G.U. Journal of Science
22(4): 267-276 (2009)

www.gujs.org

From Analytical Perspective to Heuristic Approach: Travelling Salesman Problem with Discrete Fuzzy Travel Times

Selçuk Kürşat İŞLEYEN [1], Saadettin Erhan KESEN [2♠], Ömer Faruk BAYKOÇ [3]

[1] *Gazi University, Faculty of Engineering and Architecture, Department of Industrial Engineering, 06570, Ankara, TURKEY*
[2] *Selçuk University, Faculty of Engineering and Architecture, Department of Industrial Engineering, 42075, Konya, TURKEY*
[3] *Gazi University, Faculty of Engineering and Architecture, Department of Industrial Engineering, 06570, Ankara, TURKEY*

Received: 23.07.2007 Revised: 10.04.2009 Accepted: 06.07.2009

ABSTRACT

In today's business, travelling times are affected by many factors such as traffic, weather, road etc. so deterministic approaches can not find any solution for problems where such an ambiguity happens. This paper deals with the Travelling Salesman Problem (TSP) in which travelling times are inaccurate. We use discrete fuzzy numbers to represent the uncertainty. Discrete fuzzy numbers are then converted to the triangular fuzzy numbers (TFNs). TFNs enforce the TSP model to have a non-linear objective function. Then we make an approximation and obtain linear model (LM) by inserting lower, medium, and lower values of the TFNs into one since non-linear model (NLM) can trap local optima. Finally, we develop Iterated Local Search (ILS) technique to get good solutions in a shorter time in the case that objective function is non-linear. NLM, LM and ILS are compared on a wide range of test problems that randomly generated. Results show that ILS technique is very promising and finds much better solutions in a very shorter computational time. Hence, it can be substituted in the place of NLM.

Key Words: *Travelling Salesman Problem (TSP), Discrete Fuzzy Numbers, Heuristic*.

1. INTRODUCTION

TSP is one of the oldest and most widely studied optimization problems in the fields of operations research and combinatorial optimization. In a TSP, a salesman who starts from his point of origin must visit each of *n* cities exactly once and return to the point of origin to complete the tour. If we denote the starting point as *1* and the *n* cities as integers *2,….,n.* and a complete tour can be written as $\tau =(1, i_1,\ldots, i_n,1)$.

The basic problem is to find the tour which has the minimum time ($min\ T(\tau)$).

Up to now, huge variants of TSP have been widely studied in the literature such as TSP with time windows, the multi-TSP, stochastic TSP or vehicle routing problems which are the extensions of TSP [1-5].

However, in many real world applications, we can consider the travel times among cities as fuzzy numbers. This problem arises the issue of fuzzy TSP (FTSP). Although the TSP has received a great deal of attention, number of researches on the FTSP is limited. With regards to fuzzy environment, Voxman [6] first introduced the concept of discrete fuzzy numbers which is useful for some applications. For instance, Wang et al. [7] successfully used the conception to find the pixel value of the center point of a window. Then Kung and Chuang [8] solved the shortest path problem with DFNs arc length.

♠Corresponding author, e-mail: sekesen@gazi.edu.tr

Aforementioned challenging applications of DFNs on different kind of problems inspired us to apply the concept on TSP. To our knowledge, TSP problem with discrete fuzzy travel times is not studied yet. As Voxman [6] indicated, we translated discrete fuzzy numbers into TFNs. Travel times with TFNs make the objective function of TSP non-linear. But since non-linear models can not guarantee the optimal solution, we relax the NLP by converting the model to LP. It is proved that LP finds solutions with a little deterioration.

Many researchers has recently paid closer attention to metaheuristic approaches such as Iterated Local Search (ILS), Genetic Algorithm (GA), Tabu Serach (TS), Particle Swarm Optimization (PSO) etc. for combinatorial optimization problems since they can find good solutions in a quite shorter computational time. For this purpose, we develop an ILS algorithm to the TSP, which belongs to the combinatorial optimization problems.

Remainder of the paper is organized as follows: In subsequent section, we present the notations and basic definitions for DFNs and conversion of DFNs into TFNs. Section 3 demonstrates the NLP and LP model for TSP. Section 4 is devoted to ILS technique. Section 5 shows the comparative results. Paper finalizes with concluding remarks and future directions in Section 6.

2. NOTATIONS AND BASIC DEFINITIONS

A fuzzy set on a set *X* is a function $\mu : X \longrightarrow I$, where ***I*** is the unit interval. If μ, *v* are fuzzy sets on *X*, then the fuzzy sets $\mu \vee v$ and $\mu \wedge v$ are defined by

$$\mu \vee v = \max\{\mu(x), v(x)\} \text{ and}$$
$$\mu \wedge v = mix\{\mu(x), v(x)\}.$$

If μ is a fuzzy number on *X*, then the complement, μ_C, of μ is defined by

$$\mu_C(x) = 1 - \mu(x) \text{ for each } x \in X$$

The support of μ, *supp* μ is defined by

$$supp\ \mu = \{x \in X \mid \mu(x) \neq 0\}$$

In this paper we will generally assume $X = R^1$ and *supp X* is finite. If μ has a finite support $x_1 < x_2 < \ldots < x_n$ then $Count\ \mu$ is defined by

$$Count\ \mu = \sum_{i=1}^{n} \mu(x_i) \qquad (1)$$

Definition: A fuzzy set $\mu : R^1 \longrightarrow I$ is a DFN if μ has a finite support $x_1 < x_2 < \ldots < x_n$ and there are indices *s, t,* $1 \le s \le t \le n$ such that

- $\mu(x_i) = 1$ whenever $s \le i \le t$,
- If $i \le j \le s$, then $\mu(x_i) \le \mu(x_j) < 1$,
- If $t \le p \le q$, then $1 > \mu(x_p) \ge \mu(x_q)$,

Next we define the concepts of value, ambiguity, and fuzziness for DFNs.

Definition: Let μ be a DFN with support $x_1 < x_2 < \ldots < x_n$. Then the value ***v***, of μ is defined by

$$v = \left[\sum_{i=1}^{n} x_i \mu(x_i)\right] \Big/ \sum_{i=1}^{n} \mu(x_i) \qquad (2)$$

Definition: Let μ be a DFN with support $x_1 < x_2 < \ldots < x_n$ and the value ***v*****.** the ambiguity ***a***, of μ is defined by

$$a = \sum_{i=1}^{n} | x_i - v | \mu(x_i) \qquad (3)$$

Definition: Let μ be a DFN with support $x_1 < x_2 < \ldots < x_n$. Then the fuzziness, ***f***, of μ is defined by

$$f = \frac{Count\ \mu \wedge \mu_c}{Count\ \mu \vee \mu_c} \qquad (4)$$

Since fuzzy numbers need not be symmetric, it is sensible to regard left-hand and right-hand ambiguity and fuzziness.

Definition: Let μ be a DFN with support $x_1 < x_2 < \ldots < x_n$. Let ***v*** be the value of μ. Then the right-hand ambiguity, a_R, of μ is defined by

$a_R = \sum_{i=1}^{n} | x_i - v | \mu(x_i)$ where the indices *i* run over all *i* such $x_i \ge v$

The right-hand fuzziness, f_R, of μ is defined by

$f_R = \dfrac{Count\ \mu \wedge \mu_c}{Count\ \mu \vee \mu_c}$ where the indices *i* run over all *i* such $x_i \ge v$

The left-hand ambiguity a_L, and left-hand fuzziness, f_L, of μ are defined in the same manner.

TFNs are characterized by three parameters a, b and c as is seen in Figure 1.

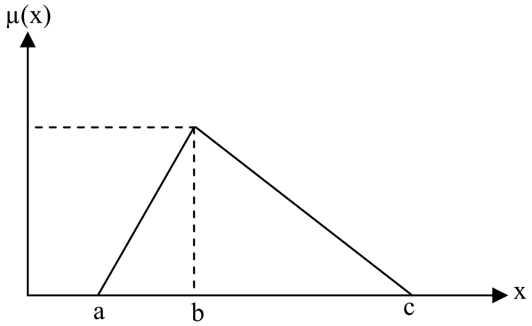

Figure 1. Generic illustration of TFNs with three parameters.

By using the right-hand ambiguity and right-hand fuzziness obtained by DFNs, we calculate the parameter *c*. Parameter *a* is calculated in the same fashion. Parameter *b* is equal to value of DFN.

If $f_R \geq \frac{1}{3}$ then

$$d_R = \frac{-12a_R k(f_R^2 k + 2f_R k + k + 2f_R + f_R^2 + 1)}{2k^3 f_R^2 - 28k^3 f_R + 2k^3 + 3k^2 f_R^2 - 42k^2 f_R + 3k^2 + 3f_R^2 k - 18f_R k + 3k + f_R^2 - 2f_R + 1} \tag{5}$$

If $f_R < \frac{1}{3}$ then

$$d_R = \frac{-3a_R(f_R + 1)k(f_R k + f_R + k + 1)}{(f_R k^2 - 3k^2 + f_R k - 3k + f_R)f_R(2k + 1)} \tag{6}$$

Voxman [6] indicates that parameter *k* should be selected bigger than and equal 2 and we chose *k* as 3 for all test problems.

$(a,b,c) = (v - d_L, v, v + d_R)$

When we calculate d_L, we use the above formulation but a_L, and f_L are used instead of a_R and f_R respectively. In the next section, conversion of DFNs into TFNs is shown numerically.

3. NON-LINEAR AND LINEAR FORM OF TSP WITH TFNs

In the case where travel times between cities are characterized by TFNs, a length of any tour will also be TFNs as shown below.

Let's suppose that we have three cities *i*, *j*, and *k*.

Also suppose our tour is $\tau = i \to j \to k \to i$

Travel times between cities are as follows:

$$d_{ij} = (a_{ij}, b_{ij}, c_{ij})$$
$$d_{jk} = (a_{jk}, b_{jk}, c_{jk})$$
$$d_{ki} = (a_{ki}, b_{ki}, c_{ki})$$

So, tour time is shown below.

$$T(\tau) = (a_{ij} + a_{jk} + a_{ki}, b_{ij} + b_{jk} + b_{ki}, c_{ij} + c_{jk} + c_{ki})$$

To get the crisp value of tour time $(T(\tau_{crisp}))$, we should know how far tour time is to the origin (0, 0, 0) and calculation is denoted in Eq. 7:

$$L\left(\tau_{crisp}\right)=\sqrt{\frac{\left(a_{ij}+a_{jk}+a_{ki}-0\right)^2+\left(b_{ij}+b_{jk}+b_{ki}-0\right)^2+\left(c_{ij}+c_{jk}+c_{ki}-0\right)^2}{3}} \tag{7}$$

After tour times are obtained as crisp values, we can search for the tour with minimum length among all feasible tours. Non-linear form of TSP problem with TFN travel times can be modeled as follows:

The following symbols are used in the model;

i,j = city (i,j =1,2,….,n) ,

$D^L{}_{i,j}$ = the travel time between city i and city j. (left-hand of the triangular fuzzy number)

$D^M{}_{i,j}$ = the travel time between city i and city j. (middle of the triangular fuzzy number)

$D^R{}_{i,j}$ = the travel time between city i and city j. (right-hand of the triangular fuzzy number)

$$\begin{cases} X_{i,j}=1, & \text{if vehicle travels directly from city i to city j} \\ X_{i,j}=0, & \text{otherwise} \end{cases}$$

N = size of the problem or number of city.

$$MIN \quad \sqrt{\frac{\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\left(D^L{}_{i,j}*X_{i,j}\right)^2+\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\left(D^M{}_{i,j}*X_{i,j}\right)^2+\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\left(D^R{}_{i,j}*X_{i,j}\right)^2}{3}} \tag{8}$$

subject to:

$$\sum_{i=1}^{n} X_{i,j}=1 \qquad \forall_j, j\neq i \tag{9}$$

$$\sum_{j=1}^{n} X_{i,j}=1 \qquad \forall_i, i\neq j \tag{10}$$

$$U(i)-U(j)+n*X_{i,j}\leq n-1 \qquad \begin{array}{l}\forall_i,(i=1,2,...,n)\\ \forall_j,(j=1,2,...,n)\end{array} \tag{11}$$

All $X_{i,j}$ = 0 or 1 and all $U(i)\geq 0$ and is a set of integers.

Constraint sets (9) and (10) are degree constraints and ensure that city i connects to one city only and that city is reached from exactly one city. Constraint set (11) is required for sub-tour elimination.

Since the objective function of the model is non-linear, it can not guarantee to find the optimal solution but can trap local optimum. Besides, solution time increases when the problem size gets bigger. To overcome this difficulty and find relatively better solution to the problem, the objective function of the problem can be made linear by using the approach shown in Eq. 12: The problem is subject to the same constraints.

$$MIN \quad \sum_{i=1}^{n}\sum_{\substack{j=1\\ i\neq j}}^{n}\sqrt{\frac{\left(D^L{}_{i,j}\right)^2+\left(D^M{}_{i,j}\right)^2+\left(D^R{}_{i,j}\right)^2}{3}}*X_{i,j} \tag{12}$$

The reason why we develop such an approach is to use it as a control mechanism. We prove that the optimal solution of the non-linear model is always less than that of linear model (See Appendix). When linear model finds better solution than does non linear-model, it is clear that the solution non-linear model gets is not optimal. Even if the solution non-linear model finds is less than that of linear model, we can not ensure that the solution non-linear model finds is optimal but that solution is amongst pretty better ones. In the next section, we further investigate the problem and offer an ILS algorithm whether we can obtain better solution than that of non-linear and linear model. By comparing these three approaches, we have a chance to evaluate their performances concurrently. To illustrate the solution phase of the NLM and LM, consider the following example with five cities. Data set of the sample problem is shown in Table 1.

Table 1. Travel times with DFNs for the illustrative example.

Cities	Travel times (Discrete Fuzzy Numbers)
1-2	0.1/26 0.1/54 0.5/60 0.9/71 1/75 0.9/76 0.6/85 0.3/87 0.1/99
1-3	0.3/43 0.4/45 0.5/54 0.7/77 0.7/80 0.4/94 0.4/98
1-4	0.1/33 0.3/39 0.6/47 0.9/67 0.9/73 0.9/74 0.3/77 0.3/94
1-5	0.1/18 0.1/19 0.6/24 0.7/31 0.6/47 0.3/55 0.1/68
2-3	0.1/16 0.3/19 0.9/21 0.6/40 0.2/83
2-4	0.2/13 0.4/32 0.4/57 1/58 1/68 1/82 0.7/85 0.4/94 0.4/98
2-5	0.3/14 0.4/23 0.9/66 1/71 1/83 0.6/90 0.3/100
3-4	0.2/16 0.3/46 0.4/49 0.6/68 0.3/75 0.3/87
3-5	0.1/16 0.3/24 0.4/62 0.7/64 0.8/72 1/77 0.8/80 0.4/87 0.3/93 0.2/98
4-5	0.1/14 0.2/24 0.2/51 0.6/55 0.8/58 0.9/72 0.8/75 0.5/84 0.2/89 0.1/93

We show the transformation of DFN to TFN for travel times between city 1 and city 2. Other DFNs are transformed by the similar way. Calculations for the other travel times are given in Tables 2, 3, and 4.

0.1/26 0.1/54 0.5/60 0.9/71 1/75 0.9/76 0.6/85 0.3/87 0.1/99

The value v, of DFN is

$$v = \left[\sum_{i=1}^{n} x_i \mu(x_i) \right] \Big/ \sum_{i=1}^{n} \mu(x_i)$$

$v = \frac{332.3}{4.5} = 73.844$ (It is the middle of the TFN)

Then the right-hand ambiguity, a_R and then the left-hand ambiguity, a_L

$$a_R = \sum_{i=1}^{n} | x_i - v | \mu(x_i) \quad x_i \geq v \qquad a_L = \sum_{i=1}^{n} | x_i - v | \mu(x_i) \quad x_i < v$$

$$a_R = 16.25 \qquad a_L = 16.25$$

The right-hand fuzziness, f_R and the left-hand fuzziness, f_L

$$f_R = \frac{Count\ \mu \wedge \mu_c}{Count\ \mu \vee \mu_c} \quad x_i \geq v \qquad f_L = \frac{Count\ \mu \wedge \mu_c}{Count\ \mu \vee \mu_c} \quad x_i < v$$

$$f_R = 0.22 \qquad f_L = 0.25$$

Both f_R and f_L are smaller than 1/3 then

$d_R = 17.084$ and $d_L = 15.949$

TFN = $(v - d_L, v, v + d_R)$

So DFN = {0.1/26 0.1/54 0.5/60 0.9/71 1/75 0.9/76 0.6/85 0.3/87 0.1/99} is equal to

TFN = $(57.895, 73.844, 90.928)$

If left hand of the TFN is less than 0, we assume it is equal to 0. In Tables 2, 3, and 4 all conversions are given.

Table 2. Left-hand of the TFN (D^L).

	1	2	3	4	5
1	0	57.895	54.760	47.627	22.810
2	57.895	0	15.011	25.934	38.442
3	54.760	15.011	0	47.289	48.675
4	47.627	25.934	47.289	0	38.685
5	22.810	38.442	48.675	38.685	0

Table 3. Middle of the TFN (D^M).

	1	2	3	4	5
1	0	73.844	71.941	66.767	36.520
2	73.844	0	31.810	69.545	69.067
3	71.941	31.810	0	60.000	71.860
4	66.767	69.545	60.000	0	65.841
5	36.520	69.067	71.860	65.841	0

Table 4: Right-hand of the TFN (D^R).

	1	2	3	4	5
1	0	90.928	90.455	91.542	48.686
2	90.928	0	42.879	104.600	109.471
3	90.455	42.879	0	71.714	99.343
4	91.542	104.600	71.714	0	92.997
5	48.686	109.471	99.343	92.997	0

All the experiments for NLM and LM are conducted by using LINGO 8.0 optimization software package. For the example, optimal tour is 1→2→3→4→5→1 and optimal tour time *($T(\tau)$)* is 274.101. When the problem is solved with linear objective function, optimal tour time *($T(\tau)$)* is 274.953 with the same city order. Note that LM solution is a little worse than that of NLM.

4. ITERATED LOCAL SEARCH (ILS)

ILS which is developed among meta-heuristics for difficult problems is not only an algorithm producing effective solutions but it is also a random search method which can easily be implemented in practice. It can be referred to as the first ILS study conducted by Martin et al. [9] for TSP. Lourenço et al. [10] gained some information about the structure of ILS algorithm in their studies. The success of the ILS is not limited to TSP. Many previous studies suggest that this method was also successful in scheduling problems. Examples are Single-Machine Total Weighted Tardiness Scheduling Problem [11], flow-shop scheduling problems [12,13], and job-shop scheduling problems [14], quadratic assignment problem [15]. For a detailed review of other applications, readers are referred to [16].

4.1. The General Structure of ILS Algorithm

The ILS algorithm, as previously mentioned, is a random search method developed for NP-hard problems. The most important characteristic of the ILS algorithm is its ability to jump to other points of the solution space (S) by masking the good characteristics of a solution which is stuck to the local optimum. This jumping action is achieved by a process called perturbation. There are four components that should be taken into consideration while applying an ILS algorithm. These are initial solution, local search, perturbation and acceptance criterion.

4.2. Local Search

Performance of ILS is remarkably sensitive to choice of embedded heuristic. In practice, there may be many different algorithms that can be used for the embedded heuristic. Two different local search heuristics were used in this study in order to increase the effectiveness of the solution of the ILS algorithm. These are:

Two-Node-Exchange

Given a tour, any two nodes in the current tour are changed. If this results in a better feasible tour the change is accepted. Procedure is repeated until no improvement is achieved.

2-p-Opt

Given a tour, its 2-p-opt neighborhood is the set of tours obtained by reversing a section of s (a set of consecutive nodes) and adjusting the arcs adjacent to the reversed section [17].

4.3. Perturbation

The objective here is to escape from being trapped local optimum by applying perturbations to the current local minimum. In Figure 2 [10], perturbation is applied the current tour (s*) and this leads to an intermediate state s' and Local Search is applied to s' and after local search a new solution s*' is reached. If s*'passes an acceptance test, we accept the s*' as a current tour, otherwise perturbation repeated on the s*. Until the termination condition is met, algorithm steps are repeated. Figure 3 shows the general working principles of ILS.

The effect of the perturbation depends on how strong the perturbation is. If the perturbation is too small, it is possible to reach the same local optimum. If the perturbation is too large, then the ILS algorithm will behave like random restart type algorithm.

The present study selected a perturbation mechanism which is effective for TSP and called double bridge move [10]. Double bridge move cuts the current tour at three random positions and uses a particular way of reconnecting the four remaining tour segments. Figure 4 shows the double bridge move as a perturbation mechanism.

In this study, the termination condition was established as a maximum number of iterations and the algorithm was limited to different number of iterations for test problems depending on problem size. The iteration number was taken to be the solution number applied perturbation. In the following section, test problems and their results are discussed.

5. COMPUTATIONAL RESULTS

Fifteen test problems are randomly generated, ranging 5 to 100 cities. Membership values are chosen corresponding to uniform distribution with a lower bound of 0.1 and upper bound of 1. Travel times also is also assumed to be uniformly distributed with a lower bound 10 and upper bound of 100. DFNs have the number of parameters between 5 and 10, corresponding to uniform distribution. ILS algorithm is coded with Microsoft Visual C# language and run on a PC with 1.80 GHz CPU and 2.0 GB RAM. At Table 5, NLM, LM, and ILS results were compared in terms of solution quality and CPU times to evaluate the performance of the ILS algorithm. Due to the stochastic nature of the ILS algorithm, each problem was run five times and the best solution (S^*_{ILS}) and average of the five run (AV(SILS)) is given. Since NLM can also trap local optimum and give the different solutions for any problem, it is run five times and the best solution (S^*_{NLM}) and average of the five run (AV(SNLM)) is given. The solutions that LM finds (SLM) are also given in Table 5.

Dashed lines in Table 5 mean that any solution is not sought because NLM is most unlikely to find any solution in a reasonable time. For the first seven problems, NLM finds local optimum. As is seen in Table 5, linear model gives us an insight about how efficient solutions that ILS finds are. As mentioned earlier in the text, LM deviates little from the NLM solution. ILS catches better solutions than those of LM. With regards to computational time, ILS works nearly perfect even if problem size much bigger.

6. CONCLUDING REMARKS AND FUTURE DIRECTIONS

In this paper, we examine the Travelling Salesman Problem with discrete fuzzy travel times. Although it is one the most prominent and studied problem in combinatorial optimization problems, no study is available in literature about the problem with discrete fuzzy travel times. We convert the discrete fuzzy numbers into triangular fuzzy numbers by using the approach introduced in Voxman [6]. Non-linear objective function is constructed for the problem. To have an insight, we also make an approximation by linearizing the objective function and we prove that LM model always has worse results than those of NLM in the case the optimality is guaranteed. We further investigate our research and develop an ILS algorithm to get good results in a very shorter computational time since NLM's solution time deteriorates when problem size gets bigger. We compare NLM, LM, and ILS solutions on a broad range of test instances and we see that ILS catches quite good results in very shorter computational time and can be substituted in the place of NLM.

This study can be extended in some points. Other metaheuristic approaches like Genetic Algorithm, Simulated Annealing, Tabu Search, Ant Colony Optimization etc. can be applied to the problem and comparative results may be given. On the other hand, other combinatorial optimization problems such as Vehicle Routing Problem, Scheduling etc. with fuzzy environment can be studied in detail.

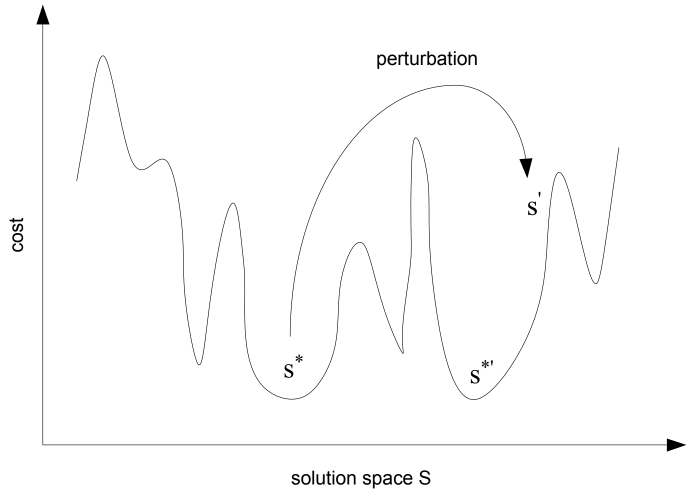

Figure 2. Perturbation phase.

Procedure Iterated Local Search
  s_0 = Initial Solution
  s^* = Local Search (s_0)
  repeat
    s' = Perturbation(s^*)
    $s^{*'}$= Local Search (s')
    s^* = Acceptance Criterion (s^*, $s^{*'}$)
  until termination condition met
end procedure

Figure 3. General working principles of ILS.

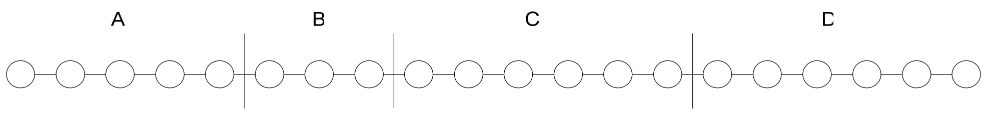

Figure 4a. Current tour before perturbation

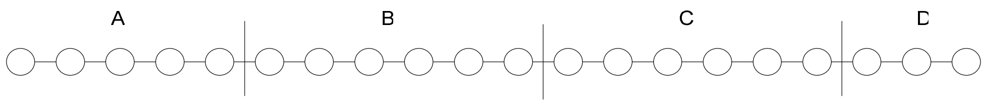

Figure 4b. New tour after perturbation

Figure 4. Double bridge move.

Table 5: Comparative results of NLM, LM, and ILS on test problems.

Prob. Parameters		Non-Linear Model			Linear Model		Iterated Local Search			
Prob #	Prob. Size	S^*_{NLM}	$AV(S_{NLM})$	CPU (Secs)	S_{LM}	CPU (Secs)	*ITER*	S^*_{ILS}	$AV(S_{ILS})$	CPU (Secs)
1	5	274.10	274.10	<1	274.95	<1	50	274.10	247.10	<1
2	7	356.72	356.72	<1	357.76	<1	50	356.72	356.72	<1
3	10	484.19	501.81	3	488.37	2	70	484.19	484.19	<1
4	12	540.04	540.04	2	543.56	<1	70	540.04	540.04	<1
5	15	717.33	735.62	29	696.85	18	100	688.35	688.35	<1
6	18	791.02	800.59	64	802.01	4	150	791.02	791.02	<1
7	20	858.72	881.76	130	870.58	7	150	858.72	858.72	<1
8	30	--	--	--	1243.27	8	200	1229.13	1230.83	<1
9	40	--	--	--	1590.12	10173	250	1576.39	1578.22	<1
10	50	--	--	--	1942.22	220	300	1921.70	1929.10	2
11	60	--	--	--	--	--	600	2296.26	2299.55	2
12	70	--	--	--	--	--	700	2604.46	2611.48	6
13	80	--	--	--	--	--	700	2895.41	2899.92	8
14	90	--	--	--	--	--	800	3219.27	3222.16	12
15	100	--	--	--	--	--	1000	3514.55	3522.34	18

Appendix: The linear model solution is always worse (bigger) than that of the non-linear model solution if we suppose that non-linear model finds the global optimum.

Consider any tour i→j→k.

Suppose that travel times between city *i* and *j* is (a_1, b_1, c_1), and travel times between city *j* and *k* is (a_2, b_2, c_2).

$$\underbrace{\sqrt{a_1^2+b_1^2+c_1^2}+\sqrt{a_2^2+b_2^2+c_2^2}}_{LM} \geq \underbrace{\sqrt{(a_1+a_2)^2+(b_1+b_2)^2+(c_1+c_2)^2}}_{NLM}$$

$$\left(\sqrt{a_1^2+b_1^2+c_1^2}+\sqrt{a_2^2+b_2^2+c_2^2}\right)^2 \geq \left(\sqrt{(a_1+a_2)^2+(b_1+b_2)^2+(c_1+c_2)^2}\right)^2$$

$$a_1^2+b_1^2+c_1^2+a_2^2+b_2^2+c_2^2+2\sqrt{\left(a_1^2+b_1^2+c_1^2\right)\left(a_2^2+b_2^2+c_2^2\right)} \geq a_1^2+b_1^2+c_1^2+a_2^2+b_2^2+c_2^2+2(a_1a_2+b_1b_2+c_1c_2)$$

$$\sqrt{\left(a_1^2+b_1^2+c_1^2\right)\left(a_2^2+b_2^2+c_2^2\right)} \geq (a_1a_2+b_1b_2+c_1c_2)$$

$$\left(\sqrt{\left(a_1^2+b_1^2+c_1^2\right)\left(a_2^2+b_2^2+c_2^2\right)}\right)^2 \geq (a_1a_2+b_1b_2+c_1c_2)^2$$

$$\left(a_1^2+b_1^2+c_1^2\right)\left(a_2^2+b_2^2+c_2^2\right) \geq (a_1a_2+b_1b_2+c_1c_2)^2$$

$$a_1^2a_2^2+b_1^2b_2^2+c_1^2c_2^2+a_1^2b_2^2+a_1^2c_2^2+a_2^2b_1^2+b_1^2c_2^2+a_2^2c_1^2+b_2^2c_1^2 \geq a_1^2a_2^2+b_1^2b_2^2+c_1^2c_2^2+2\left(a_1a_2b_1b_2+a_1a_2c_1c_2+b_1b_2c_1c_2\right)$$

$$\underbrace{a_1^2b_2^2-2a_1a_2b_1b_2+a_2^2b_1^2}+\underbrace{a_1^2c_2^2-2a_1a_2c_1c_2+a_2^2c_1^2}+\underbrace{b_1^2c_2^2-2b_1b_2c_1c_2+b_2^2c_1^2}\geq 0$$

$$\underbrace{\left(a_1b_2-a_2b_1\right)^2}_{\geq 0}+\underbrace{\left(a_1c_2-a_2c_1\right)^2}_{\geq 0}+\underbrace{\left(b_1c_2-b_2c_1\right)^2}_{\geq 0}\geq 0$$

Since all the terms are grater than 0 and we prove that objective function value of linear model is always worse (bigger) than that of non-linear model.

Acknowledgments

Authors would like to thank anonymous referees for their constructive and valuable comments, which lead to a big improvement on the paper.

REFERENCES

[1] Laporte, G., Osman I.H., "Routing problems: A bibliography", ***Annals of Operations Research***, 61: 227-262 (1995).

[2] Gendreau, M., Laporte, G., Seguin, R., "Stochastic vehicle routing", ***European Journal of Operational Research***, 88: 3-12 (1996).

[3] Bektas, T., "The multiple traveling salesman problem: an overview of formulations and solution procedures", ***Omega***, 34: 209-219 (2006).

[4] Jaillet, P., "A Priori solution of a traveling salesman problem in which a random subset of the customers are visited", Operations Research, 36: 929-936 (1988).

[5] Dumas, Y., Desrosiers, J., Gelinas, E.; Solomon, M., "Optimal algorithm for the traveling salesman problem with time windows", ***Operations Research***, 43: 367-371 (1995).

[6] Voxman, W., "Canonical representations of discrete fuzzy numbers", ***Fuzzy Sets and Systems***, 118: 457-466 (2001).

[7] Wang, G., Wu, C., Chunhui, Z., Representation and operations of discrete fuzzy numbers, ***Southeast Asian Bulletin of Mathematics***, 28: 1003-1010 (2005).

[8] Kung, J.Y., Chuang, T.N., "The shortest path problem with discrete fuzzy arc lengths", ***Computers and Mathematics with Applications***, 49: 263-270 (2005).

[9] Martin, O., Otto, S.W., Felten, E.W., "Large-step Markov chains for the TSP incorporating local search heuristics", ***Operations Research Letter***, 11: 219-224 (1992).

[10] Lourenço, H.R., Martin, O.C., Stützle, T., "A beginner's introduction to iterated local search", ***In: Proceedings of 4th Metaheuristics International Conference***, (2001).

[11] Congram, R.K., Potts, C.N., Van de Velde, S.L., "An Iterated Dynasearch Algorithm for the Single-Machine Total Weighted Tardiness Scheduling Problem", ***Informs Journal on Compuing***, 14(1): 52-67 (2002).

[12] Stützle, T., "Applying iterated local search to the permutation flow shop problem", ***Technical Report AIDA-98-04***, FG Intellektik, TU Darmstadt, 23 (1998).

[13] Yang, Y., Kreipl, S., Pinedo, M., "Heuristics for minimizing total weighted tardiness in flexible flow shops", ***Journal of Scheduling***, 3(2):89-108 (2000).

[14] Balas, E., Vazaconoulos, A., "Guided Local Search with Shifting Bottleneck for Job Shop Scheduling", ***Management Science***, 44(2): 262-275 (1998).

[15] Stützle, T., "Iterated local search for the quadratic assignment problem", ***European Journal of Operational Research***, 174(3): 1519-1539 (2006).

[16] Lourenço, H.R., Martin, O.C., Stützle, T., "Handbooks of Metaheuristics", Editor: Fred Glover, ***Kluwer Academic Publishers***, 321-353 (2002).

[17] Bianchi, L., Knowles, J., Bowler, N., "Local search for the probabilistic traveling salesman problem: Correction to the 2-p-opt and 1-shift algorithms", ***European Journal of Operational Research,*** 162(1): 206-219 (2005).