Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT
ELSEVIER
Information Sciences 152 (2003) 303–311
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Solving fuzzy optimization problems by evolutionary algorithms

F. Jiménez [a], J.M. Cadenas [a,*], J.L. Verdegay [b], G. Sánchez [a]

[a] *Department Ingeniería de la Información y las Comunicaciones, Universidad de Murcia, Espinardo, 30071 Murcia, Spain*

[b] *Department Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, 18071 Granada, Spain*

Received 1 January 2001; received in revised form 30 October 2002; accepted 7 December 2002

Abstract

In this paper mathematical programming problems with fuzzy constraints are dealt with. Fuzzy solutions are obtained by means of a parametric approach in conjunction with evolutionary techniques. Some relevant characteristics of the evolutionary algorithm are for instance a real-coded representation of solutions and the preselection scheme as niche formation and elitist technique. Three test problems with fuzzy constraints and different structures are used in order to check and compare the proposed technique. The results obtained are very good in comparison with those from another methods.

Keywords: Fuzzy mathematical programming; Evolutionary algorithms

* Corresponding author.

E-mail addresses: fernan@dif.um.es (F. Jiménez), jcadenas@dif.um.es (J.M. Cadenas), verdegay@ugr.es (J.L. Verdegay), gracia@dif.um.es (G. Sánchez).

0020-0255/03/$ - see front matter
doi:10.1016/S0020-0255(03)00074-4

1. Introduction

A constrained optimization problem can be mathematically formulated as

$$\begin{array}{ll} \text{Min} & f(x) \\ \text{s.t.} & g_j(x) \leqslant 0, \quad j = 1, \ldots, m \\ & x \in X \end{array} \tag{1}$$

where $f(x)$, the objective function, and $g_j(x)$, $j = 1, \ldots, m$, the constraints, are defined on $\mathbb{R}^n$, X is a subset of $\mathbb{R}^n$, usually called solution space or feasible set, and x is a vector of n components $x_1, \ldots, x_n$.

This problem must be solved for the values $x_1, \ldots, x_n$ belonging to the solution space that satisfy the constraints and minimize the function f. This is to be meant as one needs to find a feasible point x^*, that is a point $x^* \in X$ satisfying all the constraints, such that $f(x^*) \leqslant f(x)$ for each feasible point x. Then x^* is an optimal solution. Various optimization problems can be categorized based on the characteristics of X, $f(x)$ and $g_j(x)$, $j = 1, \ldots, m$. Thus if $f(x)$ and $g_j(x)$, $j = 1, \ldots, m$ are linear functions the model above describes a linear optimization problem. The very best known example of a linear optimization problem is the linear programming problem which, by means of the simplex algorithm, can be easily solved in a finite number of steps. Otherwise the model becomes a nonlinear optimization problem (fractional, quadratic, etc.) which, in contrast to the linear case, in general is difficult to solve, and usually deterministic algorithms are not applicable.

Although the efforts made to solve efficiently nonlinear problems have produced important progress along the last years, unfortunately there is not a universal method, like in the linear case with the simplex algorithm happens, to solve a general nonlinear programming problem. Consequently many relevant scientific and engineering optimization problems, for which real solutions are needed, cannot be adequately solved. This need of a solution leads to solve this kind of nonlinear programming problems by proposing nonoptimal solutions which may be obtained by different methods, typical and generally called heuristics (neural networks, simulated annealing, tabu search,...). Is just at this point where *evolutionary algorithms* (EA) [1,3,6] appear just as another heuristic tool to try to solve the problems in which here we are interested, i.e., general nonlinear programming problems.

Along the last years EA have shown a good performance in solving optimization problems with complex solution spaces. As it is well known EA imitate, on an abstract level, biological principles of natural selection and genetics such as a population based approach, the inheritance of information, the variation of information via crossover/mutation, and the selection of individuals based on fitness. EA are stochastic search algorithms which often can find better solutions than classical optimization techniques, mainly when an optimal solution for a concrete nonlinear programming problem is difficult to obtain.

Besides the actual difficulty of the problems to be here considered, mainly due to its nonlinear nature, in many real optimization problems encountered in engineering and other areas like artificial intelligence, operations research, etc., coefficients and data taking part into the problem may be affected of an inherent vagueness on their exact values, which often is bridged by providing approximations of the true values that permit to approach solution methods. However the problem obtained as a result of these numerical approximations may be rather than far from the former problem. Among the different types of uncertainty that can appear in the parameters defining the problems, we are here interested in the case in which this vagueness has a fuzzy nature, i.e., when this vagueness is due to a lack of precision, and hence it can be modeled by means of fuzzy sets concepts. The corresponding model is usually referred as a fuzzy constrained optimization problem.

Fuzzy constrained optimization problems have been extensively studied since the seventies. In the linear case, the first approaches to solve the so-called fuzzy linear programming problem were made in [9,11]. Since then, important contributions solving different linear models have been done and these models have been recipients of a great dealt of work. In the nonlinear case the situation is quite different, as there is a wide variety of specific and both practical and theoretically relevant nonlinear problems, each having a different solution method. Consequently in the following we will focus on *fuzzy nonlinear programming* problems in which, as said, coefficients and/or constraints defining the problem are given as fuzzy ones.

To concrete, consider a *nonlinear programming* problem with fuzzy constraints. From a mathematical point of view the problem can be addressed as:

$$\begin{array}{ll} \text{Min} & f(x) \\ \text{s.t.} & g_j(x) \lesssim b_j, \quad j = 1, \ldots, m \\ & x_i \in [l_i, u_i], \quad i = 1, \ldots, n, \quad l_i \geqslant 0 \end{array} \tag{2}$$

where $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is a n dimensional real-valued parameter vector, $[l_i, u_i] \subset \mathbb{R}$ $(i = 1, \ldots, n)$, $b_j \in \mathbb{R}$, $f(x)$, $g_j(x)$ are continuous arbitrary functions, and the symbol $\lesssim$ indicates that the corresponding constraint is a fuzzy constraint [11].

It is patent that EA could be used to solve fuzzy nonlinear programming problems like the above one because of EA are solution methods potentially able of solving general nonlinear programming problems or, at least, of approaching theoretic solution ways that, each case, are to be specified according to the concrete problem to be solved.

Therefore, from this background, the main aim of this paper is to propose and present an EA-based solution method for *fuzzy nonlinear programming* which eventually can be easily adapted to solve general nonlinear programming problems. Consequently the paper is organized as follows. Next section describes an evolutionary-parametric solution approach to solve fuzzy programming problems as in (2), Section 3 presents simulation results for several test problems, and finally, Section 4 points out the main conclusions.

2. Evolutionary techniques for fuzzy programming problems

In almost all the cases, to solve these *fuzzy nonlinear programming* problems are supposed because they can represent well enough the assumed fuzziness of the problem, they have very nice properties and, mainly, they are very easy of handling. Here we will consider the following linear membership function related to each fuzzy constraint:

$$\mu_j(x) = \begin{cases} 0 & \text{if } g_j(x) \geqslant b_j + d_j \\ \dfrac{b_j + d_j - g_j(x)}{d_j} & \text{if } b_j \leqslant g_j(x) \leqslant b_j + d_j \\ 1 & \text{if } g_j(x) \leqslant b_j \end{cases} \tag{3}$$

which gives the accomplishment degree of $g_j(x)$, and consequently of x, with respect to the ith constraint (the decision maker can tolerate violations of each constraint up to the value $b_j + d_j$, $j = 1, \ldots, m$).

Then using the results obtained in [10], the problem (2) can be easily transformed into a parametric programming problem as follows:

$$\begin{array}{ll} \text{Min} & f(x) \\ \text{s.t.} & g_j(x) \leqslant b_j + d_j(1-\alpha), \quad j = 1, \ldots, m \\ & x_i \in [l_i, u_i], \quad i = 1, \ldots, n, \quad l_i \geqslant 0 \end{array} \tag{4}$$

where $\alpha \in [0, 1]$. A fuzzy solution to (4), if there is any, may be defined as the fuzzy set of membership function:

$$\mu_s(x) = \begin{cases} \sup\limits_{x \in S(\alpha)} \alpha & \text{if } x \in \bigcup_\alpha S(\alpha) \\ 0 & \text{elsewhere} \end{cases}$$

where $S(\alpha) = \{x \in \mathbb{R}^n / z(x) = \min_{x' \in X(\alpha)} f(x')\}$ with $X(\alpha) = \{x \in \mathbb{R}^n / g_j(x) \leqslant b_j + d_j(1-\alpha), x_i \in [l_i, u_i], i = 1, \ldots, n, l_i \geqslant 0, j = 1, \ldots, m\}$ and $\alpha \in [0, 1]$.

Unfortunately, there are no much general-oriented solution methods to solve conventional nonlinear parametric programming problems in the literature, although it deserves to mention the cases of integer linear programming problems in which data are continuously varied as a linear function of a single parameter. An evolutionary-parametric based approach to solve fuzzy transportation problems have been proposed in [5]. Therefore in order to theoretically solve (4) we shall try to find an approximate solution. In this way here we are proposing an evolutionary approach finding nonoptimal, but good enough, solutions to problem (4).

Given a set $L = \{\alpha_1, \ldots, \alpha_l\}$, with $\alpha_p \in [0, 1]$ $(p = 1, \ldots, l))$, and $\alpha_p < \alpha_{p+1}$, for all $p = 1, \ldots, l - 1$, we can solve the problem (4) for each $\alpha_p \in L$ by using an EA for nonlinear constrained optimization problems. Thus, l puntual solutions $x^{\alpha_p} = (x_1^{\alpha_p}, \ldots, x_n^{\alpha_p})$, $p = 1, \ldots, l$, could be obtained, from which a fuzzy solution $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_n)$ could be constructed.

Problems as (4), for a given value α_p of the parameter α, are conventional nonlinear constrained optimization problems. As it is well known, the most usual technique in framework of the EA-based constrained optimization is the *penalty method* [4], according to which a constrained problem is transformed into an unconstrained one by associating a cost or penalty to the constraint violations. The success of this approach depends on the way in which the penalties are dealt with. Other approaches such as decoders or repair algorithms [6] also suffer from the disadvantage of being tailored to the specific problem and are not sufficiently general to handle a variety of problems. An overview of EA for constrained parameter optimization problems can be found in [7]. In this section we describe a problem-independent evolutionary technique to solve the problem. Obviously, the main interest is in solving problems with which existing methods have difficulties (complex nonlinear programming problems).

The main characteristics of the EA are the following:

(1) The EA uses a real-coded representation instead of binary as classical *genetic algorithms* [3].
(2) The initial population is generated randomly with a uniform distribution within the boundaries of the search space.
(3) Chromosome selection and replacement are achieved by means of a variant of the *preselection scheme* [3]. This technique is, implicitly, a niche formation technique and an elitist strategy. In each iteration of the EA, two individuals are picked at random from the population. These individuals are crossed n_{children} times and children mutated producing $2 \cdot n_{\text{children}}$ offspring. After, the best of the first offspring replaces the first parent, and the best of the second offspring replaces to the second parent only if the offspring is better than the parent. In this process, the following criteria are used to obtain the best of a collection:
   - A feasible individual is better than an unfeasible one.
   - A feasible individual $V_1 = (x_1^1, \ldots, x_n^1)$ is better than another feasible individual $V_2 = (x_1^2, \ldots, x_n^2)$ if $f(V_1) < f(V_2)$.
   - An unfeasible individual $V_1 = (x_1^1, \ldots, x_n^1)$ is better than another unfeasible individual $V_2 = (x_1^2, \ldots, x_n^2)$ if $\max_j\{h_j^{\alpha_p}(V_1)\} < \max_j\{h_j^{\alpha_p}(V_2)\}$, where $h_j^{\alpha_p}(V) = g_j(V) - b_j - d_j(1 - \alpha_p)$, $j = 1, \ldots, m$, $\alpha_p \in L$.

(4) After an experimentation process, we finally have used two types of crossover, *uniform crossover* and *arithmetic crossover* and three types of mutation, *uniform mutation*, *nonuniform mutation* [7], and *small mutation* which is a mutation that produces only small variations in the individuals. All operators are applied with equal probability.

Note that we are using the min–max formulation for constraint satisfaction. This method is typically used in multiobjective optimization [2] to minimize the relative deviations of the single objective functions from the individual optimum, and it can yield the best possible compromise solution when objectives with equal priority are required to be optimized. Since constraints and objectives can be dealed with in a similar way, and equal priority is assumed for all constraints, the min–max formulation is also appropriate for constraint satisfaction.

3. Simulation results

In this section we check the proposed EA and compare our results with other evolutionary techniques. Nonlinear (crisp) optimization test problems G2, G4 and G7 considered in [8] have been slightly modified in order to introduce fuzzy constraints. The resulting fuzzy test problems are the following:

G2_fuzzy problem

$$\begin{aligned}
\text{Max} \quad & \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right| \\
\text{s.t.} \quad & \prod_{i=1}^{n} x_i \gtrsim 0.75 \\
& \sum_{i=1}^{n} x_i \lesssim 7.5n \\
& 0 \leqslant x_i \leqslant 10 \quad \text{for } 1 \leqslant i \leqslant n
\end{aligned}$$

with violations $d_1 = 0.5$, $d_2 = 2$ and n equal to 20.

G4_fuzzy problem

$$\begin{aligned}
\text{Min} \quad & 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \\
\text{s.t.} \quad & 0 \lesssim 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \lesssim 92 \\
& 90 \leqslant 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \leqslant 110 \\
& 20 \leqslant 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \leqslant 25 \\
& 78 \leqslant x_i \leqslant 102, 33 \leqslant x_2 \leqslant 45, 27 \leqslant x_i \leqslant 45 \text{ for } i = 3,4,5
\end{aligned}$$

with violations $d_1 = 0.5$, $d_2 = 3$

G7_fuzzy problem

$$\begin{aligned}
\text{Min} \quad & x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\
& \quad + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \\
\text{s.t.} \quad & 105 - 4x_1 - 5x_2 + 3x_7 - 9x_8 \gtrsim 0 \\
& -10x_1 + 8x_2 + 17x_7 - 2x_8 \geqslant 0 \\
& 8x_1 - 2x_2 - 5x_9 - 2x_{10} + 12 \geqslant 0 \\
& 3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \gtrsim 0 \\
& -3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 + 7x_4 + 120 \geqslant 0 \\
& -x_1^2 - 2(x_2 - 2)^2 + 2x_1x_2 - 14x_5 + 6x_6 \geqslant 0 \\
& -5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \geqslant 0 \\
& -0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - 3x_5^2 + x_6 + 30 \geqslant 0 \\
& -10 \leqslant x_i \leqslant 10 \quad \text{for } i = 1, \ldots, 10
\end{aligned}$$

with violations $d_1 = 1$, $d_2 = 3$.

The EA was executed with the following parameter values: population size popsize = 100 and number of children $n_{\text{children}} = 10$, maximal generation number $T = 200{,}000$, crossing probability $p_{\text{cross}} = 0.9$ (0.6 uniform crossing and 0.4 not uniform crossing), mutation probability $p_{\text{mutate}} = 0.2$ (0.1 uniform mutation, 0.3 not uniform mutation and 0.6 small mutation).

Tables 1–3 show the results obtained for six values of the parameter α in the best, worst and average cases for each test problem over 10 runs. We can appreciate a good performance of the proposed technique which obtains results

Table 1
Results for G2_fuzzy

α	f_{worst}	f_{aver}	f_{best}	f_{Mic99}	f_{optimal}
0.0	0.839109	0.839811	0.840005	–	0.840021
0.2	0.821812	0.827802	0.829349	–	0.829362
0.4	0.814271	0.820274	0.821064	–	0.821119
0.6	0.799658	0.809708	0.814308	–	0.814364
0.8	0.795877	0.804061	0.808566	–	0.808623
1.0	0.792595	0.800714	0.803589	0.799953	0.803619

Table 2
Results for G4_fuzzy

α	f_{worst}	f_{aver}	f_{best}	f_{Mic99}	f_{optimal}
0.0	−31544.998	−31545.091	−31545.101	–	−31545.101
0.2	−31442.363	−31442.414	−31442.426	–	−31442.426
0.4	−31339.595	−31339.729	−31339.750	–	−31339.750
0.6	−31237.071	−31237.074	−31237.074	–	−31237.074
0.8	−31018.962	−31018.968	−31018.970	–	−31018.973
1.0	−30665.509	−30665.522	−30665.532	−30664.5	−30665.539

Table 3
Results for G7_fuzzy

α	f_{worst}	f_{aver}	f_{best}	f_{Mic99}	f_{optimal}
0.0	23.419	23.072	22.784	–	22.015
0.2	24.139	23.318	23.090	–	22.466
0.4	24.153	23.567	23.350	–	22.921
0.6	24.874	24.029	23.645	–	23.379
0.8	24.490	24.206	24.002	–	23.841
1.0	25.258	24.672	24.322	24.620	24.306

close to optimum values and, in any case, better than results shown in [8]. [1] Values in column f_{optimal} have been obtained with optimization classical techniques in [8] and this work, to be used as reference in comparisons.

4. Conclusions and future works

An evolutionary approach to find approximate fuzzy solutions to optimization problems with fuzzy constraints has been proposed in this paper. In this approach approximate fuzzy solutions are obtained from point solutions which are found by an EA for conventional, crisp, nonlinear optimization problems. We also provided a set of three fuzzy test problems that may serve as a reference for future methods.

Acknowledgements

The authors thank the Comisión Interministerial de Ciencia y Tecnología (CICyT) (Spain) for the partial support given to this work under the project TIC2001-0245-C02-01, TIC2002-04021-C02-01 and TIC2002-04242-C03-02.

References

[1] J. Biethahn, V. Nissen, Evolutionary Algorithms in Management Applications, Springer-Verlag, Berlin, Heidelberg, 1995.
[2] V. Chankong, Y.Y. Haimes, Multiobjective decision making: theory and methodology, in: A.P. Sage (Ed.), Series in Systems Science and Engineering, North-Holland, 1983.
[3] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[1] The column f_{Mic99} represent the values obtained by Michalewicz et al. [8].

[4] A. Homaifar, C.X. Qi, S.H. Lai, Constrained optimization via genetic algorithms, Simulation 62 (4) (1994) 242–254.
[5] F. Jiménez, J.L. Verdegay, Solving fuzzy solid transportation problems by an evolutionary algorithm based parametric approach, European Journal of Operational Research 113 (3) (1999) 688–715.
[6] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 1992.
[7] Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, Evolutionary Computation 4 (1) (1996) 1–32.
[8] Z. Michalewicz, S. Koziel, Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, Evolutionary Computation 7 (1) (1999) 19–44.
[9] H. Tanaka, T. Okuda, K. Asai, On fuzzy mathematical programming, Journal of Cybernetics 3 (4) (1974) 37–46.
[10] J.L. Verdegay, Fuzzy mathematical programming, in: M.M. Gupta, E. Sánchez (Eds.), Fuzzy Information and Decision Processes, 1982, pp. 231–237.
[11] H.J. Zimmermann, Description and optimization of fuzzy systems, International Journal of General Systems 2 (1976) 209–215.