Fuzzy Optimization using Simulated Annealing: An Example Set

Rita Almeida Ribeiro[1] and Leonilde Rocha Varela[2]

[1] Universidade Lusiada
Dept. Economics & Man. Sciences
Rua da Junqueira 190
1349-001 Lisbon, Portugal
Email: rar@lis.ulusiada.pt
[2] Universidade do Minho
Escola de Engenharia
Dept. Produção e Sistemas
4800-058 Guimarães, Portugal
Email: leonilde@dps.uminho.pt

Abstract. In this paper we tested a set of examples, presented in the literature, with a general fuzzy optimization approach using Simulated Annealing (SA). For linear fuzzy problems, we selected a set of examples from well-known authors. For non-linear fuzzy optimization problems we selected two crisp problems and only two fuzzy examples because there are not many fuzzy non-linear examples in the literature. The comparison of the results, obtained with our approach and the ones shown in the literature, allow us to highlight the flexibility, generality and performance of this fuzzy approach to solve either linear or non-linear fuzzy optimization problems.

keywords fuzzy optimisation, simulated annealing, fuzzy constraints, fuzzy parameters.

1 Introduction

The main objective of this paper is to assess the generality, flexibility and performance of a general fuzzification model solved with the simulated annealing algorithm (SA) [1], [2], [3]. The fuzzification model allows fuzzy coefficients, fuzzy resources, fuzzy goals and combinations. This general fuzzy optimization model is able to handle linear and non-linear problems and even unfeasible problems. However, when we have to deal with fuzzy coefficients there is a price to pay because the fuzzification model transforms the problem into a larger dimension one (more constraints and variables) and which also is non-linear.

Since the formulation of the model is non-linear, when we have fuzzy coefficients, we should use an algorithm independent of the formulation, such as SA, to solve the problem. Our rational in this work is that instead of discussing and comparing details of the fuzzy approach used, with other models proposed in the literature, we use their examples to compare their results with the ones obtained with our approach. The set of examples tested can act as a test-bed for any fuzzy approach proposed in the literature.

Specifically, for linear problems we selected a set of examples from representative authors and methods, such as: Carlsson and Korhonen [4]; [5]; Lai & Hwang [6]; Sakawa [7]; Tanaka, Ichihashi and Asai [8]; Delgado, Verdegay and Vila [9, 10]; and Zimmermann [11]. In addition, we solved a linear unfeasible solution example [12] to show that the approach formulation, can also provide solutions for unsolvable crisp problems. For non-linear problems we selected four examples, two fuzzy non-linear problem by Sakawa [7] and two crisp problems [13]. The latter selection was based on the scarcity of non-linear fuzzy examples (exceptions can be seen in [14] [7] [15]).

The selected set of examples are, first, formulated with the fuzzification model of Moura-Pires and Ribeiro [1] [2] and then solved with an SA algorithm implementation proposed by the authors [3]. With this comparison we can show how general and flexible both the fuzzification model and the SA algorithm are to solve different types of fuzzy optimization problems, from linear to non-linear, as well as unfeasible problems.

This paper is organized as follows. This first section introduces the objectives of this work and gives a brief introduction of fuzzy optimization concepts. Section 2 describes the main characteristics of the fuzzification model, as well as of the solution algorithm used to solve all the examples (different methods). Section 3 introduces the set of linear examples from different authors, solves them and discusses the results obtained by these authors and with our approach. Section 4 follows the same logic as section 3, but introduces the set of non-linear examples tested. Section 5 presents the conclusions.

2 Basics on the approach used for formulating and solving the example set

In this section we introduce the main characteristics of the fuzzy approach used in this paper [1] [2] [3], to formulate and solve a set of examples in fuzzy optimization. To make this section more easy to read we divided it in four subsections: the first briefly introduces fuzzy optimization main concepts; the second subsection presents the fuzzification model used in this work to formalize the example set; in the third subsection we describe the implemented solution process and the reasons to use the simulated annealing algorithm.

2.1 Introduction to fuzzy optimization

The main objective of a fuzzy optimization method is to find the "best" solution (decision alternative) in the presence of incomplete information, i.e., imprecise information and/or in the presence of "vague" limits in the information. There exist many forms of imprecision in fuzzy optimization problems, as for example, variable coefficients that we do not know precisely (for example, "processing times of about one hour for assembling a piece") and constraint satisfaction levels with imprecise limits (for example, "the total processing time available is around 100 hours").

A classical linear optimization problem consists on maximizing or minimizing a certain objective function subject to a set of constraints, which express, for example, the resource limitations. Formally,

$$\begin{aligned} \max / \min \; & Z = Cx \\ & Ax\{\geq, \leq, =\}B \\ & x \geq 0 \end{aligned} \tag{1}$$

The fuzzy version of this problem is generally formalized as,

$$\begin{aligned} \max / \min \; & \tilde{Z} = \tilde{C}x \\ & \tilde{A}x\{\geq, \leq, =\}\tilde{B} \\ & x \geq 0 \end{aligned} \tag{2}$$

where $\tilde{Z}$ represents a fuzzy goal, $\tilde{C}$ is the vector of fuzzy costs, $\tilde{A}$ is the matrix that contains the fuzzy coefficients of the objective(s) and of the constraints and $\tilde{B}$ is the corresponding vector of the limits of the resources. The "tilde" on top of the parameters means that they are defined by fuzzy sets. We opted for using the "tilde" on top of the right hand side parameter and not in the constraint sign has a uniform concept of fuzzy parameter for resource limits, fuzzy coefficients and fuzzy goals. In section 2.2 this point is discussed further.

The first fuzzy extension of the classical optimization problem to a fuzzy environment is due to Bellman and Zadeh [16]. Based on the similarity model of the latter authors, Zimmerman was the first author to propose a method to solve fuzzy linear programming problems with fuzzy resources (constraints) and fuzzy goals [17] [11]. Nowadays there are many fuzzification methods proposed in the literature, for resources fuzzification, goal fuzzification as well as for coefficients fuzzification (see good overviews in [18] [19] [6] [14, 20]).

2.2 Flexible fuzzification model used

The set of examples that are tested in this paper were formulated with a fuzzy model proposed by Ribeiro and Moura-Pires [1] [2]. This model allows using the following types of fuzzifications in the optimization problem: (a) fuzzy coefficients in the objective function; (b) fuzzy coefficients in the left-hand side of the constraints; (c) fuzzy resource limits of the constraints; (d) fuzzy goals; (e) any combination of the previous. Assuming the general formulation for fuzzy optimization problems defined in (2) the fuzzy model used in this work is,

$$\max Z = \sum_{k=1}^{K} \tilde{c}_k \cdot x_k$$
$$\sum_{k=1}^{K} \tilde{a}_{ik} \cdot x_k \{\geq, \leq, =\} \tilde{b}_i, i = 1, \ldots, N$$
$$x \geq 0 \tag{3}$$

The optimization model (3) is then transformed into the following system of non-linear fuzzy constraints:

$$\begin{cases} \max Z = \sum_{k=1}^{K} w_k \cdot x_k \\ \begin{cases} \max M = \min(\mu_{aik}, \mu_{bi}, \mu_{ck}) \\ \sum_{k=1}^{K} y_{ik} \cdot x_k \{\leq, =, \geq\} \tilde{b}_i \\ y_{ik} = \tilde{a}_{ik} \\ w_k = \tilde{c}_k \end{cases} \end{cases} \tag{4}$$
$$k = 1, \ldots, K \quad i = 1, \ldots, N \quad x, y, z \geq 0$$

This mathematical transformation implies the addition of as many new fuzzy constraints and as many new variables as the fuzzy coefficients of the problem (e.g. for 2 fuzzy coefficients we add two new variables and two new constraints to the formulation). Any new added constraint is represented as an equality constraint with a fuzzy resource limit; this process allows the handling of all constraints in a similar fashion. The fuzzification model includes two objectives, the initial one and another objective (M) to obtain the best of the worst violations of the fuzzy parameters, in the sense of the maxmin model [11]. In general, the objectives of this fuzzification model are two fold:

1) Find the best values for x, y and w that maximize the minimum aggregated membership values (denoted by M), considering a threshold value for a minimum acceptable violation level of the constraints.

2) Find the optimal value of Z that satisfies all the constraints as well as the first step.

The aggregation described in 1) represents the intersection of all the membership values of the fuzzy parameters considered (i.e. coefficients, and/or resource limits, and/or goals), to indicate that all the constraints/coefficients have to be satisfied with a certain level (min). In addition, a threshold value was included in our implementation to allow the decision maker to specify if he/she wants an optimistic scenario (high satisfaction level), average scenario (average satisfaction level) or a pessimistic scenario (low satisfaction level). In point 2) it must be pointed that we are describing just a single objective function, but the fuzzification model can handle multi-objective problems [21].

To clarify the non-linearity of the model lets consider that $\tilde{c}_k$ is a fuzzy set that indicates how acceptable are the values around c_k. Let $w = (w_1, w_2, \dots, w_K)$ be a set of objective function costs such that $\mu_{\tilde{c}_k}(w_k) > 0$. In fact, for each different combination of values w_k a different function to maximize over x is obtained and each set has a satisfaction level defined by: $\mu_{\tilde{C}}(w) = \min_k \mu_{\tilde{c}_k}(w_k)$ and $W_\alpha = \{w : \mu_{\tilde{C}}(w) > \alpha\}$. The same logic applies to constraint coefficients a_{ik}. For more details about the fuzzification model, see [1]).

One of the drawbacks of the fuzzification model is the addition of more constrains and variables, whenever there are fuzzy coefficients. However, it provides the advantage of handling all fuzzy coefficients in the same manner of fuzzy constraints, hence, we can know how much each constraint (being a resource or coefficient) is being violated. Another important drawback is the non-linearity of the model. The compensation for this disadvantage is the generality of the model since it can handle either linear or non-linear fuzzy optimization problems

In conclusion we can say that this fuzzification model provides trade-offs between constraint satisfaction (objective M) and the original problem objective (Z). Both fuzzy coefficients $\tilde{c}_{ik}$ and $\tilde{a}_{ik}$ and the resources $\tilde{b}_i$ are all handled in a similar fashion, i.e. as fuzzy constraints. Further, this method allows manipulating either linear or non-linear fuzzy optimization problems, as well as unfeasible crisp problems [1] [21] [2].

2.3 Solution algorithm: Simulated Annealing (SA)

Before discussing the main characteristics of the SA algorithm and of our implementation, we need to clarify why we selected the SA algorithm for solving the general fuzzification model (4), used in this work. First, and most important, since our fuzzification model is non-linear we need an algorithm that was independent of the problem to be solved, i.e. an algorithm that could solve either linear or non-linear optimisation problems. Second, it is to easy to understand because the parameters have an analogy with the annealing process of a solid [22] and the algorithm does not have too many parameters to handle. Third, since it is a guided-random search algorithm, it allows us to control the search for the “”optimum” (e.g. with parameter temperature and the stopping criteria) [22]. Fourth, the algorithm is quite simple to implement and the computational time to achieve “good” results is quite acceptable.

In 1983, Kirkpatrick and others originally proposed the Simulated Annealing algorithm using, as mentioned, an analogy with the annealing process of a solid [23] [22]. The objective of an algorithm of this nature is to find the best solution among a finite number of possible solutions, however, it does not guarantee that the solution found is indeed the global optimum. This last characteristic restricts its use to the cases where “good” local optima are acceptable. The SA technique is also interesting because it allows finding near-optimal solutions within a reasonable computational time frame. The worst drawback of the SA algorithm is the need to provide "good" initial points to run the algorithm – without good initial points the simulation can get easily stuck in a local minima [2].

The SA algorithm requires the definition of a neighbourhood structure, as well as the parameters for the cooling process [22]. A temperature parameter allows distinguishing among deep or slight alterations in the objective function. Drastic alterations occur at high temperatures and small or slight modifications at low temperatures. The four basic requirements for using the SA algorithm in combinatorial optimization problems are: a concise problem description; a random generation of the alterations from one configuration to another; an objective function that includes the utility function of the trade-offs; and a definition of the initial state, of the number of iterations to be executed for each temperature and its annealing process.

In general we can say that we have to specify: (a) how to generate a state y, neighbour of x; (b) which aggregation function (M) to use; (c) the selection of number of neighbours to generate; (d) the temperature decrease function; (e) and finally the algorithm stopping criteria. In our implementation we followed the fuzzification model described in (4) but with the simplification of assuming a single objective (Z), besides the aggregated violations of constraints (objective M in (4)). Further, when we use the SA algorithm for solving fuzzy optimization problems, the decision maker can select thresholds levels (α-cuts) as well as the tolerances/deviations (i.e.fuzzification) for each constraint parameter and/or for each objective function's coefficient and/or for each constraint's coefficient. In addition, we must point that this implementation was based on a first prototype explained in [2] but with modifications and additions, as for example the notion of *seed* [3].

As mentioned, we included in the algorithm implementation the notion of seed for the random numbers generation [22] to allow the generation of identical values (i.e. same random numbers), in different program executions but for the same example. With the seed we can repeat the a priori conditions for testing the same problem with different types of fuzzification (e.g. coefficients or resources or both). For our tests we used a seed value of 1.

For more details about the SA algorithm and its implementation, used to test the set of selected examples (after formulating them with the fuzzification model (4)), see [3].

3 Set of linear examples tested and discussion of the results

Considering that there are many methods proposed in the literature to solve fuzzy optimization problems we selected arbitrarily a set of linear and non-linear examples from the literature. We believe the set is a good sample to compare different results with the ones obtained with our approach. Instead of comparing the approach used in this work with other approaches our objective is to test a set of diverse linear and non-linear fuzzy optimization problems. We solve the set of examples with our approach and then we compare our results with both the crisp results and the fuzzy results of the authors (when we have them). This is a more understandable and simple way to discuss the benefits of our approach. We must point that in some of the examples we made some simplifications to enable comparisons between our results and the ones found in the literature. All modifications are explained with each example presentation.

In order to standardize the tests we also made some assumptions regarding the membership functions of the problems tested. In this work we always use linear triangular membership functions to solve the examples. According to the sign of the equation (e.g. bigger than, equal to, less than) we use the following notation: left-open-triangle=]value, tolerance]; right-open-triangle=[tolerance, value[; triangular= [leftTolerance, value, rightTolerance]. For the linear set of examples we followed the author's fuzzy parameters limits as close as possible. For the set of non-linear examples, instead of following the deviations (membership functions limits) considered by the author we simplified them to a 10% flexibilization either in the coefficients or in the resources limits or in the goals. This 10% tolerance is smaller than the one used by the authors but this strengths our claim of achieving better results.

The presentation and discussion of the examples (section 3 and 4) will follow the steps:

- First we depict the initial example (proposed by each author) and the fuzzified parameters that were considered in the respective method (when we have them).
- Second, we transform the example with our fuzzification model (4) but for reasons of space we only show this formulation for example 3.1. Besides the authors fuzzy parameters limits we also show our fuzzy parameters limits, if we performed some modification. We do not show our fuzzified limits for the non-linear examples because, as mentioned above, they were simplified to a 10% tolerance from the central value.
- Third, we solve the problem with the SA algorithm implementation, described in sub-section 2.3, and we discuss and compare the results we obtain with three thresholds (0.3, 0.6 and 0.9) with both the one provided by the author (denoted fuzzy-author) and the crisp result (when we have it). In some cases we added another threshold solution to enable us to compare our result with the similar one presented by an author.

It is important to remind that we will only show the mathematical transformation formulation that is performed in the examples using (4) for the first example (3.1.), because of two aspects: space considerations and because the transformation is quite straightforward. Hence, we will only show the original example formulation, the fuzzified parameters limits and the final results comparison.

Although our approach may consider all kinds of fuzzification, we solve each example with just the respective fuzzification proposed by the authors. This is the only way to make a meaningful comparison between results. As mentioned, in some cases we did simplifications but they are indicated and do not change the discussion.

In addition, we perform more than one simulation for each example, with different threshold values, to obtain three different scenarios: optimistic (high satisfaction level, 90%), average (average satisfaction level, 60%) and pessimist (low satisfaction level, 30%). These scenarios will provide more information for comparative purposes and allow us to neglect the small modifications done in the fuzzification of the examples. For each alternative test we also show the computational execution time that the SA implementation took to solve the problem. This will allow us to assess the computational effort of the fuzzification approach used.

3.1 Example by Tanaka, Ichihashi, Asai (in: [6])

In the case of Tanaka, Ichihashi and Asai method we used the example found in [6] instead of the original one [24] because it is a simpler version using the same method. The example is,

$$\begin{aligned} &\max\ 25x_1 + 18x_2 \\ &\quad s.t\ \tilde{a}_{11}x_1 + \tilde{a}_{12}x_2 \leq \tilde{780} \\ &\qquad \tilde{a}_{21}x_1 + \tilde{a}_{22}x_2 \leq \tilde{380} \\ &x_i \geq 0 \end{aligned} \tag{5}$$

where the fuzzy parameters were: $a_{11} = [12, 18]$; $a_{12} = [32, 36]$; $b_1 =]780, 850]$; $a_{21} = [19, 21]$; $a_{22} = [7, 13]$; $b_2 =]380, 480]$

Let us consider the following fuzzy parameters for the coefficients: $a_{11} = [12, 15, 18]$; $a_{12} = [31.96, 34, 36.04]$; $a_{21} = [19, 20, 21]$; $a_{22} = [7, 10, 13]$ and for the resources identical limits. Now transforming the example with our formulation (4) we obtain,

$$\max Z1 = 25x_1 + 18x_2$$
$$\begin{cases} \max Z2 = \min(\mu_{a_{11}}, \mu_{a_{12}}, \mu_{a_{21}}, \mu_{a_{22}}, \mu_{b_1}, \mu_{b_2}) \\ \quad s.t.\ y_{11}x_1 + y_{12}x_2 \leq \tilde{780} \\ \qquad y_{21}x_1 + y_{22}x_2 \leq \tilde{380} \\ \qquad y_{11} = \tilde{15} \\ \qquad y_{12} = \tilde{34} \\ \qquad y_{21} = \tilde{20} \\ \qquad y_{22} = \tilde{10} \\ \qquad x_i, y_i \geq 0 \end{cases} \tag{6}$$

The solutions for our three thresholds, the crisp solution and the fuzzy solution of the author are:

solutions	#	Z	x_1	x_2	μ	Time
Crisp solution	1	577.736	9.660	18.672	—	—
Fuzzy Author	2	623.54	12.14	17.78	0.4	—
Fuzzy ours: α= 0.3	3	673.267	12.336	20.270	0.3151	15"
Fuzzy ours: α= 0.4	4	648.875	11.699	19.800	0.4144	16"
Fuzzy ours: α= 0.6	5	644.708	12.320	18.707	0.6337	17"
Fuzzy ours: α= 0.9	6	596.519	10.386	18.716	0.9024	17"

Observing the results above we see that with our method most of our solutions, #3, #4, #5 are better than solution #2 of Tanaka, Ichihashi and Asai. Only when we set a threshold of $\alpha = 0.9$ our solution is worse than the one by Tanaka, Ichihashi and Asai with $\mu = 0.4$ (596.519 vs. 623.54). Again the results clearly show the trade-off that happens with our approach: for better satisfaction of constraints and coefficients (less violation) we get worse values for our objective function.

The time to achieve a solution with the SA implementation is quite reasonable, for all our tested solutions (around 16 seconds).

3.2 Example by Carlsson and Korhonen [4]

Carlsson and Korhonen proposed an interesting method that considers a complete fuzzification of linear programming problems. These authors used the following example to illustrate their method,

$$
\begin{aligned}
\max\ & [1, 1.5)x_1 + [1, 3)x_2 + [2, 2.2)x_3 \\
s.t.\ & \\
& [3, 2)x_1 + [2, 0)x_2 + [3, 1.5)x_3 \leq [18, 22) \\
& [1, 0.5)x_1 + [2, 1)x_2 + [1, 0)x_3 \leq [10, 40) \\
& [9, 6)x_1 + [20, 18)x_2 + [7, 3)x_3 \leq [96, 110) \\
& [7, 6.5)x_1 + [20, 15)x_2 + [9, 8)x_3 \leq [96, 110) \\
& x_i \geq 0
\end{aligned} \tag{7}
$$

where the intervals used for the coefficients and resources fuzzification are represented by exponential functions with the intervals shown in the example.

We formulated this example using the the open interval limit as the central point for the tolerances, in the same fashion as Lai & Hwang [6],

$$
\begin{aligned}
\max\ & 1.5x_1 + 3x_2 + 2.2x_3 \\
s.t.\ & \\
& 2x_1 + 1.5x_3 \leq 22 \\
& 0.5x_1 + 1x_2 \leq 40 \\
& 6x_1 + 18x_2 + 3x_3 \leq 110 \\
& 6.5x_1 + 15x_2 + 8x_3 \leq 110 \\
& x_i \geq 0
\end{aligned} \tag{8}
$$

and our fuzzy parameters are: $C_1 = [1, 1.5, 2]$; $C_2 = [1, 3, 5]$; $C_3 = [2, 2.2, 2.4]$; $a_{11} = [1, 2, 3]$; $a_{13} = [0.51.5, 3]$; $b_1 =]18, 22]$; $a_{21} = [0, 0.5, 1]$; $a_{22} = [0, 1, 2]$; $b_2 =]10, 40]$; $a_{31} = [3, 6, 9]$; $a_{32} = [16, 18, 20]$; $a_{33} = [0, 3, 7]$; $b_3 =]96, 110]$; $a_{41} = [6, 6.5, 7]$; $a_{42} = [10, 15, 20]$; $a_{43} = [7, 8, 9]$; $b_4 =]96, 110]$.

As mentioned above, for comparative purposes we show the solutions obtained with our fuzzy approach (2 simulations for different thresholds), the crisp solution for the problem, and the respective two solutions from author method.

solutions	#	Z	x_1	x_2	x_3	μ	time
Crisp solution	1	12	0	0	6	—	—
Fuzzy Author 1	2	23.76	0	1.22	10.44	0.3	
Fuzzy Author 2	3	13.08	0	0	6.52	0.9	—
Fuzzy ours α= 0.3	4	34.110	4.220	3.030	7.220	0.33	1'24"
Fuzzy ours α= 0.6	5	29.938	8.733	2.666	2.693	0.68	8'5"
Fuzzy ours α= 0.9	6	17.663	2.723	2.970	2.119	0.95	23'36"

As can be observed in the results obtained, with our approach we obtain much better results for the similar cases, #2 and #3 of Carlsson and Korhonen versus ours #4 and #6. In particular, there is a big difference between our solution #4 and the equivalent Carlsson and Korhonen solution #2 (34.110-23.76= 10.35!), if we note that the allowed violations of the fuzzy parameters were similar $\mu = 0.33$ vs $\mu = 0.3$. The differences using both methods are quite significant and show that by simplifying the membership functions to triangular ones helped to obtain better results. The time to solve this problem is large, particularly for high thresholds (e.g. $\alpha = 0.9$), which mean having less violation of the fuzzy parameters. I.e. the S.A.algorithm implementation takes time to find a solution when we require better satisfaction levels (less flexibility) for completely fuzzified problems.

3.3 Example by Chanas [5]

Chanas used the following example (fuzzy resources) to test his method,

$$\begin{aligned} &\max\ 3x_1 + 4x_2 + 4x_3 \\ &s.t.\ 6x_1 + 3x_2 + 4x_3 \leq \tilde{1200} \\ &\quad\ \ 5x_1 + 4x_2 + 5x_3 \leq \tilde{1550} \\ &\quad\ \ x_i \geq 0 \end{aligned} \tag{9}$$

where the fuzzy parameters were $Z = [1600, 1750[$; $b_1 =]1200, 1300]$; $b_2 =]1550, 1750]$.

In our approach the formulation and fuzzy parameters used are identical and the solutions obtained were:

solutions	#	Z	x_1	x_2	x_3	μ	time
Crisp solution	1	1550	0	387.5	0	—	—
Fuzzy Author	2	1649.8	0	412.45	0	0.57	—
Fuzzy ours α= 0.3	3	1691.732	0.792	416.918	5.421	0.3	16”
Fuzzy ours α= 0.4	4	1656.010	2.929	408.313	3.492	0.451	13”
Fuzzy ours α= 0.5	5	1646.578	3.090	405.108	4.219	0.5102	16”
Fuzzy ours α= 0.6	6	1626.023	2.195	402.956	1.905	0.6080	14”

Observing the results we see that our solutions are better only for a $\mu < 0.5$. This implies that we need to violate the constraints a little more than with the Chanas parametric approach to obtain better results. However, Chanas´ method does not handle a complete flexibilization of the model and does not have the facility of setting a threshold satisfaction level. For example, with a smaller violation of the constraints ($\mu = 0.3$ versus $\mu = 0.57$) we obtain a better level for the objective function (1691.732 versus 1649.8). Of course, in our method, the higher the violation of the constraints the smaller the objective function is.

Chanas [5] also solved other examples using his parametric method such as Zimmermann multi-objective example (example 3.10). Since the results obtained were similar for both authors, we leave the comparison of those results for later.

3.4 Example by Lai & Hwang [6]

Lai and Hwang presented an example with fuzzy resources to discuss their parametric method,

$$\begin{aligned} \max\ & 4x_1 + 5x_2 + 9x_3 + 11x_4 \\ s.t.\ & x_1 + x_2 + x_3 + x_4 \leq \tilde{15} \\ & 7x_1 + 5x_2 + 3x_3 + 2x_4 \leq \tilde{120} \\ & 3x_1 + 5x_2 + 10x_3 + 15x_4 \leq \tilde{100} \\ & x_i \geq 0 \end{aligned} \tag{10}$$

where the fuzzy parameters were $b_1 =]15, 18]$; $b_3 =]100, 120]$. Our formulation followed the same fuzzy parameters tolerances. The results obtained were:

solutions	#	Z	x_1	x_2	x_3	x_4	μ	time
Crisp solution	1	99.29	7.14	0	7.86	0	—	—
Fuzzy Authors τ=0.1	2	101.28	7.283	0	8.017	0	0.1	—
Fuzzy Authors τ=0.3	3	105.248	7.569	0	8.331	0	0.3	—
Fuzzy Authors τ=0.9	4	117.19	8.427	0	9.273	0	0.9	—
Fuzzy ours: α= 0.1	5	116.463	8.394	0.210	8.246	0.693	0.11	20”
Fuzzy ours: α= 0.3	6	110.686	8.628	0.339	7.018	1.029	0.33	19”
Fuzzy ours: α= 0.9	7	101.223	6.852	0.453	7.746	0.167	0.90	19”

Comparing results #2 and #3 from Lai and Hwang with our corresponding first two solutions (#5 and #6) we see that our method performs better than the authors method, i.e. we obtain higher objective function values. However we get a smaller value for solution #7 vs. #4 because our philosophy is that for higher satisfaction values of the fuzzy parameters (i.e. less violation of constraints) we should "pay" more. We do have a trade-off between better solutions and bigger violation of the fuzzy parameters.

Lai and Hwang also tested the same problem for bigger violations of the constraints, but since our method would always perform better (due to its generality) for lower thresholds (higher violation of constraints) we did not perform more comparisons.

3.5 Example by Sakawa [7]

Sakawa presented the following multiple objective example,

$$\begin{aligned} &\min\ C_{11}x_1 - 4x_2 + x_3 \\ &\max\ -3x_1 + C_{22}x_2 + x_3 \\ &\text{equal}\ 5x_1 + x_2 + C_{33}x_3 \\ &s.t.\ a_{11}x_1 + a_{12}x_2 + 3x_3 \leq 12 \\ &\quad x_1 + 2x_2 + a_{23}x_3 \leq b_2 \\ &\quad x_i \geq 0 \end{aligned} \tag{11}$$

where the fuzzy parameters are $C_{11} = [0, 2, 2.5]$; $C_{22} = [-1.25, -0.75, -0.25]$; $C_{33} = [-0.25, 0, 1]$; $a_{11} = [0, 3, 4]$; $a_{12} = [0.5, 1, 1.5]$; $a_{23} = [0.5, 1, 1.5]$; $b_2 = [8, 12, 14]$.

In order to use our SA implementation we have to transform the problem into a single objective one. It should be noted that our fuzzification model (section 2.2.) allows solving multiple objective problems but the implemented algorithm solution is not yet prepared for this. Hence, we start by considering the following simplifications in the objective functions,

$$\begin{aligned} &\max\ -C_{11}x_1 + 4x_2 - x_3 \\ &\max\ -3x_1 + C_{22}x_2 + x_3 \\ &\max\ 5x_1 + x_2 + C_{33}x_3 \end{aligned} \tag{12}$$

After we use a simple combination of the objectives to obtain a single objective and our formulation of the example becomes,

$$\begin{aligned} &\max\ -\tilde{2}x_1 + 4x_2 - x_3 - 3x_1 + 0.\tilde{75}x_2 + x_3 + 5x_1 + x_2 + \tilde{0}x_3 \\ &s.t.\ \tilde{3}x_1 + \tilde{1}x_2 + 3x_3 \leq 12 \\ &\quad x_1 + 2x_2 + \tilde{1}x_3 \leq \tilde{8} \\ &\quad x_i \geq 0 \end{aligned} \tag{13}$$

where our fuzzy parameters are identical to the above except for $b_2 =]8, 14]$.

The solutions obtained were:

solutions	#	Z	x_1	x_2	x_3	μ	time
Crisp solution	1	17	0	4	0	—	—
Fuzzy Author	2	12.91(*)	2.3074	2.3443	1.921	0.65	—
Fuzzy ours: α= 0.3	3	34.729	0.196	5.756	0.000	0.3737	36”
Fuzzy ours α= 0.6	5	25.552	0.015	4.943	0.000	0.6773	1’34”
Fuzzy ours: α= 0.9	6	19.281	0.016	4.276	0.000	0.951	7’39”

(*) approximate solution calculated from the three O.F. in Sakawa example.

In this case it is more difficult to compare the results because we made some assumptions in the simplifications to obtain a single objective. However, if we consider that the decision maker wants to have the best possible value for all objectives and that the equality objective can be transformed into a maximizing objective, then our results are much better than the ones presented by Sakawa (for all solutions #3, #4, #5, #6). It should also be noted that there seems to be a correlation between the time to solve problems and higher threshold levels (less violation) when the problems include fuzzification of some coefficients (time=7’39” for #6). Our SA implementation algorithm behaviour decreases when there is an increase in both fuzzy parameters and higher satisfaction values for the violations (less flexibility).

3.6 Example by Delgado, Verdegay, Vila [9]

Delgado, Verdegay and Vila illustrated their method with the following example,

$$\begin{aligned} &\max\ 5x_1 + 6x_2 \\ &s.t.\ \tilde{3}x_1 + \tilde{4}x_2 \leq \tilde{18} \\ &\quad\ \ \tilde{2}x_1 + \tilde{1}x_2 \leq \tilde{7} \\ &\quad\ \ x_i \geq 0 \end{aligned} \tag{14}$$

where the fuzzy parameters were: $a_{11} = [3, 2, 4]$; a_{12}=[4, 2.5, 5.5]; $b_1 = [18, 16, 19]$; $a_{21} = [2, 1, 3]$; $a_{22} = [1, 0.5, 2]$; $b_2 = [7, 6, 9]$. These authors then construct two alternative auxiliary problems to handle the fuzzy parameters with their parametric method. In this paper we only discuss the authors solutions for the first auxiliary problem.

Our fuzzy membership functions limits are: $a_{11} = [2, 3, 4]$; a_{12} = [2.5, 4, 5.5]; $b_1 =]18, 19]$; $a_{21} = [1, 2, 3]$; $a_{22} = [0.5, 1.25, 2]$; $b_2 =]7, 9]$.

The solutions obtained were:

solutions	#	Z	x_1	x_2	μ	time
Crisp solution	1	28	2	3	—	—
Fuzzy Author τ= 0.3	2	31.22	2.14	3.42	0.7(*)	—
Fuzzy Author τ= 0.6	3	29.84	2.08	3.24	0.4(*)	—
Fuzzy Author τ= 0.9	4	28.46	2.02	3.06	0.1(*)	—
Fuzzy ours α= 0.3	5	34.36	2.581	3.576	0.3380	9”
Fuzzy ours α= 0.6	6	31.77	0.656	4.749	0.6402	9”
Fuzzy ours α= 0.9	7	29.63	1.337	3.658	0.9062	14”

(*) this corresponds to $(1 - \tau)$.

Comparing the results for the same τ and α we see that with our approach we always obtain better results than the parametric method proposed by Delgado,Verdegay and Vila. Our method is more flexible, hence it allows a wider search of space. In terms of time our algorithm performed quite well considering it has fuzzy coefficients and that solution #7 has high satisfaction value.

We also tested other examples by the same authors Delgado,Verdegay, Vila and Campos [25] [10] for their parametric method. Since their method did not change our results proved to be always better. For reasons of space we will not present these results.

3.7 Example by Zimmermann [11]

Zimmermann was the first author to propose a fuzzy method to deal with fuzzy resources. The example he presented to discuss his symmetrical method was,

$$\begin{aligned} &\min\ 41400x_1 + 44300x_2 + 48100x_3 + 49100x_4 \\ &s.t.\ 0.84x_1 + 1.44x_2 + 2.16x_3 + 2.4x_4 \leq \tilde{170} \\ &\quad 16x_1 + 16x_2 + 16x_3 + 16x_4 \leq \tilde{1300} \\ &\quad x_i \geq 0 \end{aligned} \tag{15}$$

and the fuzzy parameters limits were: $b_1 = [160, 170[$, $b_2 = [1200, 1300[$ and $b_3 = [0, 6[$.

To solve this problem with our method we do not need to make any modification in the above formulation. Because we already tested problems with only fuzzy resources in this test we only used two thresholds, 0.3 and 0.6 because they are enough to draw conclusions.

The solutions obtained were:

solutions	#	Z	x_1	x_2	x_3	x_4	μ	time
Crisp solution	1	3,864,975	6	16.29	0	66.54	—	—
Fuzzy Author	2	3,988,250	17.411	0	0	66.54	—	—
Fuzzy ours α= 0.3	3	3,701,349	4.058	11.743	2.777	58.647	0.3560	6”
Fuzzy ours α= 0.6	4	3,759,867	5.540	14.401	0.200	58.716	0.6170	6”

As Zimmmermann comments in his book his solution is not very good for this example (the results are worse than the crisp solution). With our approach the results tested (#3 and #4) are both significantly better than the crisp and Zimmermann solutions (#1 and #2). This clearly shows the flexibility of our approach compared with the symmetric method of Zimmermann. In addition the computational time to solve this problem is quite small, which means that the SA algorithm quickly obtains a good solution.

3.8 Unfeasible crisp example (without crisp solution) [12]

This following example is an unfeasible crisp problem,

$$\begin{aligned} \max\ & 40x_1 + 30x_2 \\ s.t.\ & 0.4x_1 + 0.5x_2 \leq 20 \\ & 0.2x_2 \leq 5 \\ & 0.6x_1 + 0.3x_2 \leq 21 \\ & x_1 \geq 30 \\ & x_2 \geq 15 \end{aligned} \tag{16}$$

However, when we fuzzify the resource limits or even the resource limits and the constraints coefficients we can find a feasible solution. We tested the two types of flexibilization (one test for resources and another for resources and coefficients) using the 10% tolerance from the preferred value (the given ones).

The solutions obtained for only fuzzified resources (B_i) are,

solutions	Z	x_1	x_2	B_1	B_2	B_3	B_4	B_5	μ	time
Crisp solution	No	No	No	—	—	—	—	—	—	—
Fuzzy Author	1570	28	15	—	—	—	—	—	—	—
Fuzzy ours α=0.1	1891.8	33.00	19.06	22.73	3.81	25.52	33.00	19.06	0.25	48"
Fuzzy ours α=0.3	1882.7	30.98	21.45	23.12	4.29	25.02	30.98	21.45	0.33	45"
Fuzzy ours α=0.6	1766.9	28.37	21.07	21.88	4.21	23.34	28.37	21.07	0.61	1'4"
Fuzzy ours α=0.9	1598.6	28.97	14.66	18.92	2.93	21.78	28.97	14.66	0.90	1'5"

The solutions for fuzzified resources and coefficients are,

solutions	Z	x_1	x_2	μ	time
Crisp solution	No	No	No	—	—
Fuzzy Author	1570	28	15	—	—
Fuzzy ours α=0.1	2180.05	32.61	24.26	0.14	18"
Fuzzy ours α=0.3	1999.69	33.14	19.06	0.40	18"
Fuzzy ours α=0.6	1841.28	30.22	19.79	0.62	19"
Fuzzy ours α=0.9	1621.35	29.12	14.62	0.90	20"

and the solutions for the fuzzy coefficients and fuzzy resources are now:

	A_{11}	A_{12}	C_{11}	C_{12}	C_{21}	C_{31}	C_{32}	B_1	B_2	B_3	B_4	B_5
α=0.1	43.59	32.22	0.37	0.46	0.20	0.61	0.30	22.80	6.14	24.19	24.58	30.03
α=0.3	42.13	31.65	0.39	047	0.21	0.57	0.31	21.77	3.95	24.98	33.14	19.06
α=0.6	40.84	30.70	0.39	0.51	0.21	0.58	0.29	21.74	4.06	22.30	30.22	19.79
α=0.9	40.47	30.29	0.40	0.50	0.20	0.59	0.30	19.13	2.91	21.68	29.12	14.62

The performance of the SA algorithm in this example is quite enlightening. The time to solve a more flexible problem is much smaller than when we are less flexible (just fuzzy resources). The reason is that with more flexibility we obtain many more possible solutions to choose from.

Another interesting aspect is that all solutions found in this work are better in terms of the objective function value, but in terms of violating constraint 4 (the one that is causing problems) for a small threshold we have more violation ($x_1 = 24.58$ versus $x_1 = 28$), i.e. we violate more for lower thresholds and this should not be the case. We believe the reason for this is due to the nature of the SA algorithm (sometimes it gets stuck in local optimum [2] [3]). For the other thresholds we obtain better results and less violations.

This example clearly shows the potential of using a fuzzy approach to obtain results for otherwise unfeasible problems.

4 Set of non-linear examples tested and discussion of the results

In this section we will present some non-linear examples from the literature. However, most of the examples selected are crisp and not from fuzzy authors because there are not many fuzzy methods that can handle non-linear problems. Some exceptions can be seen in the following books [14] [7, 15]. However, since most of the examples were too big, in this section the two fuzzy examples discussed are from Sakawa´s book [7].

In summary, the selected example set to be tested with our fuzzy approach is: (4.1) a non-linear peak-load pricing problem [13]; (4.2) a non-linear sales force allocation problem [13]; (4.3) and (4.4.) two non-linear fuzzy examples from Sakawa [7], one with only fuzzy goals and another with fuzzy goals and fuzzy coefficients.

For this set of non-linear problems we considered a simplification of using deviations of 10% for all resources coefficients or goals fuzzifications. I.e. the membership functions were constructed considering a fuzzification of 10% from the preferred value of the resource or coefficient or goal value to be fuzzified. In addition, for all our solutions we show the values obtained for the fuzzy parameters.

4.1 Crisp peak-load pricing example [13]

This is a crisp peak-load pricing problem that we fuzzified to assess what are the gains obtained by being flexible and how the fuzzy approach can handle non-linear problems. We tested the fuzzification of all the variables coefficients with the above mentioned 10% tolerances. The example formulation is,

$$\begin{array}{l} \max\ 60P - 0.5P^2 + 0.2FP + 40F - F^2 - 10C \\ s.t.\ 60 - 0.5P + 0.1F \leq C \\ \quad 40 + 0.1P - F \leq C \\ \quad F, P, C \geq 0 \end{array} \tag{17}$$

The solutions obtained for the three thresholds considered are:

solutions	Z	P	F	C	μ	time
Crisp solution	2202.3	70.31	26.53	27.5	—	—
Fuzzy ours α=0.1	2903.30	75.08	27.83	29.40	0.15	32"
Fuzzy ours α=0.3	2715.25	71.64	24.48	28.59	0.30	44"
Fuzzy ours α=0.6	2513.14	74.59	28.75	25.72	0.61	1'03"
Fuzzy ours α=0.9	2274.75	66.56	25.86	29.26	0.95	1'51"

And the fuzzy parameters solutions are:

	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	C_{11}	C_{12}	C_{13}	C_{21}	C_{22}	C_{23}
α=0.1	64.86	0.458	0.197	43.16	0.929	9.49	60.78	0.541	0.095	40.04	0.099	1.013
α=0.3	64.20	0.473	0.211	42.40	0.941	10.53	59.40	0.466	0.104	37.32	0.097	0.989
α=0.6	62.28	0.485	0.196	41.48	0.961	9.81	61.80	0.492	0.096	38.64	0.101	0.962
α=0.9	60.60	0.495	0.201	40.28	0.990	9.93	60.24	0.497	0.101	40.28	0.099	1.006

As can be observed, by being flexible the results improve considerably. Even considering a threshold of 90% (meaning that we only allow a violation of constraints of 10% or that we want a satisfaction for our constraints of 90%) we do obtain better results than the crisp solution, $Z = 2274.75$ versus crisp $Z = 2202.3$. All the other results for the objective function are much better than the crisp solution. It seems that the expense of considering small deviation on the coefficients pays off in term of obtaining higher profits.

If we consider a further fuzzification of the resources and/or goals we could have achieved even better results. However, we must point that these better results would be obtained at the expense of using more or less resources (depending on the constraint sign).

In terms of time to solve the problem this example is not very good for higher thresholds $\alpha = 0.6$ and 0.9 because the SA took more than one minute to solve a problem with 3 variables and 2 constraints. However, if we think it is a non-linear problem (hence more difficult to sole) the results are acceptable.

4.2 Crisp sales force allocation example [13]

This is a small sales force allocation crisp problem with only non-negativity constraints (it is a simplified version of a problem by Lodish et al in: Interfaces 18, 1 (1996): 5-20, presented in [13]). We should also point that the authors also show an integer version of the same problem, but here is not considered.

$$\begin{aligned} &\max\ 200x_1^{0.5} + 150x_2^{0.75} + 180x_3^{0.6} + 300x_4^{0.3} - 50x_1 - 50x_2 - 50x_3 - 50x_4 \\ &s.t.\ x_i \geq 0 \end{aligned} \tag{18}$$

The solutions obtained for this problem with the different thresholds are:

solutions	Z	x_1	x_2	x_3	x_4	μ	time
Crisp solution	1125.876	4	25.6	6.85	2.3	—	—
Fuzzy ours α=0.1	1551.41	5.31	34.39	11.43	1.68	0.38	8”
Fuzzy ours α=0.3	1474.81	8.40	33.17	6.66	3.91	0.43	9”
Fuzzy ours α=0.6	1458.75	4.29	29.17	8.77	8.77	0.77	10”
Fuzzy ours α=0.9	1204.00	5.54	32.11	7.54	2.58	0.90	12”

And the results for the objective function fuzzified coefficients are:

	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}
α=0.1	207.42	162.30	191.73	325.83	47.32	45.63	46.46	52.17
α=0.3	206.87	162.27	194.81	307.36	48.31	45.64	46.38	53.59
α=0.6	209.84	160.91	191.40	316.34	50.24	46.20	46.09	51.77
α=0.9	201.65	152.97	183.50	305.75	49.01	49.19	50.28	49.31

The results obtained for this example show a similar behaviour than in the previous example. All the solutions are better than the crisp solution; hence, we may say that flexibility is compensatory in terms of results. Obviously, as in the previous example, the higher the threshold (more satisfaction of constraints) the less flexible the problems are and lower results are obtained.

The curiosity of this example is that we are only fuzzifying the coefficients of the objective function and hence we can observe exactly what are the gains obtained for deviations on the coefficients (in each test with different thresholds).

Interesting enough because this example does not have constraints, besides the non-negativity ones, the time to solve the problem is quite small (maximum 12 seconds). It does show that the calculations of the constraints add a considerable weight to the SA algorithm.

4.3 Example 7.2. by Sakawa [7]

In this example of a non-linear problem the author only considered fuzzy goals,

$$\begin{aligned} \max\ F_1 &= 2x_1^2 + 4(x_2-20)^2 + 3(x_3-15)^2 \\ \min\ F_2 &= (x_1-10)^2 + 2(x_2-25)^2 + 3(x_3+5)^2 \\ \text{equal}\ F_3 &= 3(x_1+15)^2 + 2(x_2+10)^2 + (x_3+20)^2 \\ s.t.\ & (x_1+5)^2 + (x_2+8)^2 + (x_3-10)^2 \le 200 \\ & x_i \le 10 \\ & x_i \ge 10 \end{aligned} \tag{19}$$

The fuzzy goals limits, proposed by the author, were: F_1 $=]950, 2200]$; $F_2 = [1900, 1750[$; $F_3 = [1300, 1900, 2500]$.

Our solutions and the fuzzy solution of the author are:

solutions	F_1	F_2	F_3	x_1	x_2	x_3	μ	time
Crisp solution	–	–	–	–	–	–	—	—
Fuzzy Author	2063.41	1646.96	1853.06	4.3348	0.0225	3.0358	—	—
Fuzzy ours α=0.1	2171.81	1418.50	1752.27	3.9191	0.0744	1.8058	0.50	42”
Fuzzy ours α=0.3	2145.54	1439.67	1809.61	4.2278	0.0478	2.3287	0.61	40”
Fuzzy ours α=0.6	2156.05	1439.69	1800.25	4.1909	0.0148	2.2438	0.68	43”
Fuzzy ours α=0.9	2094.10	1471.61	1853.98	4.3236	0.0567	3.0570	1	52”

All the results for objective functions F_1 and F_2 that we obtained are considerably better than the author ones. Of course the less flexibility we allow the worse results we obtain (F_1 with $\alpha = 0.9$ is 2094.10 is relatively better than the author $F_1 = 2063.41$). The more intriguing case is the equality objective, F_3, where only for the less flexible, F_1 with $\alpha = 0.9$, our results are better than the author one. We believe that the reason for this is that we only considered 10% tolerance on the memberships; hence, the values close to the preferred one have bigger membership values that the ones obtained by the author.

4.4 Example 7.6 by Sakawa [7]

In this example Sakawa considered the fuzzification of the parameters that are depicted in the example formulation, as well as the goals:

$$\begin{aligned} \min\ F_1 &= (x_1+5)^2 + A_{11}x_2^2 + 2(x_3-A_{12})^2 \\ \min\ F_2 &= A_{21}(x_1-45)^2 + (x_2+15)^2 + 3(x_3+A_{22})^2 \\ \text{equal}\ F_3 &= A_{31}(x_1+20)^2 + A_{32}(x_2-45)^2 + (x_3+15)^2 \\ s.t.\ & B_1x_1 + B_2x_2 + B_3x_3 \le 100 \\ & x_i \le 10 \\ & x_i \ge 10 \end{aligned} \tag{20}$$

The fuzzy parameters limits proposed by the author were: A_{11}=[3.8, 4, 4.3], $A_{12} = [48.5, 50, 52]$, $A_{21} = [1.85, 2, 2.2]$, $A_{22} = [18.2, 20, 22.5]$, $A_{31} =$

$[2.9, 3, 3.15]$, $A_{32} = [4.7, 5, 5.35]$, $B_{11} = [0.9, 1, 1.1]$, $B_{12} = [0.8, 1, 1.2]$, $B_{13} = [0.85, 1, 1.15]$. The author also fuzzified the goals and the limits proposed are: $f1 = [5400, 3300[$; $f2 = [6900, 3900[$, $f3 = [7800, 10000, 13300]$. We must remind the readers that we simplified these deviations to 10% for all fuzzy parameters.

The solutions given by the author, as well as our solutions were:

solutions	F_1	F_2	F_3	x_1	x_2	x_3	μ	time
Crisp solution	–	–	–	–	–	–	—	—
Fuzzy Author	4816.80	4526.11	10455.02	8.177	5.878	2.258	—	—
Fuzzy ours α=0.1	3784.17	4407.86	11017.46	7.990	3.980	3.433	0.22	34”
Fuzzy ours α=0.3	4242.07	4494.45	10780.07	7.841	5.064	2.390	0.66	45”
Fuzzy ours α=0.6	4625.42	4512.46	10371.89	7.749	5.736	1.155	0.75	36”
Fuzzy ours α=0.9	4787.88	4508.31	10350.00	7.868	5.526	1.785	0.90	53”

and our solutions, found for the fuzzy parameters are:

	A_{11}	A_{12}	A_{21}	A_{31}	A_{32}	B_{11}	B_{12}	B_{13}	
α=0.1	3.919	45.584	1.923	18.272	3.221	4.846	1.064	1.069	0.956
α=0.3	4.072	46.959	1.934	19.377	3.175	5.027	1.058	1.021	1.077
α=0.6	3.865	47.715	2.036	19.320	3.028	5.046	0.973	1.057	0.950
α=0.9	4.004	49.219	1.967	19.622	2.974	4.979	1.001	1.014	1.016

As can be observed we obtained considerably better results for F_1 and F_2, for the three thresholds. Even for $\alpha = 0.9$ the results are better, particularly for $F_1 = 4787.88$ (our solution) versus $F_1 = 4816.8$ (author solution). In this example we also obtain better results for the equality objective, F_3, except for lower thresholds, $\alpha = 0.1$ and 0.3. In terms of the results obtained for the coefficients ours are quite close to the central value, given by the author, which represents the preferred value for that coefficient. This is due to a more restrict fuzzification of the coefficients (only 10% tolerance), than the author considered. However, we still obtain better results for the objective functions and close enough values for the fuzzy parameters. This was an interesting example to test because it included fuzzification of coefficients and goals.

In terms of time the SA took to solve the problem, even though it cannot be compared with other results (the solving time was not a consideration in the author proposal) it seems quite reasonable when compared with the previous example (4.3). For the more restrictive case, $\alpha = 0.9$, it took 53 seconds versus 52 seconds for the same threshold of the previous example. Further, comparing the times to solve the two examples (4.3 and 4.4) we can say that the SA solving time is more dependent on the dimension of the problem than in being more or less fuzzified.

5 Conclusions

This paper compared the results obtained by solving several examples with a fuzzy approach and with different methods proposed in the literature. Most of the results obtained with our approach are better than the ones shown by other methods. Even for some methods that showed better results, for similar satisfaction values, we could provide better results for lower satisfaction values since our method allows simulations with different thresholds.

The selected set of examples included seven linear examples, four non-linear and one unfeasible problem. We believe this set of examples provided a significant test-bed for discussion of the method used.

We also showed that the approach allows a way to study the trade-off between better objective function and worse satisfaction values for the fuzzy parameters (i.e. more violation is required) and vice versa. This characteristic, in combination with its generality, makes this method a very flexible method. We also pointed the main disadvantage of the fuzzification model used, because the flexibility and generality of the approach is gained by having a larger dimension formulation as well as a non-linear one.

Finally, we showed that using the simulated annealing algorithm for solving fuzzy optimization problems is a good solution technique for solving this type of problems. However, the implementation needs further improvements to allow using the full potential of multiple objective fuzzy optimization problems as well as other types of membership functions.

References

1. Pires, F.M., J. Pires, Moura, and R. Ribeiro (1996) Solving Fuzzy Optimisation problems: Flexible Approaches using Simulated Annealing. In Proceedings of the World Automation Congress (WAC´96). Montpelier, France: TSI Press series.
2. Ribeiro, R.A. and F. Moura-Pires (1999), Fuzzy Linear Programming Via Simulated Annealing. Kybernetica, 35(1):57–67.
3. Varela, L.R. and R.A. Ribeiro (2001), Utilização de Simulated Annealing em Optimização Difusa (in Portuguese). Investigação Operacional, 21(2):205–231.
4. Carlsson, C. and P. Korhonen (1986) A Parametric Approach to Fuzzy Linear Programming. Fuzzy Sets and Systems, **20**:17–30.
5. Chanas, S. (1983) The Use of Parametric Programming in Fuzzy Linear Programming. Fuzzy Sets and Systems, **11**:243–251.
6. Lai, Y.-J. and C.-L. Hwang (1992) Fuzzy Mathematical Programming: Methods and Applications. Lecture Notes in Economics and Mathematical Systems. Vol. 39. Springer-Verlag.
7. Sakawa, M. (1993) Fuzzy Sets and Interactive Multiobjective Optimization. In: Applied Information Technology. M.G. Singh (Ed). Plenum Press.
8. Tanaka, H. and K. Asai (1984)Fuzzy Linear Programming Problems with Fuzzy Numbers. Fuzzy Sets and Systems, **13**:1–10.
9. Delgado, M., J.L. Verdegay, and M.A. Vila (1989) A General Model for Fuzzy Linear Programming. Fuzzy Sets and Systems, **29**:21–29.

10. Delgado, M., J.L. Verdegay, and M.A. Vila (1990) Relating Different Approaches to Solve Linear Programming Problems With Imprecise Costs. Fuzzy Sets and Systems, **37**:33–42.
11. Zimmermann, H.-J. (1996) Fuzzy Set Theory and its Applications. 3rd edition. Kluwer Academic publisher.
12. Pires, F.M. and R.A. Ribeiro(1998) A New Risk Function for Fuzzy Linear Programming. In: Proceedings of the World Automation Congress (WAC´98),Hawaii. TSI Press.
13. Winston, W.L. and S.C. Albright (1997) Pratical Management Science: Spreadsheet Modeling Applications. Duxbury Press- International Tomhson Publishing Inc.
14. Lai, Y.-J. and C.-L. Hwang (1994) Fuzzy Multiple Objective Decision Making. Lectures Notes in Economics and Mathematical Systems. Springer-Verlag.
15. M. Delgado, J. Kacprzyk, J.L. Verdegay and M.A. Vila (Eds). (1994) Fuzzy Optimization. Studies in Fuzziness. Physica-Verlag.
16. Bellman, R.E. and L.A. Zadeh (1970) Decision-Making in a Fuzzy Environment. Management Science, Vol.17(4) :141–164.
17. Zimmermann, H.-J. (1978) Fuzzy Programming and Linear Programming with Several Objective Functions. Fuzzy Sets and Systems, **1**:45–55.
18. Kickert, W.J.M. (1978) Fuzzy Theories on Decision Making. Frontiers in Systems Research, Vol 3. Martinus Nijhoff Social Sciences Division.
19. Fedrizzi, M., J. Kacprzyk, and J.L. Verdegay, (1991). A Survey of Fuzzy Optimization and Mathematical Programming. In: Interactive Fuzzy Optimization, M. Fedrizzi, J. Kacprzyk, and M. Roubens (Eds). Springer-Verlag.
20. Delgado, M., J.L. Verdegay, and M.A. Vila (1994) Fuzzy Linear Programming: From Classical Methods to New Applications. In: Fuzzy Optimization, M. Delgado, J. Kacprzyk, J.L. Verdegay and M.A. Vila (Eds). Physica-Verlag. :111–134.
21. Ribeiro, R.A. and F.M. Pires (1999), Fuzzy Site Location Problems and Simulated Annealing. Series Studies in Locational Analysis, 13(June):61–76.
22. Reeves, C.R., Ed. (1995) Modern Heuristic Techniques for Combinatorial Problems. Advanced Topics in Computer Science. McGraw Hill.
23. Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi (1983) Optimization by Simulated Annealing. Science, Vol. 220(4598):671–680.
24. Tanaka, H., H. Ichihashi, and K. Asai (1985) Fuzzy Decision In Linear Programming Problems With Trapezoid Fuzzy Parameters. In: Management Decision Support Systems Using Fuzzy Sets and Possibility Theory, J. Kacprzyk and R. Yager (Eds), Verlag TUV Rheinland. :146–155.
25. Campos, L. and J.L. Verdegay (1989) Linear Programming Problems and Ranking of Fuzzy Numbers. Fuzzy Sets and Systems, **32**:1–11.