Computer Science Journal of Moldova, vol.10, no.1(28), 2002

Fuzzy Sets-based Control Rules for Terminating Algorithms

José L. VERDEGAY, Edmundo VERGARA-MORENO

Abstract

In this paper some problems arising in the interface between two different areas, Decision Support Systems and Fuzzy Sets and Systems, are considered. The Model-Base Management System of a Decision Support System which involves some fuzziness is considered, and in that context the questions on the management of the fuzziness in some optimisation models, and then of using fuzzy rules for terminating conventional algorithms are presented, discussed and analyzed. Finally, for the concrete case of the Travelling Salesman Problem, and as an illustration of determination, management and using the fuzzy rules, a new algorithm easy to implement in the Model-Base Management System of any oriented Decision Support System is shown.

Keywords: Decision Support System, Fuzzy Rules, Travelling Salesman Problem, Algorithm.

1 Introduction

From a broad point of view the interface between two different areas, Decision Support Systems and Fuzzy Sets and Systems, is considered in this paper. On the one hand, the term Decision Support System (DSS) was coined at the beginning of the 70s to feature the computer programs that could support a user in making decisions when facing ill-structured problems. Nowadays, software for supporting decision-making is available for almost any management problem. On the other hand, in the early sixties, based on the fact that classical logic does not reflect, to the extent that it should, the omnipresent imprecision in the real world, L. A. Zadeh proposed the Theory of Fuzzy Sets and Fuzzy Logic. Nowadays Fuzzy Logic is employed with great success in the conception, design, construction and utilisation of a wide range of products and systems whose functioning is directly based on the ways human beings reason.

But, in spite of the high levels achieved in these two fields, there is a gap in the interface between them, i.e., Fuzzy Logic-Based DSS has been not so widely exploited, although in this interface context there are a number of important problems to be solved. Among them, and for the sake of the interest that for the authors of this paper this kind of problems exists, the following questions are to be pointed out: the practical determination of membership functions, which will be not considered here (interested readers are invited to consult 3,7,8,12,13,15,16,17,19), the numerical accuracy need in that fuzzy environment, the use of new algorithms to find good enough solutions, instead of optimal ones, etc. Consequently the primary aim of this paper is to describe briefly some of these problems as well as their solution ways in a DSS framework, in order to bridge the above mentioned gap and help to produce more effective and friendship Fuzzy Logic-based DSS.

In order to structure the contents of the paper, let us consider first of all the concept of DSS. In a very general meaning a DSS is a system that supports technological and managerial decision making by assisting in the organisation of knowledge about ill-structured, semi-structured or unstructured issues, and its main components are a Data-Base Management System, a Model-Base Management System and a Dialog Generation and Management System [10]. Consequently DSS are especially useful in areas providing problems with an unfamiliar structure, and much more specifically, in situations where the knowledge available from experts has a human nature, i.e., it is either imprecise or vague. At this juncture, as is well known, the Theory of Fuzzy Sets and Fuzzy Logic is the most appropriate tool for dealing with this kind of lack of precision. Amongst the three components described for a DSS, all with an equal level of importance, for the sake of convenience here only the second one will be considered. Therefore as follows, the Model-Base Management System (MBMS) will be focused on, and an imprecise framework assumed.

Consequently, in Section 2 the management of the corresponding and omnipresent fuzziness in the existing models in the MBMS is to be considered. Among all the possible development areas that could be approached, only that one considering optimisation problems, in particular Linear Programming (LP) problems, will be discussed. The reason for this is twofold. On the one hand, it is well known that in the framework of DSS, LP provides a very powerful context that has been used in a range of different applications, and has a documented history of success [14]. On the other hand, LP problems assuming fuzzy parameters, i.e., Fuzzy Linear Programming (FLP) problems is one of the best studied topics in the field of Fuzzy Sets and Fuzzy Logic [2, 5]. Finally, as a matter of illustration showing the relevance of the presented way to manage the fuzziness in the previous section, and bridging the announced gap between DSS and Fuzzy Sets and Systems, in Section 3 the Travelling Salesman Problem will be considered, and a new algorithm proving the efficiency of using fuzzy rules as termination criteria and being able of implementation in some oriented DSS will be provided.

2 Managing the lack of precision in the MBMS

It is absolutely clear that in real applications, and hence in the MBMS of any DSS, the perfect knowledge of the exact data taking part into the involved models is almost impossible, and then it is usual to approximate those values, data and/or models in different ways: for instance by using heuristic algorithms instead of conventional and experienced ones. From this last point of view amongst the many reasons that might justify the use of FLP in DSS [1, 13], and more specifically in the MBMS, here we are going to concentrate on the following two:

1) Because FLP is useful for accurately modelling the inherent vagueness in the data which the user often has available, and

2) Because it may help to find solutions for problems in which to find an optimum solution is not easy.

The situation that we envisage, referring to the first aspect, involves a decision-maker who has a DSS, and who faces the need to solve a given LP problem with the pertinent MBMS, for which purpose he must provide the numerical data with which he is going to attempt to find the solution to the problem described. As it was told, only on very few occasions will the decision-maker know precisely all the readings for the parameters that he needs for the DSS to work. Generally the information, that the decision-maker has, corresponds rather to information of an imprecise nature, which he has to adapt to the special characteristics of the system that he is using. Specifically, to illustrate this, let's assume that the problem, that is being dealt with, is as simple as the following one:

$$\max \{x_1 + 2x_2 / x_1 + 5x_2 \leq 35;\ x_1 + x_2 \leq b;\ 2x_1 + x_2 \leq 17;$$

$$3x_1 + x_2 \leq 24;\ x_1, x_2 \geq 0\}$$

Let's assume that the real, and exact value of the parameter b is $b = 11.1$. It is clear that in this case the optimum solution for the problem above would be: $x_1 = 4.775$ and $x_2 = 6.225$, with a value for the target function of $z^1 = 17.225$.

Let's admit, nevertheless, that what the decision-maker knows about that parameter b is that it has a value very close to 11, but never less than 11. The normal trend with a view of obtaining a solution for this problem means that the value that is given to b is 11, therefore using the process of "rounding off", since generally the reason is that there is no way to represent a value very close to 11 and never less than it, if that piece of data is not accompanied by a probability distribution. But by acting this way, the nature of the problem is altered, because it jumps from one problem that, due to the very nature of the information held by the decision-maker, it has a clearly imprecise approach, and therefore it is fuzzy, to another problem of a conventional nature, i.e., it is precise in the values of its coefficients.

But the treatment given to the data here, by modifying its real value to make it easier to match the data to the model that may be used, (i.e. the one that is available in the MBMS, though making it simpler to solve the problem) it may really cause serious problems, since it may lead us to propose the application of solution that are very far removed from the authentic, optimum policies that should correspond to the problem in question, if it had been approached in its original fuzzy terms. It is clear that such action may lead to serious consequences depending on the context that is being dealt with (economics, health, etc..).

In particular, in the example that is being used to illustrate this argument, the fact of forcing parameter b to take the value $b = 11$, means that the optimum solution to the problem would be: $x_1 = 5$ and $x_2 = 6$, with a value for the target function of $z^2 = 17$, which is clearly much lower than (it is really just a question of scales) the one obtained for the real problem.

The models and techniques offered by the FLP allow these miss-function to be solved without any difficulty. In fact, the modelling of the imprecision in the values of the parameters may be approached from the viewpoint that the latter are fuzzy numbers. In this sense, regardless of the wide range of different models that may be considered for implementing in an MBMS (fuzzy constraints, fuzzy objectives, etc.) the above mentioned problem could be dealt with using the following model:

$$\max\{cx/A^f x \leq_f b^f, x \geq 0\}$$

where A^f an b^f refer to the fact that we are considering fuzzy numbers in the coefficients that define the restrictions (thereby allowing, as a trivial case, there also to be real numbers when there are no ambiguities), and symbol $\leq_f$ means that the way of comparing both members in the inequality, due to formal coherence, must be done by using a relationship for ordering the fuzzy numbers. This comparison relation $\leq_f$ may be any one from the extensive list available [19], which in turn would also allow the decision-maker to have a greater degree of freedom when it comes to establishing preferences. In more specific terms, in order to provide that theoretical model with a way for operating, let's briefly refer back to the different indices for comparing fuzzy numbers that have been described in the literature [19]. Amongst the different approaches described for comparing these amounts, for the sake of simplicity, here we shall only deal with the one that is derived from the use of indices for comparison.

Hence, by denoting as it is usual by $F(R)$ the set of fuzzy numbers, if

$$I : F(R) \rightarrow [0, 1]$$

is a comparison index for this kind of numbers, then

$$\forall X^f, Y^f \in F(R),\ X^f \leq_f Y^f \Leftrightarrow I(X^f) \leq I(Y^f)$$

whereby, according to index I that is used, different auxiliary models may be obtained for effectively solving the problem described above from the practical viewpoint. So, in general, the auxiliary model that is used to solve the problem described above from the practical viewpoint, would be approached as:

$$\max\{cx/I(A^f x) \leq I(b^f), x \geq 0\}.$$

As a trivial example of that setting, consider two fuzzy numbers $X^f, Y^f \in F(R)$, denoted as usually as $X^f = (X, X_i, X_d)$ and $Y^f = (Y, Y_i, Y_d)$, and the form of comparison is that given by Yager's First Index [19],

$$X^f \leq_f Y^f \Leftrightarrow (1/3)(X + X_i + X_d) \leq (1/3)(Y + Y_i + Y_d).$$

Then the previous model takes the following operating form,

$$\max\{cx/(A + A_i + A_d)x \leq (b + b_i + b_d), x \geq 0\}$$

which, with a sufficiently clear denotation, obviously does not involve any theoretical hindrance for solving it.

Therefore, and referring to the example for illustration that has been used so far in this Section, one possible approach might be the following:

$$\max\Big\{x_1 + 2x_2/x_1 + 5x_2 \leq 35;\ x_1 + x_2 \leq_f 11^f;\ 2x_1 + x_2 \leq 17;$$

$$3x_1 + x_2 \leq 24;\ x1, x_2 \geq 0\}\,.$$

Assuming the simplest case, i.e., that 11^f is a triangular number with a membership function (11, 11, 11.3), we may obtain a whole range of auxiliary problems that solve the previous example by providing different solutions. Specifically, by means of an example, the auxiliary model that would be obtained would be:

$$\max\{x_1 + 2x_2/x_1 + 5x_2 \leq 35;\ x_1 + x_2 \leq_f 11.1$$

$$2x_1 + x_2 \leq 17;\ 3x_1 + x_2 \leq 24;\ x_1, x_2 \geq 0\}\,.$$

From which we would obviously obtain the optimum solution to the problem in question.

With regards to the second reason that we use in this article to justify the use of the FLP models in DSS, i.e., its help in finding solutions for those in which it is not easy to find their optimum solution, the question posed is the following.

As it is well known, there are a lot of NP problems (Knapsack, Travelling Salesman, etc.) which cannot effectively be solved in all cases, but which are of the utmost importance in a number of different DSS. In these problems the decision-maker must usually accept approximate solutions instead of optimum ones. At this point the aim here is to show how the FLP can help classical MP models and techniques by providing approximate (fuzzy) solutions that may be used by the decision-maker as help to quickly obtain a good enough solution for these problems.

Let's justify this fact as follows. As it is well known, an algorithm for solving a general classical optimisation problem can be viewed as an iterative process that produces a sequence of points according to a prescribed set of instructions, together with a termination criterion. Usually we are interested in algorithms that generate a sequence $x_1, x_2, ..., x_N$ that converges to an overall, optimum solution. But in many cases however, and because of the difficulties in the problem, we may have to be satisfied with less favorable solutions. Then the iterative procedure may stop either 1) if a point belonging to a prefixed set (the solution set) is reached, or 2) if some prefixed condition for satisfaction is verified.

But, the conditions for satisfaction are not to be meant as universal ones. In fact they depend on several factors such as the decision-maker, the features of the problem, the nature of the information available, ... In any case, assuming that a solution set is prefixed, the algorithm will stop if a point in that solution set is reached. Frequently, however, the convergence to a point in the solution set is not easy because, for example, of the existence of local optimum points, and hence we must redefine some rules for terminating the iterative procedure.

Roughly speaking, the possible criteria to be taken into account for terminating the algorithms are no more than control rules. From this point of view, the control rules of the algorithms for optimisation problems can be associated to the two above points: the solution set, and the criteria for terminating the algorithm. As it is clear, fuzziness can be introduced in both points, not assuming it as inherent in the problem, but as help for obtaining, in a more effective way, some solution for satisfying the decision-maker's wishes. This is meant so that the decision-maker might be more comfortable when obtaining a solution expressed in terms of satisfaction instead of optimisation, as is the case when fuzzy control rules are applied to the control processes.

Therefore, and in the particular case of optimisation problems [4, 18], it makes sense to consider fuzziness

a) In the Solution Set, i.e., there is a membership function giving the degree with which a point belongs to that set, and

b) On the conditions for satisfaction, and hence Fuzzy Control rules on the criteria for terminating the algorithm.

In the particular and easy case of Linear Programming problems, and hence of the very well known Simplex Algorithm, if a conventional problem is assumed

$$\min\{cx/Ax = d;\ x \geq 0\}$$

the Simplex Algorithm, with the conventional denotation, can be summarised as follows,

1) Find an initial extreme point x with basis B.

2) Let x be an extreme point with basis B, and let R be the matrix corresponding to the nonbasic variables. Compute $c_B B^{-1} R - c_R$.

If this vector is non positive then stop, x is an optimal extreme point.

Else select the most positive component $c_B B^{-1} a_j - c_j$ and compute $y_j = B^{-1} a_j$:

If $y_j = B^{-1} a_j$ is less than or equal to 0 Then stop. Objective unbounded.

If $y_j = B^{-1} a_j$ is neither less than nor equal to 0 Then go to step 3.

3) Find the new extreme point by changing the current basis. Repeat step 1.

Therefore, as may be seen, in the Simplex Algorithm control rules appear mainly in the second step as

- The non positivity of the vector $c_B B^{-1} R - c_R$ could be meant in a soft sense,

- The positivity of the component $c_B B^{-1} a_j - c_j$ could be measured according to some membership function, and

- The accomplishment of $y_j = B^{-1} a_j \leq 0$, if this is viewed as a constraint, could be fuzzified.

If the first possibility is considered, a new second step can be formulated,

2') Let x be an extreme point with basis B. Compute $c_B B^{-1} R - c_R$. If

$$\forall j = 1, ..., n,\ c_B y_j - c_j <_f 0,\ c_j \in c_R$$

Then stop.

Thus this condition is stated as a fuzzy constraint, meaning that the decision-maker can accept violations in the accomplishment of the control rules, $c_B y_j - c_j < 0$, to obtain a near, and therefore approximate, optimal solution instead of a full optimal one.

3 Using fuzzy rules for terminating algorithms in MBMS.

The above fuzzy rules, meant as termination criteria in the algorithms used in practical realisations of the MBMS, will be illustrated here by means of a very well known problem, to which a great deal of work is devoted in DSS: the Travelling Salesman Problem (or Travelling Salesperson Problem, TSP) [13]. TSP finds application in a variety of situations: postal routes, tightening the nuts on some piece of machinery on assembly lines, etc. In short TSP is addressed as follows: Let G be a directed graph in which the nodes represent cities and each edge has assigned to a positive cost (the distance between each two cities). If a route of G is defined as a directed cycle that includes every vertex in G, and the cost of a route is the sum of the cost of the edges on the route, the TSP is to find a route of minimum cost.

We denote by $i = 1$ the first city of the route and by $2, 3, ...n$ the other cities, d_{ij} – the distance between the city i and the j one, the value of the variable x_{ij} is 1 if j is the next city in the route to city i and 0 otherwise. If $N = 1, 2, ..., n$, the mathematical formulation of the TSP is:

$$\min z = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, ..., n,$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, ..., n,$$

$$\sum_{i \in Q} \sum_{j \in \bar{Q}} x_{ij} \geq 1 \quad \forall Q \subset N$$

$$x_{ij} \in 0, 1 \quad i, j = 1, 2, ..., n.$$

In order to introduce fuzzy termination criteria in the exact algorithms of the TSP, we consider that the value of a TSP optimal solution is not a crisp unknown value, but a vague value, because in a great dimension TSP, for which the exact algorithms known need a lot of time to obtain an optimal solution, the decision maker can be comfortable in having an almost optimal solution instead of the very optimal one. In such a situation the optimal value can be seen as a fuzzy set on $[L_0, U_0]$ defined by a membership function as:

$$\mu(z) = \begin{cases} 1 & if \quad z < L_0 \\ f(z) & if \quad L_0 \leq z \leq U_0 \\ 0 & if \quad z > U_0 \end{cases}$$

where $f(z) \in [0, 1] \forall z \in [L_0, U_0]$, is a not increasing continuous function, L_0 a lower bound and U_0 an upper bound of the optimal value of the TSP which shall be determined “a priori” as it will be shown. As usual, this membership function shows that if the value z of a TSP route is greater than U_0 then it is not allowed by the decision maker. A lower value to L_0 can be a good solution and values between L_0 and U_0 are admissible, but the level of admission will be increasing when z decreases. Obviously, the highest level of admission is obtained when z is equal to L_0.

If the decision maker accept a not optimal solution with a membership degree not lower than $\alpha (0 < \alpha < 1)$, a termination criterion is:

$$\mu(z) \geq \alpha \text{ or } z \leq f^{-1}(\alpha) \tag{1}$$

The values of L_0, U_0 and the function f must be the correct ones in order to provide the expected results by the decision maker. Unsuitable values of L_0, U_0 can produce solutions with great errors. Equally, an incorrect function f can cancel out the flexibility.

From [18] it follows that in such situations a good option is to use a concave function in the definition of the membership function, concretely a function as

$$f(z) = \sqrt[n]{\frac{U_0 - z}{U_0 - L_0}} \tag{2}$$

for which the bounds L_0 and U_0 can be computed by suitable efficient existing algorithms.

In order to illustrate the use of fuzzy termination criteria in the TSP, the well known algorithm by Little et al. [6], that here is denoted by LMSK algorithm in short and that is specially designed for solving TSP, has been considered.

4 Solution method

4.1 LMSK algorithm

This is a branch and bound algorithm that uses relaxation of TSP as a matching problem denoted by PA (TSP). The algorithm starts by solving the PA (TSP) by the Hungarian Method; if the obtained solution does not possess sub-routes then it is an optimal solution of the TSP. Otherwise the algorithm proceeds the branch. In each iteration, one chooses a sub-problem TSP_k, the most recent among the unsolved sub-problems. If the optimal value is lower than the best current value, then it is saved as the best current value, or alternatively one branch according to this problem has sub-routes or does not have. If the optimal value is equal to or greater than the best current value, then one rejects the sub-problem and starts another iteration. The rule for branching consists in choosing a variable x_{ij} and to obtain two sub-problems by assigning 0 and 1 values to the selected variable. The process terminates when no unsolved sub-problem does exist.

In a TSP_k (node k) sub-problem, as a consequence of the above branching, there are variables x_{ij} with fixed values (0 or 1). Speaking in graph terms, there are an (i, j) edge included or not in the route. We denote by I the included edge set and by E the excluded edge set. Then, TSP_k can be described as:

$$\begin{aligned}
\min \sum_{(i,j)\in I} d_{ij} + \sum_{i\in S}\sum_{j\in T} d'_{ij}x_{ij} \\
\text{s.a.} \quad \sum_{i\in S} x_{ij} = 1, \quad \forall j \in T, \\
\sum_{j\in T} x_{ij} = 1, \quad \forall i \in S, \\
x_{ij} \in 0,1, \quad \forall (i,j),\ i \in S,\ j \in T
\end{aligned}$$

where:

$$S = \{i/(i,j) \notin I\ \forall j\}\,, \quad T = \{j/(i,j) \notin I\ \forall i\}\,,$$

$$d'_{ij} = \begin{cases} d_{ij} & (i,j) \notin E \\ \infty & (i,j) \in E \end{cases}$$

furthermore d_{ij} are the coefficients of the matrix of reduced distance of previous node, that is to say, the matrix that rests after the obtaining the optimum assignation in the previous node.

This sub-problem is solved by using the Hungarian Method. If the obtained solution has sub-routes one proceeds the branch. A rule for branching is to choose a variable x_{rs} where $r \in S$ and $s \in T$, and to make two nodes assigning value 1 or 0 each one. Little et al [6] suggest to select a variable x_{rs} with value 0 if this variable has the maximal potential of increasing in the objective function of the sub-problem. In order to make it, let

$$\{\bar{d}_{ij}\} \quad i \in S, \quad j \in T$$

be the reduced cost of the optimum solution of the sub-problem. Then, for each edge (i,j), $i \in S$, $j \in T$ with reduced cost 0, we compute:

$$p_{ij} = \min\{\bar{d}_{ik}/h \in T - j\} + \min\{\bar{d}_{hj}/h \in S - i\}$$

which is the minimum amount to increase the optimum value of the assignation to the subproblem, if the chosen variable is fixed to 0. Therefore we can choose x_{rs} such that:

$$p_{rs} = \max \{p_{ij}/i \in S,\ j \in T,\ \bar{d}_{ij} = 0\} \tag{3}$$

when the variable of branching x_{rs} is chosen, all the new nodes can be obtained making $x_{rs} = 1$ and $x_{rs} = 0$. In the first new node, I has the edge (r, s) as new element, and in the second new node, E has the edge (r, s) as new element.

Steps of LMSK algorithm

Step 1: [Starting] Let $U = \infty$ (best bound and real value) and $L = TSP$ (subproblem list).

Step 2: [Selecting a sub-problem] If $L = \Phi$ then one terminates the process, because the route associated to U is an optimal one (if $U = \infty$, the TSP has not solution).

If $L \neq \Phi$, one chooses the more recent sub-problem TSP$_i$, and one removes it from the list L. Go to step 3.

Step 3: [Upper bound determination] Solve PA(TSP$_i$) by means of the Hungarian Method. Let Z_i be the obtained value.

If $Z_i \geq U$, go to step 2.

If $Z_i < U$ and the solution is a route for TSP (there are no subroutes) then make $U = Z_i$.

If $Z_i < U$ and the solution is not a route for TSP (there are subroutes) go to step 4.

Step 4: [Branching] Choose x_{rs} according to (3) and generate two new sub-problems TSP$_{i1}$ and TSP$_{i2}$ by fixing $xrs = 0$ and $xrs = 1$. Let $L = L \cup TSP_{i1}, TSP_{i2}$.

Go to step 2.

Remark: Note that the termination criterion of this algorithm is $L \neq \Phi$.

4.2 Fuzzy termination criteria in the LMSK algorithm

To introduce a fuzzy termination criterion in the LMSK algorithm, we make a change at the starting step in order to determinate the bounds L_0 and U_0. L_0 is computed by using the method proposed in [11], and the upper bound U_0 is computed by means of the process described in [9]. In the same starting step, the decision maker will choose and fix α (the lowest level of admission). Finally, at step 2 one must include the fuzzy termination condition (1). Therefore the following new algorithm is obtained:

Step 1: [Starting] Let $U = \infty$ (best bound and real value) and $L = TSP$ (subproblem list). Solve by means of the Hungarian Method PA(TSP). If optimum matching is a route of TSP go to step 2. Else, go to 1'.

Step 1': Find L_0 and U_0, then make $U = U_0$ (best real bound) and go to step 1";

Step 1": Fix $\alpha(0 < \alpha \leq 1)$. If $0 < \alpha < 1$ let $z_0 = f^{-1}(\alpha)$ (bound for the admissible solution, where f is as in (2)). If $L \neq \Phi$ go to step 2. Else, go to step 4.

If $\alpha = 1$ (the decision maker do not want to improve an admissible solution), let $L = \Phi$ and go to step 2.

Step 2: [Selecting a sub-problem] If $L = \Phi$ or $U \leq z_0$ stop the process, as the associated route with U is admissible; if $L \neq \Phi$ go to step 1". Otherwise stop.

If $L \neq \Phi$ and $U > z_0$, select the more recent problem TSP_i, remove it from the list L and go to step 3.

Step 3: [Upper bound determination] Solve $\text{PA}(\text{TSP}_i)$ by means of the Hungarian Method. Let Z_i be the obtained value.

If $Z_i \geq U$, go to step 2.

If $Z_i < U$ and the solution is a route for TSP (there are no sub-routes) then let $U = Z_i$.

If $Z_i < U$ and the solution is not a route for TSP (there are sub-routes) go to step 4.

Step 4: [Branching] Choose x_{rs} according to (3) and generate two new sub-problems TSP_{i1} and TSP_{i2} by fixing $x_{rs} = 0$ and $x_{rs} = 1$. Take $L = L \cup TSP_{i1}, TSP_{i2}$.

Go to step 2.

The introduction of the fuzzy termination criterion on the algorithm has made it more flexible. Now, the decision maker can control the iterations because at step 1" he can introduce little values for α

and to increase them if he want to improve the admissible solution. Consequently the decision maker will take into account the time used for obtaining admissible solutions.

For the sake of illustration, let us consider finally the following TSP for 10 cities, with a distance matrix given by:

$$(d_{ij}) = \begin{bmatrix} - & 1 & 62 & 56 & 54 & 27 & 30 & 27 & 55 & 60 \\ 90 & - & 66 & 77 & 52 & 98 & 12 & 55 & 7 & 64 \\ 30 & 41 & - & 60 & 59 & 17 & 72 & 82 & 76 & 21 \\ 57 & 33 & 33 & - & 64 & 78 & 62 & 24 & 70 & 72 \\ 95 & 32 & 69 & 74 & - & 97 & 94 & 92 & 96 & 55 \\ 29 & 25 & 40 & 61 & 25 & - & 27 & 81 & 57 & 94 \\ 98 & 52 & 8 & 11 & 89 & 61 & - & 55 & 91 & 37 \\ 52 & 50 & 90 & 33 & 64 & 86 & 37 & - & 91 & 88 \\ 45 & 83 & 31 & 79 & 70 & 22 & 46 & 18 & - & 91 \\ 96 & 62 & 88 & 9 & 2 & 67 & 64 & 43 & 85 & - \end{bmatrix}$$

We consider the diagonal elements of the matrix and the distances of excluded edges in the iterations with a value $M = 10 \times (\max d_i j)$. Then, $d_{ii} = M$ and $d_{ij} = M$ if the edge (i, j) is excluded of a possible route. Then solving the problem with the exact algorithm LMSK, one obtains the optimal route

$$1 \to 2 \to 9 \to 6 \to 5 \to 10 \to 4 \to 8 \to 7 \to 3 \to 1$$

with a total distance $z = 218$, after solving 15 sub-problems (original problem included).

On the other hand when using the LMSK algorithm with a fuzzy termination criterion, a function f as (2), $n = 2$, and bounds $L_0 = 208$ and $U_0 = 308$ (at the starting step 1'), the admissible solutions for the different values of α is shown in the following table:

One can observe that for $\alpha = 0.8$, an admissible solution is obtained by solving only the 66% of all the sub-problems that the classical algorithm solves. However, the admissible value obtained is very close to the optimum. It is then evident that the saving in time is upper in comparison with the difference between the admissible value and the

α	$z_0 = f^{-1}(\alpha)$	Admissible route	Admissible value	Sub-problems solved (It.)
0.5	283	1-7-3-10-5-2-9-8-4-6-1	258	8
0.8	244	1-8-7-4-3-19-5-2-9-6-1	221	10
0.94	219.64	1-2-9-6-5-10-4-8-7-3-1	218	14

Table 1. Admissible solutions for the example with the LMSK algorithm with fuzzy termination criteria

optimum value. Furthermore for $\alpha = 0.94$ one obtains an admissible solution which is the exact one, and by performing less iterations than the original classical algorithm. The more the number of cities are in the TSP, the more these advantages are evident.

Acknowledgement

Research supported partly under the project PB98-1305 of the Spanish M.E.C.D.

References

[1] J.M. Cadenas, J.L. Verdegay (1999): *Optimisation Models with Imprecise Data.* SPUM (In Spanish)

[2] M. Delgado, J. Kacprzyk, J.L. Verdegay, M.A. Vila, Eds (1994): *Fuzzy Optimisation: Recent Advances,* Physica-Verlag.

[3] P. Dunn-Rankin (1983): *Scaling Methods. Lawrence Erlabaum Associates.* Hillsdale.

[4] F. Herrera, J.L. Verdegay (1996): *Fuzzy Control Rules in Optimisation Problems.* Scientia Iranica 3, 89–96.

[5] F. Herrera, J.L. Verdegay (1997): *Fuzzy Sets and Operations Research: Perspectives.* Fuzzy Sets and Systems 90, 207–218.

[6] J.D.C. Little, K.G. Murty, D.W. Sweeney y C. Karel (1963): *An algorithm for the traveling salesman problem,* Operations research 11, 972–989.

[7] J.L. Melia (1990): *One dimension Scaling Methods.* Valencia (in Spanish).

[8] G.A. Miller (1967): *The Magical Number Seven Plus or Minus Two: Some Limits on our Capacity for Processing Information. The Psychology of Communication.* Penguin Books Inc.

[9] D.J. Rosenkrantz., R.E. Stearns y P.M. Lewis II (1977): *An analysis of several heuristics for the traveling salesman problem.* SIAM J. Computing 6, 563–581.

[10] A.P. Sage (1991): *Decision Support Systems Engineering.* Wiley Series in Systems Engineering. John Wiley and Sons

[11] H.M. Salkin and K. Mathur (1989): *Foundations of integer programming,* North-Holland, New York.

[12] A. Sancho and J.L. Verdegay (1999): *Methods for the Construction of Membership Functions.* International Journal of Intelligent Systems 14, 1213–1230.

[13] A. Sancho-Royo, J.L. Verdegay y E. Vergara-Moreno: *Some Practical Problems in Fuzzy Sets-based Decision Support Systems.* MathWare and Soft Computing VI, 2-3 (1999), 173–187.

[14] E.Turban (1988): *Decision Support and Expert Systems (Managerial Perspectives).* Macmillan Series in Information Systems. Macmillan Publishing Company.

[15] B. Türksen (1991): *Measurement of Membership Functions and their Acquisition.* Fuzzy Sets and Systems 40, 5–38.

[16] B. Türksen and A. Wilson (1994): *A Fuzzy Set Preference Model for Consumer Choice.* Fuzzy Sets and Systems 68, 253–266.

[17] B. Türksen and A. Wilson (1995): *A Fuzzy Set Model for Market Share and Preference Prediction.* European Journal of Operative Research 82, 39–52.

[18] E.R. Vergara (1999): *New Termination Criteria for Optimisation Algorithms.* Ph. D. Dissertation. Universidad de Granada (in Spanish).

[19] X. Wang, E. Kerre (1996): *On the Classification and the Dependencies of the Ordering Methods.* In D. Ruan (Ed.): Fuzzy Logic Foundation and Industrial Applications. International Series in Intelligent Technologies. Kluwer 73–90

J.L. Verdegay, E. Vergara-Moreno, Received November 16, 2001

José L. VERDEGAY[1],
Depto. de Ciencias de la Computación e
Inteligencia Artificial.
Universidad de Granada.
18071 Granada, SPAIN

Edmundo VERGARA-MORENO,
Depto. de Matemáticas.
Universidad Nacional de Trujillo.
Trujillo, PERU