Fuzzy Linear Programming via Simulated Annealing

Rita Almeida Ribeiro
Universidade Nova Lisboa
FCT, Dept. Informatica
2825 Monte Caparica
Portugal
e-mail: rr@di.fct.unl.pt

Fernando Moura Pires
Universidade Nova Lisboa
FCT, Dept. Informatica
2825 Monte Caparica
Portugal
e-mail: fmp@di.fct.unl.pt

Abstract

This paper shows how the simulated annealing (SA) algorithm provides a simple tool for solving fuzzy optimization problems. Often, the issue is not so much how to fuzzify or remove the conceptual imprecision, but which tools enable simple solutions for these intrinsically uncertain problems. A well-known linear programming example is used to discuss the suitability of the SA algorithm for solving fuzzy optimization problems.

Keywords: fuzzy optimization, simulated annealing, fuzzy linear programming, fuzzy multiple objective decision making.

1. Introduction

This paper shows how the simulated annealing (SA) algorithm provides an innate method for solving fuzzy optimization problems. Traditional optimization models have to be constructed crisp and unambiguous, characteristics achieved through reduction and assumptions from the ill-structured reality. Fuzzy optimization has been focusing, first in solving models which reflect real life uncertainty, and second on transforming them into equivalent crisp problems to benefit from efficient existing solving algorithms. This second stage of removing the conceptual fuzziness of the problem also removes the real-life vagueness and fuzziness of the human reasoning process. As Zeleny's states "humans do not maximise functions, but search for recognisable patterns" [11]. It is important to have methods capable of directly handling all types of fuzzy optimal problems.

Simulated annealing is a stochastic algorithm with a physical analogy of "melting" the system being optimised (a solid) using "high temperatures", and then proceeding by slowly lowering the temperature until the system "crystallises/freezes" and no more changes occur [7]. The melting process can be viewed as stretching the constraints of an optimization problem which are a direct result of their fuzzification, and the crystallisation as the search for the "best" solution. The simulated annealing is considered a good tool for solving crisp optimization problems that are NP-complete [7]. Here it is discussed its extension to fuzzy optimization problems.

The focus is on fuzzy linear optimization problems and, specifically, on problems with a single objective. The first method for solving fuzzy linear programming problems was proposed by Zimmermann [12]. The author first, constructs a crisp model of the problem; second, obtains its crisp results using an existing solving algorithm; third, uses the results obtained to fuzzify the problem by considering subjective constants of admissible violations for the goal aspiration level and for the constraints; fourth, defines an equivalent crisp problem using an auxiliary variable that represents the maximisation of the minimisation of the deviations (violations) on the constraints. Zimmermann used Bellman and Zadeh´s [1] interpretation that a fuzzy decision is a confluence of goals and constraints, denoted the max-min model because it considers that the best fuzzy decision is the union of the aggregated intersections of goal and constraints. Since this approach is well-known, Zimmermann´s example [13] of deciding on the size and structure of a truck fleet is selected, in order to compare the results obtained with the simulated annealing algorithm. This paper also emphasises that the generality and independence of the SA algorithm makes it well-suited for dealing with other forms of fuzziness found in optimization problems.

The paper is organised in five sections. First this introduction, second a section introducing the main concepts involved in fuzzy linear programming optimization and third a section describes the simulated annealing (SA) algorithm as well as its modification for handling these type of problems. The fourth section discusses the comparison of results obtained with the max-min approach and with the SA algorithm and the fifth section presents the conclusions of this work and future developments.

2. Fuzzy Optimisation

Traditional optimization is associated with problems of maximising or minimising a utility function, subject to constraints representing limited resources. The aim is to maximise or minimise an objective function while satisfying the problem constraints. Essentially, traditional optimal concepts such as,

Max f(x) subject to x ∈ X

are based on unique solutions and existence of complete information. An illustration of this problem is the linear programming model.

Fuzzy optimization's main aim is to find the "best" solution (decision alternative) under incomplete information, i.e. imprecise information and/or vague resources limits. There are many forms of imprecision when dealing with fuzzy optimization. For instance, coefficient variables that are not known precisely (e.g. production time of about 2 hours to make a shirt) or constraints satisfaction that should be around a limit (e.g. total available production time should be around 200 hours). The current challenge is to enable the construction of models using everyday imprecise and vague language that can be translated into a fuzzy quantitative method which can provide the best solution possible.

The concepts of fuzzy goal and fuzzy constraints were first introduced by Bellman and Zadeh [1]. The authors state that a fuzzy decision can be viewed as the intersection of fuzzy goal(s) and the problem constraints - since they are all defined as fuzzy sets in the space of alternatives. The "optimal" decision is the point at which the intersection of fuzzy goal(s) and constraints take the maximum membership value. This method is usually called the max-min approach. A systematic description and classification of problem types, methods and approaches proposed in the literature can be seen in [8].

In general the fuzzification of the linear programming model includes six forms of imprecision as depicted in table 1 (more subtle distinctions are made in [8]).

This paper discusses the first three cases of imprecision using Zimmermann's example of the truck fleet [13] and solves it with the simulated annealing algorithm. To cope with the imprecise information three approaches are followed. First, maximisation/minimisation of the objective function subject to fuzzy constraints. Second, consideration of fuzzy constraints as well as a fuzzy goal. Third, maximisation of the aggregated deviations from the boundaries of the fuzzy goal and fuzzy constraints, represented by the membership values of the goal and constraints inequalities satisfaction. It was also tested the imposition of a threshold to reflect a satisfactory constraint level, which is equivalent to an alpha-cut on the function. The third approach described is identical to the one proposed by Zimmermann, hence this is tested and the results obtained with the SA algorithm and with Zimmermann solution procedure are compared. The fuzzification of cases 4), 5) and 6) is not discussed in detail here.

Table 1. Modes of Imprecision in Fuzzy Optimisation

Case 1	Imprecision in the constraints boundaries. This implies the fuzzification of the inequalities limits, **C**. E.g. "the total time of painting should be *considerably fewer* than 100 hours ($\mathbf{c}_j$ **is** $\tilde{\leq} 100$)".
Case 2	A fuzzy goal is imposed on the objective function. Essentially this implies fuzzifying the utility function by considering a limiting goal (similar to goal programming). E.g. "the total budget for the project should be kept *well below* 100,000 ECU ($\mathbf{Z} \tilde{\leq} 100,000$)".
Case 3	Compound imprecision. This implies combinations of the above sources of imprecision.
Case 4	The parameters (coefficients) of the variables of the constraints are not known precisely. This means that the coefficients are fuzzy numbers. E.g. "the cost per hour of producing a shirt (**x**) *is around* 10 ECU ($\mathbf{a}_{ij} \tilde{=} 20$)".
Case 5	The coefficients of the variables in the objective function are not known precisely. This means that coefficients are fuzzy numbers such as in 4). E.g. "the selling price of product **x** is around 100 ECUs per item".
Case 6	All possible combinations of uncertainty.

Considering the first three types of fuzzification, the general conceptual process for any optimization problem is:

1) Depending on the type of problem, formalise it as a linear programming problem or as a multiple objective problem, or even as a non-linear programming problem.
2) If the intention is to fuzzify of the objective(s), define the goal(s) for those objective(s).
3) Define the membership functions for representing the fuzzification of each constraint. E.g. triangular, sinusoid, trapezoidal and so forth.
4) Define thresholds for the degree of acceptance of deviations on the constraints satisfaction.
5) Define the aggregation operator to combine the

constraints (and goals when dealing with the symmetric model) as for example any t-norm.

6) Solve with the simulated annealing algorithm.

In addition it should be noted that if the symmetric model of Bellman and Zadeh [1] is used, there is no difference between objectives and constraints in the problem model, as well as no difference between single or multiple objectives. With the symmetric model a mathematical programming problem becomes a constraint satisfaction problem, where a decision is a confluence of constraints and objectives:
Find **x** such that

$$\sum_j g_j x_j \left\{\tilde{\geq}, \tilde{\leq}, \tilde{=}\right\} Z$$

$$\sum_j a_{ij} x_j \left\{\tilde{\geq}, \tilde{\leq}, \tilde{=}\right\} c_i$$

$$x_j \geq 0$$

Hence, the maxmin model of the fuzzy problem is:

$$\max \left[\min_k \mu_k(x) \right]$$

where $\mu_k(x)$ is the membership values of the goal and constraints satisfaction.

The problem of embedding fuzziness, where necessary, to deal with imprecision leads to the consideration of an extension of the fuzzy optimization problem to accept fuzzy coefficients. The normalisation of the combination of the fuzzy constraints and objective(s) with the fuzzy variables coefficients leads to the symbolical robust fuzzy model:

$$\sum_i \tilde{y}_k \, x_i \left\{\tilde{\leq}\tilde{\geq}\tilde{=}\right\} c_j$$

$$\sum_k y_k \tilde{=} a_k \qquad \text{where } a_k \text{ are fuzzy numbers}$$

With this specification and using the same solution procedure, the results obtained with the SA algorithm are also successful [9]. Since it is beyond the scope of this paper to address fuzzy coefficients no further details are discussed here.

Most of the literature on fuzzy optimization is concerned with fuzziness at the modelling level, from goal preferences to goal priorities. The majority of authors follow the maxmin approach, using the equivalent crisp model of selection of the best ("max"), which represents the aggregation of the minimum deviations from the model levels (e.g. Z). In summary, they do not propose a fuzzy solution but an equivalent crisp one (overview in [8]). However, some attempts were made to spread the benefits of using algorithms appropriate to solve fuzzy optimization problems, such as simulated annealing (see for example [5]).

3. Simulated Annealing

Simulated annealing is a stochastic algorithm used for optimization problems where the objective function corresponds to the energy of the states of a solid [4]. The SA algorithm requires the definition of the neighbourhood structure as well as the parameters for the cooling schedule. The temperature parameter distinguishes between large and small changes in the objective function. Large changes occur at high temperatures and small changes at low temperatures. It is an evolutionary process moving in small steps from one stage to another, avoiding the problem of getting stuck in a local minimum by allowing uphill and downhill moves for the temperature.

SA is a good tool for solving optimization problems considered NP-complete, i.e. computationally inefficient since the search for the optimum is an exponential function of the size of the problem [7]. Some good examples on the capabilities of the simulated annealing algorithm to solve optimization problems can be found in [2, 3, 4, 7].

According to [7] the four basic requirements for using the SA algorithm in combinatorial optimization problems are: (a) concise description of the problem; (b) random generation of the changes from one configuration to another; (c) an objective function containing the utility function of the trade-offs; (d) the initial state, the number of iterations to be performed at each temperature and its annealing scheme (for a detailed discussion of the algorithm see [4, 7]). When discussing fuzzy optimization none of the basic requirements are affected because the fuzziness is usually expressed either in the stretching of constraints or as fuzzy coefficients in the variables.

Two implementations of the SA algorithm have been developed. One maximises the aggregation of the tolerance intervals (membership values of the goals and constraints), i.e. the fuzzification of both the objective and constraints (case 2 and 3 of table 1). The other only handles the fuzzification of constraints (case 1, table 1).

The objective of the first implementation of the simulated annealing algorithm is to solve the maxmin model by maximising the aggregation of the membership values of the goals and constraints. This solution procedure is equivalent to Zimmermann´s

proposal [12] Its implementation is defined in pseudo-code 1.

Pseudo-code 1. Simulated Annealing maxmin

```
Set Nr = number of constraints;
Select a initial state x ∈ X;
Select a initial temperature T > 0;
Set temperature change counter t = 0;
For k := 1 to Nr do μ(k) = membership value of the
constraint Rk(x);
Miu1= aggregation(μ(1), ..., μ(Nr));
Repeat
    Set repetition counter n =0;
    Repeat
        Generate state y, a neighbour of x;
        k := 1
        Repeat
            μ(k) = membership value of the constraint
                   Rk(y);
            k := k+1;
        until k > Nr or μ(k-1) = 0;
        If μ(k-1) ≠ 0
            then
                Miu2 = aggregation (μ(1), ..., μ(Nr));
                Calculate δ = Miu2 - Miu1;
                If δ > 0 then x:= y; Miu1 := Miu2
                    else If random(0,1) < exp(δ/T) then
                            x:= y; Miu1 := Miu2
        n := n+1
    until n = N(t)
    t := t + 1
    T = T(t)
until stopping criterion true.
```

The variable k is a counter for the number of constraints. It is used to calculate the memberships of each constraint μ(k). Miu represents the aggregation of the memberships of the fuzzy constraints. Any aggregation operator could have been considered, here the t-norm min is used. The variable δ depicts the difference between the previous the new Miu and the last Miu obtained. This represents the old and new energy states difference. The probability function of moving to a smaller energy state (the goal) is given by the exponential of δ divided by the control parameter T. The smaller the temperature T the less probable any change will occur. The N(t) represents the number of neighbours generated as possible solutions to be tested. The T(t) is the decreasing function of the temperature. Usually, this decreasing factor is around 0.9.

The second implementation of the SA is similar except that the δ is obtained by the difference between the new and old values for the objective function value and some other small details. Hence, we will not detail this implementation any further.

The membership functions of the fuzzy goal and constraints used in the two implementations are triangular functions. This choice was based on the necessity of obtaining results compatible with Zimmermann's example. Any other membership function (e.g. sinusoidal) could have been used since there are no limitations of linearity when using the SA algorithm. For example, for the case $\tilde{\leq}$ the the function is:

$$\mu_k(\mathbf{x}) \begin{cases} 0 & \mathbf{R_k(x) > c_k + d_k} \\ 1 - \dfrac{\mathbf{R_k(x) + c_k}}{\mathbf{d_k}} & \mathbf{c_k < R_k(x) \leq c_k + d_k} \\ 1 & \mathbf{R_k \leq c_k} \end{cases}$$

where $\mathbf{R_k} = \sum_j \mathbf{a_{kj} x_j}$ or $\sum_j \mathbf{g_j x_j}$ and $\mathbf{d_k}$ are the deviations from the crisp value.

In summary, the main advantages of using simulated annealing (SA) for solving fuzzy optimization problems, as can be observed in both implementations, are (a) its convenience and easy development; (b) its flexibility regarding fuzzy linear or non-linear problems (see example in [10]); (c) its easy modelling of the problem because no rigid structure is required; (d) and it does not require mathematical modifications on the problem for reasons of equivalence or others. The main disadvantage of the simulated annealing algorithm is the need to define the initial states that satisfy the constraints, for each variable. Another disadvantage is the tuning of the temperature because it can imply a bigger search and consequently more computing time.

4. Example of fuzzy linear optimization

4.1. Introduction

Zimmermann's example [13] is selected because it is a standard linear programming problem well-known and discussed in the literature [6, 8]. The crisp linear programming example, to be tested with the SA algorithm is:

$$\begin{aligned} \textbf{Min} \quad & 41400\mathbf{x}_1 + 44300\mathbf{x}_2 + 48100\mathbf{x}_3 + 49100\mathbf{x}_4 \\ \textbf{s.t.} \quad & 0.84\mathbf{x}_1 + 1.44\mathbf{x}_2 + 2.16\mathbf{x}_3 + 2.4\mathbf{x}_4 \geq 170 \\ & 16\mathbf{x}_1 + 16\mathbf{x}_2 + 16\mathbf{x}_3 + 16\mathbf{x}_4 \geq 1300 \\ & \mathbf{x}_1 \geq 6 \\ & \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4 \geq 0 \end{aligned}$$

The crisp solution presented by Zimmermann is:

$$\textbf{Crisp solution: } \mathbf{x}_1 = 6; \mathbf{x}_2 = 16.29; \mathbf{x}_3 = 0; \mathbf{x}_4 = 58.96; \quad \mathbf{Z} = 3,864,983.$$

After obtaining the crisp results Zimmermann fuzzifies the problem, considering the following tolerance intervals:

Lower Bound	Deviation	Upper bound (b+d)
Z=b1= 3,700,000	d1= 500,000	4,200,000
c2= 170	d2= 10	180
c3= 1,300	d3= 100	1,400
c4= 6	d4= 6	12

In order to solve the problem the symmetric model of Bellman and Zadeh [1] is assumed, which considers that in a fuzzy environment there is no difference between goals and constraints. Within this model the objective moves into a constraint with a defined stretched boundary such as any other fuzzy constraint. The resolution process proceeds to transform the problem into a crisp one, using for objective the maximisation of the minimisation of the membership values of the constraints deviations, determined with functions similar to the ones presented on point 3. After, Zimmermann transforms the fuzzy problem into a crisp equivalent one (see details in [13]) using an additional variable L, which represents the maximisation of the minimisation of the deviations, in the sense of the maxmin model.

In order to compare Zimmermann results with the ones given by the simulated annealing algorithm the min and max operators were used. The results comparison of the two methods are depicted in Table 2 and Table 3 and discussed in section 4.3.

4.2. SA Implementation Choices

In the implementation of the SA algorithm it is necessary to choose the following parameters: how to generate a state y a neighbour of x; the aggregation function - Miu; the number of neighbour generated - N(t); the decreasing temperature function - T(t); and finally the stopping criterion.

The choice of how to generate a state **y** as a neighbour of **x**, is done by defining a new state which is a random point **y** where the distance to the point **x** is random and less than t, defined by

$$\rho(\mathbf{t}) = \begin{cases} 1 & \mathbf{t} < 100 \\ 2 & 100 \le \mathbf{t} < 150 \\ 3 & 150 \le \mathbf{t} < 250 \\ 5 & 250 \le \mathbf{t} < 350 \\ 15 & \mathbf{t} \ge 350 \end{cases}$$

The aggregation function selected is the intersection, and the operator used in this implementation is the t-norm min. The intersection represents the logical "and" to signify that all the constraints must be satisfied. As defined by Bellman and Zadeh [1] a decision is represented by the confluence of goals and constraints (intersection) and the best decision is the one with the maximum value.

The choice for number of neighbours to be generated, N(t), follows the heuristic of generating 200 neighbours, if $t < 400$, and generating 250, if $t > 400$. This heuristic takes in account that more sons should be generated when the temperature, T, decreases to have more options to test. The temperature's function T(t), is a function which uses a decreasing factor of 0.99. The stopping criterion used for the two implementation is reached when the temperature is less than 0.0001.

4.3. Results Discussion

First, we solved the crisp problem with the simplex algorithm and the SA. The results obtained for the objective function Z are equivalent, though with different solution for the variables because there are multiple solutions for the crisp problem. Secondly, we tested the fuzzy Zimmermann approach with the fuzzy SA approach. Table 3 depicts the results for the fuzzy solutions.

The fuzzy results obtained for the Maxmin approach with the SA algorithm are also equivalent to the ones obtained with the Zimmermann method (table 3), considering the same aspect of having multiple solutions for the same Z. Table 3 also depicts two other results obtained with the second implementation of the SA (case 1 of table 1). For the first test it is set a threshold of 0.43 to compare with the aggregated value obtained in the Zimmermann solution. For the second test it is used a threshold of 0.6 to verify the quality of the solution if the contraints are less violated. The results of the constraints violations are depicted in table 4.

Comparing the results for the two implementations of the SA (case 1 and 3 of table 1), the fuzzy-constraints with $\mu > 0.43$ and the Maxmin, it can be inferred that the objective function results (respectively Z= 3,986,458 and Z=3,987,394) and the deviations for the constraints limits are quite close, except for constraint number C3. However, the implementation that considers degrees of satisfaction for its constraints is less demanding on the user since it does not require to pre-define a goal for the objective function and, moreover, a solution which ensures a degree of satisfaction for its constraints seems more reliable.

The second test performed with the SA (Case 1, table 1) using a better satisfaction degree for its constraints (μ>0.60) presents a worse result for the objective function (Z=4,035,284) but still well below the goal limit of the 4,200,000 and has the advantage of ensuring an overall degree for the constraints satisfaction of 60%.

Table 3 - Problem Results

	Zimmer-mann	Simulated Annealing Approach		
Vari-ables	Maxmin (Case 3)	Maxmin (Case 3)	Fuzzy-Constraint (Case 1) (μ>0.43)	Fuzzy-Constraint (Case 1) (μ>0.60)
Z	3,988,250	3,986,458	3,987,394	4,035,284
x1	17.41	13.98	16.83	17.28
x2	0	5.16	0.79	1.06
x3	0	1.5	0.56	0.1
x4	66.54	63.28	65.76	66.568
C1	174.3	174.29	174.3	176
C2	1343.3	1342.71	1343	1360
C3	17.4	15.213	16.83	17.28

Table 4 - Miu Results

	Simulated Annealing Approach		
Variables	Maxmin (Case 3)	Fuzzy-Constraint (Case 1) (μ>0.43)	Fuzzy-Constraint (Case 1) (μ>0.60)
Z	0.43		
C1	0.43	0.43	0.6
C2	0.43	0.43	0.6
C3	1	1	1
Miu*	0.43		

Miu* is the ggregated (intersection) memberships of the constraints and goals for Maxmin results.

Observing now table 4, we can see that the SA provides several advantages regarding the Zimmermann approach. First, it tell us all the contraint violations and not only the aggregated value. Second, if it is used the implementation where the thresholds can be defined, the decisor can select the degree of satisfaction for his/her constraint, i.e. the degree of violation allowed.

In summary, the use of SA seems more flexible and adaptable to fuzzy optimization problems than the crisp mathematical transformations required in Zimmermann method. The implementation of only fuzzyfying the constraints and leaving the objective function crisp, has the advantage of not having to stipulate a goal and the respective deviation. Many times it is quite dificult for a decisor to set a goal, since what he wants is the maximum or a minimum for his/her problem. The main disadvantage of the SA approach regarding the Zimmermann one is the last ensures the optimum while the former gives the best approximate result.

5. Conclusions

This research contributes to the quest of improving fuzzy decision support models. As Zeleny rightly states: "the question is no longer how to formalise, reduce or remove the conceptual imprecision and fuzziness (the 'crisp' treatment of 'fuzzy' systems), but how to enhance, amplify and utilise natural vagueness and fuzziness to reflect the purposes of human communication, co-operation and knowledge production." [11].

This paper shows that the simulated annealing algorithm is a good candidate tool for solving fuzzy linear optimization problems without requiring mathematical reductions or transformations. A fuzzy linear example is compared with Zimmermann's first fuzzy approach and the simulated annealing one, in order to highlight their main differences.

In a future work the authors are planning to solve fuzzy optimization problems with genetic algorithms to evaluate the performance of the simulated annealing algorithm. Another interesting comparison, under consideration, is to evaluate the results obtained with the SA and the Tabu search algorithm.

References

1. Bellman, R. E. and L. A. Zadeh (1970) Decision-Making in a Fuzzy Environment. Management Science. Vol.17(No.4) 141-164.
2. Connoly, D. (1992) General Purpose Simulated Annealing. Journal of the Operational Research Society. 43(5). 495-505.
3. Connoly, D. T. (1990) An Improved Annealing Scheme for the QAP. European Journal of Operational Research. 46. 93-100.
4. Eglese, R. W. (1990) Simulated Annealing : A Tool for Operational Research. European Journal of Operations Research. 46. 271-281.
5. Ishibuchi, H., H. Tanaka and S. Misaki (1994) "Fuzzy Flow Shop Scheduling by Simulated

Annealing." Fuzzy Optimization. M. Delgado ed. Physica-Verlag.

6. Kickert, W. J. M. (1978) "Fuzzy Theories on Decision Making." Frontiers in Systems Research, Vol 3. Martinus Nijhoff Social Sciences Division.
7. Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi (1983) Optimization by Simulated Annealing. Science. Vol. 220(4598) 671-680.
8. Lai, Y.-J. and C.-L. Hwang (1994) "Fuzzy Multiple Objective Decision Making." Lectures Notes in Economics and Mathematical Systems. Springer-Verlag.
9. Pires, F. M., J. Pires Moura and R. Ribeiro (1996) Solving Fuzzy Optimisation problems: Flexible Approaches using Simmulated Annealing. ISSCI'96. Montpelier
10. Ribeiro, R. A. and F. M. Pires (1995) "Fuzzy Site Location problems and Simulated Annealing." Series Studies in Locational Analysis. Boffey ed. (to appear).
11. Zeleny, M. (1994) "Fuzziness, Knowledge and Optimization: New Optimality Concepts." Fuzzy Optimization: Recent Advances. Delgado, Kacprzyk, Verdegay and Vila ed.
12. Zimmermann, H.-J. (1978) Fuzzy Programming and Linear Programming with Several Objective Functions. Fuzzy Sets and Systems. 1. 45-55.
13. Zimmermann, H.-J. (1986) "Fuzzy Sets Theory - and its Applications." International Series In Management Science/Operations. Ignizio ed. Kluwer-Nijhoff Plublishing. Boston.