=k/2*
Repeat
Repeat
$x_r = x_r + t*k$ (*move in dimension r*)
Compute $\Delta d_l = d_l - d_{l-best}$ for $l = 1,2,..,j$
"d_l:l-th priority objective function deviation of x_r from its target"
If $\Delta d_l \leq 0$ *or* $d_l \leq 0$, $l = 1,2,..,j$
$x_{r-best} = x_r \& d_{l-best} = d_l$
Terminate replication for this dimension
Else if $\Delta d_l \leq 0$, $l = 1,2,..,j-1$
Compute $\Delta z_l = z_l - z_{l-best}$
Prob1=exp($\frac{-\frac{\Delta z_l}{z0_l}}{T}$)
Prob2=random (0, 1)
If prob2 > prob1 **Then** $x_r = x_{r-best}$ And $d_{l-best} = d_l$
End
Update repeat counter
Until (terminate a certain number of repeat in any dimension)
$x_r = x_{r\_}gold$
Repeat
$x_r = x_r$ - t*k *(move in reverse direction in dimension r)*
Compute $\Delta d_l = d_l - d_{l-best}$ for $l = 1,2,..,j$
If $\Delta d_l \leq 0$ *or* $d_l \leq 0 l = 1,2,..,j$
$x_{r-best} = x_r$ And $d_{l-best} = d_l$
Terminate replication for this dimension
Else if $\Delta d_l \leq 0 l = 1,2,..,j-1$
Compute $\Delta z_l = z_l - z_{l-best}$
Prob1=*Exp*($\frac{-\frac{\Delta z_l}{z0_l}}{T}$)
Prob2=random between 0, 1
If prob2 > prob1 **Then** $x_r = x_{r-best}$ And $d_{l-best} = d_l$
End
Update repeat counter
Until (terminate a certain number of repeat in any dimension)
If $d_{l-best} \leq d_{l-gold}$ **then** $x_{r-gold} = x_{r-best}$ And $d_{l-gold} = d_{l-best}$
Update *r(dimension counter)*
Until (end search in all dimensions)

Until (end condition for objective function with priority *j*)
Update j *(selected objective functions counter)*
Until (end number of priorities, *j=m*)

Fig. 2. Pseudo code of the proposed algorithm based on the NLGP in multiple objectives optimization

Table 2
Numerical examples for comparison of the proposed NLGP and the classic NLGP

Example 1	Example 2	Example 3
$\boldsymbol{f^1_{(x)} = -5x_1 + x_2 \le 50}$	$f^1_{(x)} = x_2 + x_1 \le 1000$	$f^1_{(x)} = x_2(x_1 + 1) \le 100$
$\boldsymbol{f^2_{(x)} = 6x_1x_2 \ge 100}$	$f^2_{(x)} = -x_1^3 + 4x_2^2 \ge 5000$	$f^2_{(x)} = -x_1^2 + 4x_2^2 \ge 5000$
$\boldsymbol{f^3_{(x)} = 5x_1 + x_2^2 \le 60}$	$f^3_{(x)} = 10 * x_1 + x_2^2 \le 1000$	$f^3_{(x)} = \sqrt{x_1} + x_2^3 \le 100$
$\boldsymbol{f^4_{(x)} = x_1^2 + x_2 \ge 60}$	$f^4_{(x)} = -x_1x_2 \ge 40$	$f^4_{(x)} = -x_1^3x_2^2 \ge 40$
$\boldsymbol{x_1 \ge 0}$	$f^5_{(x)} = 5x_1^2x_2 \le 700$	$x_1 \le 0$
$\boldsymbol{x_2 \ge 0}$	$x_1 \le 0$	$x_2 \le 0$
	$x_2 \ge 0$	
Example 4	**Example 5**	
$\boldsymbol{f^1_{(x)} = 3x_1 + x_2 \le 400}$	$f^1_{(x)} = 3x_1^3 + x_2 \le 600$	
$\boldsymbol{f^2_{(x)} = -6x_1 + 4x_2 \ge 100}$	$f^2_{(x)} = -6x_1^2 + 4x_2^2 \ge 100$	
$\boldsymbol{f^3_{(x)} = x_1 + x_2 \le 600}$	$f^3_{(x)} = x_1 + x_2 \le 600$	
$\boldsymbol{x_1 \ge 0}$	$f^4_{(x)} = -x_1^2 + 3x_2 \ge 400$	
$\boldsymbol{x_2 \ge 0}$	$x_1 \ge 0$	
	$x_2 \ge 0$	

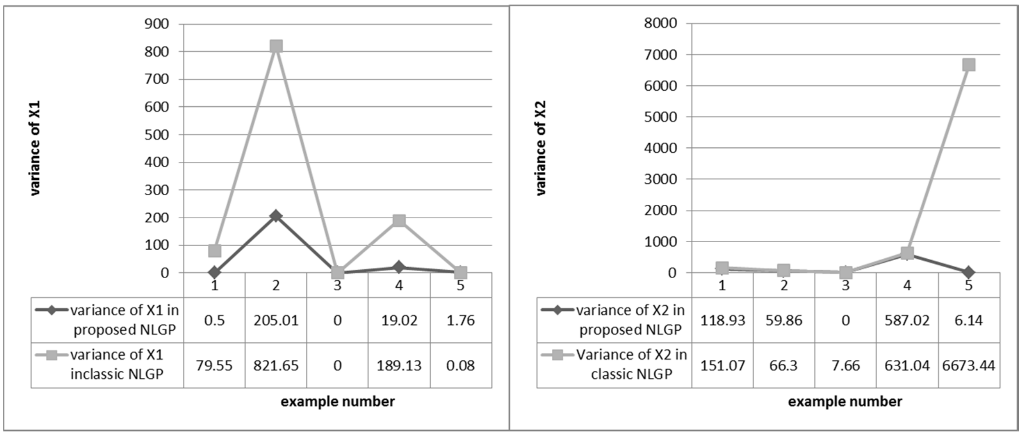

Fig. 3. Robustness of two algorithms results on different initial points for numerical examples in both variables

5. Selecting of Efficient Initial Solution in the Proposed Algorithm

Figure 3 illustrates that both algorithms are sensitive to the initial solution, however, the classic approach has more sensitivity to the initial point, so in this study it is tried to find a proper initial solution. Moreover, it is clear that the proposed heuristic algorithm increases the diversity of search in the solution space to find the better solution, so efficient initial point can decrease the number of iterations to find the preferred solution. As it was mentioned before, the DEA concept can be used to find the most efficient DMU where it can produce more outputs by consuming fewer inputs. So we can set different initial points as decision making units and profit and cost objectives as outputs and inputs, respectively. But it is important to note that the importance level of each objective is different lexicographically. This difference causes classic DEA not to be able to find real efficient points corresponding to the ordinal weighted inputs and outputs. In this paper, a method is proposed to determine weighted inputs and outputs in the classic DEA to find the efficient initial point. This algorithm is illustrated in Figure 4.

In the mentioned algorithm of Figure 4, λ_i and μ_i are the coefficients that are determined based on the importance level of objective functions. As described before, the classic DEA assumes that the inputs or outputs have the same importance. However, in lexicographic goal programming we need to assign an ordinal weight to each objective, so the mentioned coefficients will change the real outputs and inputs to enter their importance to the efficiency calculation model. The DEA is insensitive to the coefficients of outputs and inputs, however, it is sensitive to adding some values to outputs and inputs. In other words, by adding a large number to an input/output in all DMUs, the input/output effect is decreased in efficiency calculation. So the coefficients should be tuned according to the objectives ordinal weights.

Step1 $DMU_t \equiv x_t$ *(start pointt)*

Step2 Sort objectives according to their priories ($f^{(1)}, f^{(2)}, f^{(3)}, \ldots$)

Step3 Ask the decision makerto Determine $\lambda^{(i)}$, $\mu^{(i)}$ as importance factor of inputs and outputs respectively, where $\lambda^{(1)} \le \lambda^{(2)} \le \ldots \le \lambda^{(m)}$ and $\mu^{(1)} \ge \mu^{(2)} \ge \ldots \ge \mu^{(m)}$.

Step4 According to the goal type if decreasing the d_{it}^- is of interest go to step 5, Else go to step 6. (d_{it}^- is the deviation from target for the *i*-th objective in considering of *t*-th initial point according to the equations 1 and 2)

Step5 Calculate F_{it} as following :

$$F_{it} = \lambda_i T_i + \left(\frac{d_{it}^-}{\mu_i \min(all\ t\ \ |d_{it}^-|)}\right)$$

Step6 Calculate O_{it} as following:

$$O_{ij} = \lambda_i T_i - \left(\frac{d_{it}^+}{\mu_i \min(all\ t\ \ |d_{it}^+|)}\right)$$

Step7 put F_{it} as *ith* input of DMU_t and O_{it} *ith* as output of DMU_t .

Step8 Run DEA

Fig. 4. Proposed pseudo code for determining the ordinal weighted inputs and outputs of DEA

Different examples have been solved by classic NLGP and the proposed algorithm in 50 runs. Then the average number of iterations during the search to find the preferred solution is calculated in each example. We applied our method to produce inputs and outputs and applied DEA to find the efficient initial point. Results of searching iterations for both algorithms are reported in Table 3. Finally, we did a paired comparison test with the null hypothesis of $\bar{d}$≥0. The statistical test result confirms that the algorithms with efficient initial point need less computational time when we use DEA to find the initial solution as illustrated in Table 4. Moreover, results show that the *t* statistic value for the proposed NLGP is greater than the *t* statistic value of the classic NLGP; this means that null hypothesis is accepted with more probability for the proposed NLGP.

Table 3
Examples of applying DEA to find the best initial solution of classic and proposed NLGP

	Proposed NLGP		Classic NLGP	
	Average number of iterations	Points found by DEA(A. number of iteration)	Average number of iterations	Points found by DEA(A. number of iteration)
Example 1	4320.833	573	1951.233	36
Example 2	2635.13	1	564.8667	1
Example 3	5312.62	938	1600.36	122.5
Example 4	3426.1	2416.88	1691.46	1559.66
Example 5	3101.72	1	1130.46	1
Example 6	1744.14	542	328.24	255.5
Example 7	2105.92	1371.5	239	211
Example 8	4288.18	3438.667	1422.32	1169

Table 4
Results of paired comparison test with the null hypothesis of $\bar{\mathbf{d}}$≥0

	$\bar{d}$	$S.D$	t_0	p_{value}
Proposed NLGP	2207	1440	4.33	0.998
Classic NLGP	697	722	2.73	0.985

6. Conclusion

This paper proposes a new simulated annealing based heuristic to solve the nonlinear lexicography goal programming. The literature review on the heuristic solutions of the multiple objectives indicates that developing of a solution approach with variable step size during the search stages with considering of efficiency concept for initial solution is a research gap which stimulated this research. The use of this algorithm causes to find a better preferred solution compared to the classic NLGP. Furthermore, we showed that choosing an efficient initial point may affect the number of search iterations to find the final preferred solution and its effect on the proposed NLGP is more than the classic NLGP. In this study, 5 numerical examples were analyzed and finally the comparisons were tested by pairwise comparison statistical test. Selection of more initial points by other efficiency calculation methods and using evolutionary algorithms with multiple starting points in the searching stage can be considered as future directions of this study.

7. References

[1] Arbaiy, N. and J. Watada (2011). Fuzzy Goal Programming for Multi-level Multi-objective Problem: An Additive Model. Software Engineering and Computer Systems. J. Zain, W. Wan Mohd and E. El-Qawasmeh, Springer Berlin Heidelberg. 180: 81-95.

[2] Charnes, A., W. W. Cooper and E. Rhodes (1978). Measuring the efficiency of decision making units. European Journal of Operational Research 2(6): 429-444.

[3] Chen, A. and X. Xu (2012). Goal programming approach to solving network design problem with multiple objectives and demand uncertainty. Expert Systems with Applications 39(4): 4160-4170.

[4] Coshall, J. T. and R. Charlesworth (2011). A management orientated approach to combination forecasting of tourism demand. Tourism Management 32(4): 759-769.

[5] Du, Y., L. Xie, J. Liu, Y. Wang, Y. Xu and S. Wang (2014). Multi-objective optimization of reverse osmosis networks by lexicographic optimization and augmented epsilon constraint method. Desalination 333(1): 66-81.

[6] Ghoseiri, K. and S. F. Ghannadpour (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. Applied Soft Computing 10(4): 1096-1107.

[7] Ignizio, J. P. (1978). A Review of Goal Programming: A Tool for Multiobjective Analysis. The Journal of the Operational Research Society 29(11): 1109-1119.

[8] Jones, D. F., S. K. Mirrazavi and M. Tamiz (2002). Multi-objective meta-heuristics: An overview of the current state-of-the-art. European Journal of Operational Research 137(1): 1-9.

[9] Köhn, H.-F. (2011). A review of multiobjective programming and its application in quantitative psychology. Journal of Mathematical Psychology 55(5): 386-396.

[10] Kulturel-Konak, S., A. E. Smith and B. A. Norman (2006). Multi-objective tabu search using a multinomial probability mass function. European Journal of Operational Research 169(3): 918-931.

[11] Liao, C.-N. and H.-P. Kao (2010). Supplier selection model using Taguchi loss function, analytical hierarchy process and multi-choice goal programming. Computers & Industrial Engineering 58(4): 571-577.

[12] Liberatore, F., M. T. Ortuño, G. Tirado, B. Vitoriano and M. P. Scaparra (2014). A hierarchical compromise model for the joint optimization of recovery operations and distribution of emergency goods in Humanitarian Logistics. Computers & Operations Research 42(0): 3-13.

[13] Mandow, L. and J. L. Pérez de la Cruz (2001). A heuristic search algorithm with lexicographic goals. Engineering Applications of Artificial Intelligence 14(6): 751-762.

[14] Modiri, M., S. M. Rabbani and H. H. Gharebolagh (2010). Influence of fuzzy Goal Programming in Production Optimization Case study: Cement Industry. Journal of Optimization in Industrial Engineering 3(6): 43-52.

[15] Puente-Peinador, J., C. Vela, I. González-Rodríguez, J. Palacios and L. Rodríguez (2013). GRASPing Examination Board Assignments for University-Entrance Exams. Recent Trends in Applied Artificial Intelligence. M. Ali, T. Bosse, K. Hindriks et al., Springer Berlin Heidelberg. 7906: 171-180.

[16] Saber, H. M. and A. Ravindran (1993). Nonlinear goal programming theory and practice: a survey. Comput. Oper. Res. 20(3): 275-291.

[17] Soliman, O. and R. Sarker (2011). Interactive Fuzzy Goal Programming Model Based on Differential Evolution for Regional Sustainability Development under Climate. Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011. E. Corchado, V. Snášel, J. Sedano et al., Springer Berlin Heidelberg. 87: 415-426.

[18] Suman, B. (2004). Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem. Computers & Chemical Engineering 28(9): 1849-1871.

[19] Suman, B., N. Hoda and S. Jha (2010). Orthogonal simulated annealing for multiobjective optimization. Computers & Chemical Engineering 34(10): 1618-1631.

[20] Tamiz, M., D. F. Jones and E. El-Darzi (1995). A review of Goal Programming and its applications. Annals of Operations Research 58(1): 39-53.

[21] Zanakis, S. H. and S. K. Gupta (1985). A categorized bibliographic survey of goal programming. Omega 13(3): 211-222.