Double-ended nearest and loneliest neighbour – a nearest neighbour heuristic variation for the travelling salesman problem

Fernando Guilherme Silvano Lobo Pimentel
Bank of Portugal
fpimentel@bportugal.pt

Abstract

This paper presents a new tour construction heuristic for the travelling salesman problem that introduces the concept of loneliness of a city computed from the average distance of that city to all others and combines it with ideas from other nearest neighbour heuristics. Having the same time complexity of the faster nearest neighbour heuristics, the new method clearly leads to better tours, outperforming them as well as several other tour construction heuristics reported in the literature. A promising feature of the proposed heuristic is that it gives some priority to more isolated locations in travel route definitions. The earlier distribution of goods and services to loneliest sites might be considered a positive social externality that is appealing to the application of heuristics by public or private institutions that are engaged in acts of social responsibility.

Keywords: Travelling Salesman, Heuristics, Nearest Neighbour

Resumo

Este artigo apresenta uma nova heurística para o problema do caixeiro viajante que introduz o conceito de solidão de uma cidade - calculada como a distância média dessa cidade a todas as outras - e o combina com ideias de outras variações de heurísticas do vizinho mais próximo. Tendo a mesma complexidade das heurísticas de vizinho mais próximo mais rápidas, o novo método conduz a melhores resultados que estas heurísticas, ultrapassando igualmente várias outras heurísticas reportadas na literatura. Uma característica interessante da heurística proposta é que dá prioridade a localizações mais isoladas na definição de rotas. A antecipação da distribuição de bens e serviços a localizações mais periféricas pode ser considerada uma externalidade social positiva, tornando a heurística passível de adopção por determinadas entidades por razões não meramente económicas mas também sociais.

Palavras chave: Caixeiro viajante, Heurísticas, Vizinho mais próximo

1. Introduction

The travelling salesman problem (TSP) is one of the most famous and well studied problems in combinatorial optimization. Being very simple to formulate, hard to solve and gathering an extensive body of literature around it, the problem is a very good candidate for the testing of algorithmic ideas that can be easily compared to existing approaches.

Given a set of n nodes or cities and a matrix n×n of distances between them, the problem consists of finding the shorter (optimal) tour that goes through all the nodes and returns to the first without passing twice through the same node. In this paper, we will consider the symmetric TSP, where the distance (a,b) equals (b,a) for every entry of the distance matrix.

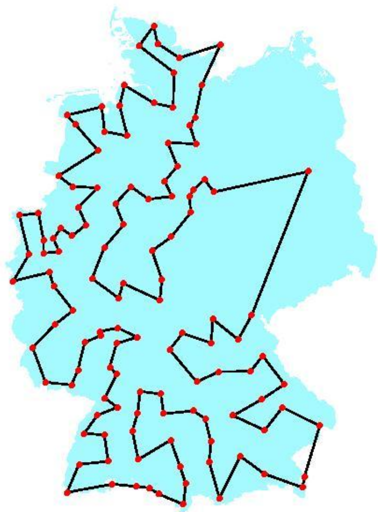

Figure 1 – 120 Western German Cities Optimal Tour found by Groetschel [Georgia Tech 2005]

This problem has a wide range of applications, from logistics and transportation, such as organizing the school bus routes to pick up children in a school district, to the scheduling of machines to drill holes in an electronic circuit board [Georgia Tech 2005].

In spite of being very simple to state, the problem is very hard to solve (except for a minimum number of cities) since the number of solutions to be tested grows very fast with n. In fact, the total number of routes that one can think of is given by n!/2. If one has 100 cities (which is considered a very small problem) to deal with, the number of tours is given by 100!/2 which is approximately $(9.33/2) \times 10^{157}$. Testing such a high number of solutions to find the best one, is impractical.

So far, no fast algorithm has been developed that solves the problem. By fast, we mean an algorithm with a polynomial time complexity, i.e. an algorithm with a running time that grows proportionally to some power of n as n becomes larger. Its complexity led to the classification of the travelling salesman problem as an NP-Hard problem (NP stands for non-polynomial). In fact, if such polynomial time algorithm were found, that would mean that this class of problems was not so difficult after all, and P (Polynomial) = NP (Non-polynomial). Such an unlikely discovery (most scientists think that P≠NP) would solve one of the Millennium prize problems worth 1 million dollars and awarded by the Clay Mathematical Institute.

In spite of its difficulty, the travelling salesman problem research has experienced major developments when it comes to the number of nodes that can be tackled, or the variety of methods being applied and their results. Some methods (exact methods) aim at finding the optimal but since that is not viable for problems with too many nodes or cities, another class of methods, entitled heuristics, have flourished which try to get as close as possible to the solution.

There are many heuristics, all of them reaching a balance between the quality of the solution quality and the time to compute it. The simplest heuristics, as the one that we will propose in this paper, are usually faster and lead to more modest solutions. They belong to the class of tour construction heuristics.

2. Tour construction heuristics

Tour construction heuristics are important whether as simple ways of obtaining low cost good solutions (given their faster procedures), or as a means of delivering initial solutions to be improved through the application of the more sophisticated "tour improvement heuristics" - such as local search heuristics, genetic algorithms or taboo search, among others. Results have been reported of how, in general, tour construction heuristics enhance the performance of other heuristics – when used in conjunction with them [Tsai et al 2004], [Hwang et al 1999], [Pertunnen 1994].

Tour construction heuristics are characterized by progressively building a tour from the start, through a sequence of steps, until a valid solution is reached without ever trying to improve such solution. On the other hand, tour improvement heuristics start with a valid tour and try to reduce its cost provided that the new route remains valid.

In table 1 we present a brief taxonomy of TSP tour construction heuristics.

Table 1 – TSP main tour construction heuristics taxonomy [Johnson and McGeoch 2002] and [Nilsson]

Heuristics that grow fragments / Pure augmentation heuristics. (heuristics that construct routes merely by adding one edge at a time to the tour, and making the choice of the edge to be added based on its length [Johnson and McGeoch 2002])	
Name	Basic Idea
Nearest Neighbour Heuristics	Start in one city and keep finding and adding to the tour the next nearest unvisited city.
Multiple Fragment Heuristic (greedy heuristic) and variants such as Boruvka	Repeatedly, select and add to the route the shortest edge provided that no node shows up twice and no cycles are created.

<table>
<tr><td>Savings Heuristic</td><td>Start with a pseudo-tour consisting of a multigraph that has two edges from an arbitrary central city to each of the other cities. Then, successively, look for the best way to shortcut this graph by replacing a length-2 tour from one (non-central) city to another by a direct link.</td></tr>
<tr><td colspan="2">Heuristics that grow tours
(more complex heuristics that still build tours incrementally but not, anymore, solely on the basis of each edge length)</td></tr>
<tr><td>Name</td><td>Basic Idea</td></tr>
<tr><td>Nearest Insertion (NI) and its variants Nearest Addition (NA) and Nearest Augmented Addition (NA+)</td><td>Start with a partial sub-tour and keep inserting the nearest neighbour to any of the cities in the sub-tour according to the following rule:
NI-Insert between two consecutive cities such that the insertion causes the minimum increase in tour length
NA-Insert next to the nearest neighbour on the side (before or after) that causes the minimum increase in length
NA+-Insert as in NI but restrict attention to pairs of consecutive cities where at least one is no further from the city to insert than twice the distance to its nearest neighbour in the tour.</td></tr>
<tr><td>Random Insertion and its variants (random addition and random augmented addition)</td><td>Basically, these heuristics differ from Nearest variants in the choice of the city to add which, in this case, is simply chosen randomly. The initial tour starts with two maximally distant cities.</td></tr>
<tr><td>Farthest Insertion and its variants (farthest addition and farthest augmented addition)</td><td>Basically, these heuristics differ from Nearest variants in the choice of the city to add which, in this case, is simply chosen as the city whose minimal distance to a tour city is maximal. The initial tour starts with two maximally distant cities.</td></tr>
<tr><td>Cheapest Insertion and its variants (as Convex Hull)</td><td>Start with a partial tour and for each city, not in the sub-tour, find an edge in the sub-tour such that, inserting the city between its ends, would lead to the minimum increase in tour length. After evaluating that for each city, insert the one leading to minimum increase in tour length and keep repeating the procedure.</td></tr>
<tr><td colspan="2">Heuristics based on trees</td></tr>
<tr><td>Name</td><td>Basic Idea</td></tr>
<tr><td>Minimum Spanning Tree heuristic</td><td rowspan="2">These heuristics start by finding a minimum spanning tree which is a tree that connects all the cities with minimum cost. Such tree is the basis for a TSP tour in which to visit all cities requires that some are visited more than once. The idea then is to depart from such tour and somehow avoid going back to the same cities using a shortcutting strategy.</td></tr>
<tr><td>Christofides heuristic</td></tr>
</table>

We focus our attention in the nearest neighbour heuristics which belong to the group of tour construction heuristics by pure augmentation. We are searching for an equally intuitive and time consuming heuristic as the nearest neighbour but that leads to better tours.

2.1 Nearest neighbour heuristics

Nearest neighbour

Among the tour construction heuristics by pure augmentation, the nearest neighbour heuristic is the most obvious one. Surely, its popularity relies on being very intuitive and simple to implement. The procedure consists of choosing one initial starting node and progressively adding to the route the node closest to the one previously added until all nodes are included in the tour. The time complexity of this procedure is $O(n^2)$ since basically, for each node out of n nodes, one has to search the other n nodes to figure out which one is the closest (in practice, one just has to search the nodes are still not included in the tour, but this is an approximate complexity measure). Therefore, one considers that the time complexity ~ $n*n= n^2$.

Double-ended nearest neighbour

Since after step two, the route under construction with this heuristic has two nodes at its ends, a first variation of the heuristic is to consider the new nodes closer to each of the route's ends and add to the tour the one that is closer to the route's respective endpoint. This way, the route grows with successive augmentations to both of its ends. This heuristic is known as the double-ended nearest neighbour and its time complexity is twice the previous one, which in polynomial terms still means a quadratic time complexity.

Repetitive nearest neighbour

Since the quality of the tours obtained depends on the initial node considered, another variation of this heuristic is the repetitive nearest neighbour, which computes the tours obtained through the application of the nearest neighbour heuristic for every starting node and chooses the best route among all of them. As expected, this heuristic leads to better tours. However, since it computes n nearest neighbour heuristics, its complexity is now cubic ($O(n^3)$) which means that it is out of the scope of the quadratic complexity we are looking for (the best that seems to be possible when working with distance matrices).

Improved nearest neighbour

Another lesser known variation of the nearest neighbour heuristic consists of determining and selecting the shortest edge of the distance matrix as the tour starting edge and then proceed exactly as in the nearest neighbour to include the following nodes. This favourably solves the problem of not knowing which node to start with and the proponents of such idea refer that, statistically, it leads to better results than the nearest neighbour method, naming it as the improved nearest neighbour method [4], of complexity $O(n^2)$.

Therefore, in summary, we have the following nearest neighbour heuristics, and we will benchmark our own against the ones with the shortest time complexity ($O(n^2)$):

Heuristic	Time Complexity
Nearest neighbour	$O(n^2)$
Double-ended nearest neighbour	
Improved nearest neighbour	
Repetitive nearest neighbour	$O(n^3)$

In spite of its interest and widespread use to provide initial travelling salesman tours subject to further improvement, the nearest neighbour heuristic and its variations that were presented so far, suffer from a major problem. Their greedy nature of systematically and solely trying to reach the next closest node leads to the postponement of the connection of more distant cities to the route. As a consequence, later in the tour construction, several cities still remain that are quite apart from each other, forcing the method to include them at a higher cost. Figure 2 illustrates this problem in a route definition example produced with the nearest neighbour heuristic, starting at node 1. The desirable outcome of a heuristic application to the example would be something more like the tour in Figure 3.

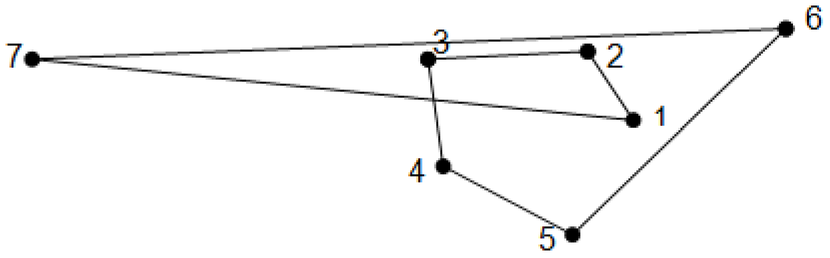

Figure 2 – Tour resultant of nearest neighbour heuristic application starting at node 1

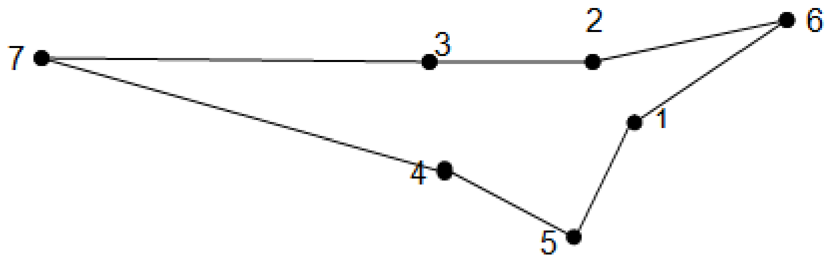

Figure 3 – A more desirable outcome of a TSP heuristic application starting at node 1

The difference between the routes in Figures 2 and 3, lies on the decisions that had been taken in nodes 1 and 3.

Considering Figure 2, in node 1, for example, the nearest neighbour simply attends to the closest node which is node 2, ignoring node 6 which will have to be added later to the route at a higher cost. The good decision would be the one illustrated in Figure 3, where starting at node 1, node 6 is given priority over node 2, because it clearly stands in a more distant location from other nodes, making it a risky decision to postpone its connection to the route.

This example suggests that, among nearby cities, a heuristic that gives some priority to those standing in more distant locations, would probably be more successful than the nearest neighbour heuristic.

2.2 Construction priority heuristic

We have found in the literature a tour construction heuristic that functions under a similar principle, considering tour construction priorities as a global concern for optimality in the choice of the next city to add to the route. The construction priority heuristic [Hwang et al, 1999] elaborates matrices (or lists in a second version of the heuristic) of each city's neighbours that are closer than a certain distance and then gives priority to the cities that have less neighbours in that matrix, and to those with more distant neighbours in the matrix.

However, the previous heuristic is rather more complex than the nearest neighbour. It requires the creation and evaluation of such neighbour matrices or lists. This resulted in a time performance decrease when compared to the nearest neighbour.

Anyway, the construction priority heuristic does not obey to the criteria for a nearest neighbour heuristic (it is not even a pure augmentation heuristic) and therefore it does not fulfil our demand for a new nearest neighbour heuristic that outperforms the nearest neighbour variations with smaller time complexity without compromising time complexity.

2.3 Modified nearest neighbour heuristic

After developing the major new idea for the heuristic to be presented in this paper, we have found a paper that mentions another nearest neighbour variation, which, although formulated in different terms, includes basically the same new criteria for adding a new edge to the route, considered hand in hand with the nearest neighbour.

Its author named it as the modified nearest neighbour method [Mehendale 2008]. The selection of an edge to the tour obeys two criteria:

- The cost of the edge that gets included in the route is minimum
- The cost of the edges that get excluded out of the route is maximum

It is suggested that these criteria be observed in any order and at any stage of the selection. The first of the two criteria above corresponds to the selection of the edge directed towards the city closest to the sub-tour endpoint, just like in the nearest neighbour heuristic.

The second criteria is equivalent to the idea behind the heuristic that we will propose next that of favouring more distant cities, since it is equivalent to stating that one should select a certain edge only if the sum of the costs of all the edges finishing in its endpoint is maximum. This is the same as saying that the cost of the edges that get excluded from the route is a maximum, since no two edges can have its termination in the same endpoint, and that choosing one edge eliminates n edges, being n the number of cities. And, if the sum of the costs of all the edges finishing in an edges' endpoint is maximum, that corresponds to give priority to the cities that sum a greater distance to all other cities.

However, a simple idea was lacking to enunciate such criteria and there were no experimental results testing its effectiveness. Also, the heuristic we propose next incorporates some other ideas from other nearest neighbour heuristic variations, which makes it more effective than the modified nearest neighbour heuristic. To distinguish the following heuristic from this one, there is also the fact that the heuristic we propose works solely on the basis of a distance matrix (modified from the initial one) which makes it particularly simple and comparable to the nearest neighbour in terms of time performance.

3. Double-ended nearest and loneliest neighbour heuristic

3.1 Nearest and loneliest neighbour

As we have introduced previously, the basic idea behind the heuristic that we propose in this paper is that cities more distant from others should be given some priority in the tour construction to avoid its later inclusion in the route at a higher cost. To make it possible we introduce the concept of loneliness of a city, computed from the average distance of that city to all others.

Together with the distance to the closest neighbours, the loneliness of the closer neighbours will be also criteria for selecting the next node to be added to the route. Lonelier neighbours will be preferred over the others.

In a pre-processing step that runs in a negligible time, a new distance matrix is obtained such that shorter new distances from a city to others are an equally weighted function of both shorter old distances to those cities and a higher loneliness of that city.

Such pre-processing step is done through the following C code:

```
/*1-Starting from x and y Eucledian Coordinates (given for the problem) load distances
between city pairs for the original distance matrix and store it in a 2-D array*/
for(i = 0; i < NumberofCities; i++)
   for(j = 0; j < NumberofCities; j++) array [i][j]= sqrt(pow((x[i]-x[j]),2)+pow((y[i]-y[j]),2)) ;

/*2-Calculate the distance of each city to all others naming its value as distset*/
for(i = 0; i < NumberofCities; i++)
{  distset[i]=0;
    for(j = 0; j < NumberofCities; j++) distset[i]= distset[i] + array [i][j] ;
}

/*3-Calculate the minimum, maximum and average (between both) of the distances of each
city to all others*/
min_distset = distset[0]  ;
for(i = 1; i < NumberofCities; i++) if (distset[i] < min_dist) min_distset=distset[i] ;
max_distset = distset[0];
for(i = 1; i < NumberofCities; i++)  if (distset[i] > max_distset)  max_distset=distset[i] ;
average_distset =  (max_distset + min_distset)/2   ;

/*4-Update the old distances of each city to all others such that higher distances (compared to
the average) are proportionally rewarded with smaller new distances*/
for(i = 0; i < NumberofCities; i++)
   if (distset[i] > average_distset) distset[i]=average_distset-(distset[i] - average_distset);
    else distset[i]=average_distset+(average_distset - distset[i] );

/*5-Calculate the new distance matrix from the combination of the two criteria*/
for(i = 0; i < NumberofCities; i++)
   for(j = 0; j < NumberofCities; j++) array [i][j]=((NumberofCities*array[i][j])+ distset[j])/2;
```

In synthesis, the calculation of the new distance Matrix previously detailed contemplates the following operations:

Distance Matrix pre-processing:

i) Calculate the average of the distances of each city to all others (sum the distances and divide by n)

ii) Calculate the minimum, maximum and average (between both) of the average distances of each city to all others

iii) Calculate the symmetric of the distance of each city to all others with respect to the average calculated in the previous step (this will guarantee that a higher loneliness is rewarded with a shorter cost in the distance matrix). Keep the matrix of those symmetric values.

iv) Calculate the new distance matrix when each new entry is the average between its old entry (the initial cost) and the respective entry of the matrix obtained in the previous step.

Based on the new metric, the algorithm proceeds as a nearest neighbour heuristic that starts with including the shortest edge among all (as in the improved nearest neighbour heuristic). The advantage of using the improved nearest neighbour idea in the new heuristic is that one favourably solves the problem of not knowing which distance matrix entry to include first. The algorithm proceeds the following way, working with the new distance matrix:

Nearest and loneliest neighbour heuristic:

i) Find the shortest edge and take it as the first tour edge, selecting one of the two nodes as the starting node

ii) Add to the tour the node closer to the route’s endpoint provided that such node is not already part of the route

iii) Proceed with the previous step until all nodes are part of the route

iv) Return to the starting node by adding it to the end of the route

3.2 Double-ended nearest and loneliest neighbour

An improvement to the previous heuristic is obtained through its combination with the double-ended nearest neighbour heuristic as the experimental results show.

Therefore, the final heuristic that we propose as the most effective nearest neighbour heuristic of quadratic complexity, is the following, working with the pre-processed new distance matrix obtained according to the nearest and loneliest neighbour heuristic description.

<u>Double-ended nearest and loneliest neighbour heuristic:</u>

i) Find the shortest edge and take it as the first tour edge

ii) Consider the nodes closer to each of the route's ends and add to the tour the one closer to the route's respective endpoint provided that such node is not already part of the route

iii) Proceed with the previous step until all nodes are part of the route

iv) Return to the starting node by adding it to the end of the route

The implementation of the new heuristic in C Programming language and the submission of the code to a Borland Studio C++ Compiler, led to the results presented in Table 2 for 11 DIMACS Implementation Challenge Site [Johnson et al 2008] known instances.

The results for the nearest neighbour and double-ended nearest neighbour heuristics were collected from the DIMACS Implementation Challenge Site. To make explicit the contribution of the double-ended idea to the new heuristic we have included the nearest and loneliest neighbour heuristic in separate without such contribution.

Following the improved nearest neighbour idea, nearest and loneliest neighbour and double-ended nearest and loneliest neighbour, both have been included in the improved version (first edge is the shortest).

Table 2 – Gap (in distance) to the optimum of the final routes obtained through each nearest neighbour heuristic

Instance	n (number of cities)	Heuristic				
		Nearest neighbour (NN)	Double-ended nearest neighbour (DENN)	Nearest and loneliest neighbour (NLN)	Double-ended nearest and loneliest neighbour (DENLN)	Optimum
eil101	101	779	825	698	702	629
gil262	262	2882	2904	2710	2689	2378
pr1002	1002	332679	317056	315040	315172	259045
u1060	1060	299527	*	267358	269258	224094
vm1084	1084	299538	*	291698	285723	239297
pcb1173	1173	69752	69752	69037	68604	56892
d1291	1291	63753	*	62527	62469	50801
nrw1379	1379	69982	70163	66284	65928	56638
fnl4461	4461	231585	226838	212557	211150	182566
brd14051	14051	584403	578741	552133	550624	469385
d15112	15112	1948107	1937410	1866300	1851043	1573084

*Data not available in the DIMACS Implementation Challenge Site

Table 3 – Gap (in percentage) to the optimum of the final routes obtained through each nearest neighbour heuristic

Instance	N (number of cities)	Heuristic				
		Nearest neighbour (NN)	Double-ended nearest neighbour (DENN)	Nearest and loneliest neighbour (NLN)	Double-ended nearest and loneliest neighbour (DENLN)	Optimum
eil101	101	23.85	31.16	10.97	11.61	0.00
gil262	262	21.19	22.12	13.96	13.08	0.00
pr1002	1002	28.43	22.39	21.62	21.67	0.00
u1060	1060	33.66	*	19.31	20.15	0.00
vm1084	1084	25.17	*	21.90	19.40	0.00
pcb1173	1173	22.60	22.60	21.35	20.59	0.00
d1291	1291	25.50	*	23.08	22.97	0.00
nrw1379	1379	23.56	23.88	17.03	16.40	0.00
fnl4461	4461	26.85	24.25	16.43	15.66	0.00
brd14051	14051	24.50	23.30	17.63	17.31	0.00
d15112	15112	23.84	23.16	18.64	17.67	0.00

*Data not available in the DIMACS Implementation Challenge Site

The results show that DENLN heuristic is the one leading to the best results overall. In 8 of the 11 instances, including all the 5 larger ones, DENLN outperforms NLN showing that the double ended idea combination with NLN compensates.

In all the 11 instances, any of these two heuristics (NLN and DENLN) dominate the Nearest Neighbour and Double-ended Nearest Neighbour heuristics.

Although the experimental tests showed that the DENLN matrix pre-processing runs in negligible time when compared to the following step, one might consider that this pre-processing has a time complexity of $O(n^2)$, which added to the time complexity of $O(2n^2)$ of the double-ended nearest neighbour and the time complexity of $O(n^2)$ of the improved nearest neighbour, totals a time complexity of $O(4n^2)$ which is still ~ $O(n^2)$.
Therefore, the new heuristic DENLN is still in the time complexity class of nearest neighbour faster variations, dominating all of them in the tour quality.

Still according to the DIMACS Implementation Challenge Site [Johnson et al 2008], and besides outperforming the faster nearest neighbour heuristics, the new heuristic reported seems to dominate several other tour construction heuristics of the more advanced class of tour construction heuristics that grow tours such as Nearest Insertion, Nearest Addition, Nearest Augmented Addition, Farthest Addition and Random Addition, since none of these heuristics is of a smaller complexity than our own and they lead to worst results in practically all of the referred instances. However, there is one tour construction heuristic that grows fragments that seems to dominate our own, which is the Farthest Insertion heuristic, since it shares the same time complexity but systematically seems to lead to better results.

We would like to finish the presentation of the heuristic pointing out what might be one significant feature from a certain point of view.

As soon as it approaches the vicinity of a few cities, the new heuristic will tend to favour the lonelier ones, provided they are not too far away. Therefore, besides its interesting performance strictly as a TSP heuristic that searches a tour as short as possible, DENLN is also a social heuristic. We all know how living away from the centres difficult the access to some benefits such as having a doctor available or buying that recent product. The fact that the heuristic gives some priority to lonelier locations in tour definitions might be considered a positive social externality. The application of the heuristic to schedule transportation will partly contribute to an earlier distribution of goods and services to more peripheral sites, which makes it appealing to public or private institutions that are engaged in acts of social responsibility.

However, it should be taken into account that isolated sites are not always those with higher distances to other cities. A city can grow in between two city centres and still be quite isolated. Therefore our assumptions are just an approximation of reality.

An interesting goal to pursuit would be to quantify the extent to which does the heuristic benefit the more isolated sites in comparison with other methods.

This heuristic might be also particularly useful for users less demanding in terms of solution optimality but keener on understanding and being able to explain the method they are using.

4. Conclusions

Following the presentation of the simplest travelling salesman problem heuristic (nearest neighbour) which, in spite of its shortcomings, remains popular for being intuitive, fast, easy to implement, and delivering solutions prone to improvement by other heuristics, we have mentioned two other heuristics that elaborate on the idea of favouring more isolated cities among those close to the sub-tour endpoints. These heuristics however, whether for their complexity - in the case of the construction priority heuristic, whether for not being tested, formulated in simple terms and subject to further improvement – in the case of modified nearest neighbour, did not respond adequately to the purpose of finding and testing the most performing nearest neighbour heuristic of time complexity ~ $O(n^2)$.

The double-ended nearest and loneliest neighbour (DENLN) heuristic reached such a goal and showed an interesting performance when taking into account its short time complexity and its domination in several other tour construction heuristics. An interesting feature of the heuristic is that by favouring more peripheral cities, it contributes to an earlier than usual transport delivery to isolated sites, in the context of logistics and transportation. It might therefore be considered as contributing, though modestly, to fight the localization handicap of some sites, making it a social friendly heuristic.

Acknowledgements and final comment
We acknowledge and thank the reviewer's comments and suggestions that significantly helped to improve the paper.

The analyses, opinions and findings of this paper represent the views of the author; they are not necessarily those of the Banco de Portugal or the Eurosystem.

REFERENCES

Chen-Fa Tsai, Chun-Wei Tsai, Ching-Chang Tseng, 2004. A new hybrid heuristic approach for solving large travelling salesman problem. Information Sciences – Informatics and Computer Science: An International Journal, Volume 166, Issue 1-4,pp. 67-81.

Georgia Tech, 2005. http://www.tsp.gatech.edu, Last updated 01/2005, accessed in 03/11/2011.

Hwang, C.P., Alidaee, B., Johnson, J.D., 1999. A Tour Construction Heuristic for the Travelling Salesman Problem. The Journal of the Operational Research Society, Volume 50, Number 8, pp. 797-809.

Johnson, D. and McGeoch, L., 2002. The travelling salesman problem and its variations, chapter 1 - Experimental analysis of heuristics for the STSP, pp 369-443. Kluwer Academic Publishers.

Johnson, D. et al, 2008. http://www2.research.att.com/~dsj/chtsp/, Last Updated in 11/12/2008, accessed in 03/11/2011

Kratica, J., Radojevi, S., 1995. One improvement to nearest neighbour method for solving travelling salesman problem. http://masters.donntu.edu.ua/2009/kita/aleksandrova/library/article05.pdf, accessed in 03/11/2011

Mehendale, D., 2008. Hamiltonian Graphs and the Traveling Salesman Problem. arXiv:0704.3672v5 [math.GM], pp. 8-10.

Nilsson, Christian. Heuristics for the Travelling Salesman Problem. http://www.ida.liu.se/ ~TDDB19/ reports_2003/ htsp.pdf, accessed in 15/01/2012.

Pertunnen, Jukka, 1994. On the significance of the initial solution in travelling salesman heuristics. The Journal of the Operational Research Society, Volume 45, Number 10, pp. 1131-1140.

Fernando Guilherme Silvano Lobo Pimentel nasceu em 05/12/1971, é licenciado em Engenharia Física Tecnológica pelo IST e mestre em Investigação Operacional e Engenharia de Sistemas também pelo IST. Trabalha actualmente em sistemas de informação no Banco de Portugal tendo sido por dois anos adjunto do Secretário de Estado da Ciência, Tecnologia e Ensino Superior. Tem interesses de investigação na área da matemática discreta e aplicações.