

Document downloaded from:

<http://hdl.handle.net/10251/37058>

This paper must be cited as:

Rodríguez Villalobos, A.; Ruiz García, R. (2012). The effect of the asymmetry of road transportation networks on the traveling salesman problem. *Computers and Operations Research*. 39(7):1566-1576. doi:10.1016/j.cor.2011.09.005.



The final publication is available at

<http://dx.doi.org/10.1016/j.cor.2011.09.005>

Copyright Elsevier

The effect of the asymmetry of road transportation networks on the Traveling Salesman Problem

Alejandro Rodríguez^{1*}; Rubén Ruiz²

¹ Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática, Universitat Politècnica de València, Pza. Ferrándiz Carbonell 2, 03801 Alcoy, Spain.

arodriguez@doe.upv.es

² Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática, Universitat Politècnica de València, Camino de Vera s/n, 46021 Valencia, Spain.

rruiz@eio.upv.es

August 22, 2011

Abstract

The routing of vehicles on road transportation networks is an area of great importance to transportation planners within scientific literature. This field includes well known and studied problems like traveling salesman problems or TSP or the more realistic asymmetric variant or ATSP, whose applications extend to other areas of transport and operations research. This work studies the effect that the asymmetry of road transportation networks, geographical location and territory have over *TSP* and *ATSP* methods. We conduct comprehensive experiments in order to assess the effects that these factors have on some of the best known algorithms for the *TSP/ATSP*. We demonstrate that all these factors have a significant influence in solution time and quality. Furthermore, we show that the solutions obtained with Euclidean matrices and those obtained with real distance matrices differ significantly.

Keywords: Asymmetry, Asymmetric Traveling Salesman Problem, Algorithms, Geographic Information Systems, road transportation networks

*Corresponding author. Tel/Fax: +34 96 652 84 89

1 Introduction

The Traveling Salesman Problem, or *TSP* for short, is one of the most well known and thoroughly studied combinatorial optimization problems (Lawler et al., 1985). The objective is to find the minimum cost (usually minimum distance) route visiting a set of n locations, where each location is visited exactly once. The tour must start and finish at the same location. A solution to the *TSP* problem is represented by a permutation of the n locations. The *TSP* is a well known \mathcal{NP} -Hard problem.

In routing problems, and more precisely, in the *TSP* (symmetric and asymmetric), there is a distance or cost matrix. Each element in the matrix contains the travel distance, time, or any other cost function between any two locations o, d , where $o, d \in n, o \neq d$. Usually, travel time, speeds and costs are a function of the distances between locations or nodes.

A wide range of the research work on the *TSP* can be applied to other discrete optimization problems, and to applications in several fields such as genome reconstruction, scheduling operations, machine movements for hole drilling in circuit boards or other objects, etc. and of course, its application in the routing problems and transport like the routing of aircrafts, ships, school buses, etc. (Gutin and Punnen, 2002). As is well known, the Asymmetric Traveling Salesman Problem (*ATSP*) where the distance matrix is not necessarily symmetric, is a more general case than the Symmetric Traveling Salesman Problem (*STSP*) where the cost or distance matrix is always assumed to be symmetric. In related scientific literature these two versions of the *TSP* are often investigated independently, with a strong bias towards the second. Apart from the list of n locations, the input data for the *TSP* is just the distance matrix. As a result, carefully estimating distances between nodes is extremely important. The need for real matrices and distances has been highlighted several times in the *TSP* literature (Flood, 1956), and also for Vehicle Routing Problems (*CVRP*) (Clarke and Wright, 1964), or for other variants as well (Toth et al., 2001). Although in some forms of transportation by air, sea and train the Euclidean distances (symmetric *TSP*) can be a reasonable approximation; in other cases, the Euclidean distances may constitute a gross underestimation of reality, especially in urban transportation, where distances on road networks can be highly asymmetric.

In this work we deal with the issue of asymmetry in the distance matrix from road transportation networks. The main objective of this research is to measure the effect that the asymmetry of the road network has in solving the *TSP*, both the symmetric as well as the asymmetric variant. As we will demonstrate and measure, symmetric solutions –those obtained with symmetric and Euclidean distance matrices– have little in common with regard to sequence and total distance with real solutions (those obtained with asymmetric and real distances). There is usually a very significant difference between the solution of an asymmetric instance and the solution of a symmetric one based on an approximation of the real distances. Not only is the total distance significantly different, but so is the sequence of nodes in the solution. Furthermore, different state-of-the-art methods for the *STSP* and *ATSP* are shown to differ in effectiveness and in efficacy when tested

against asymmetric real distances, compared against original performance in Euclidean settings. Some methods even no longer work when faced with asymmetric matrices. However, it is not the intention of this paper to carry out a comparison about state-of-the-art methods. Some other interesting factors that also affect the level of asymmetry and the performance of *STSP* and *ATSP* methods, like territory, geographical location and problem size are also studied. More precisely, this paper addresses the following research questions: What is the effect of the asymmetry over the effectiveness and efficiency of the main *TSP/ATSP* heuristics? Is it feasible to reduce the Asymmetric Traveling Salesman Problem to a symmetric one? How do all the factors behave for different problem sizes? What is the most adequate heuristic in each case?

The remainder of this paper is organized as follows; Section 2 further substantiates the importance of considering asymmetry in routing problems. Section 3 elaborates on the research questions and hypotheses, together with the studied factors and variables, experimental design and computational tests. Section 4 presents a thorough analysis of the different results from many perspectives, like CPU times, quality of the solutions and quantitative and qualitative comparisons. Finally, the conclusions of this work are presented in Section 5.

2 The real world is asymmetric

Given a *TSP* instance with n locations or nodes, the distance matrix between any possible pair of nodes o, d , where $o, d \in n, o \neq d$, is denoted by $C_{[n \times n]}$ and is a square matrix where the diagonal is usually disregarded. This matrix has $n \times (n - 1)$ elements with all the distances. In the vast majority of the routing literature, the locations or nodes are determined by their coordinates in a 2D plane and the distances between each pair of nodes are calculated by the simple Euclidean distance, given by the Pythagorean formula. In this case, it is straightforward to see that the distance between the nodes o and d is the same as the distance between d and o , i.e., $c_{od} = c_{do}, \forall o, d \in n, o \neq d$. In this case, the matrix C can be summarized by an upper or lower triangular matrix with $\frac{n \times (n-1)}{2}$ elements. A slightly more elaborated approach for obtaining the matrix C is to calculate the orthodromic distance between the geolocations of two nodes. Basically, the orthodromic distance is the shortest distance between any two points on the surface of a sphere, measured along a straight path on the surface of the sphere itself. This is often referred to as the great-circle distance. Orthodromic distances are also symmetric in nature. Note that orthodromic distances are much more accurate than Euclidean distances when measuring long distances in Earth as Euclidean distances would traverse the Earth nucleus, not considering the Earth's curvature.

It has been known for many decades (Daganzo, 1984) that Euclidean or orthodromic distances have little resemblance to real distances between nodes or locations that are linked through transportation networks or roads. As a matter of fact, the Euclidean or orthodromic distance is a very loose and weak lower bound of the shortest path that communicates any two nodes in a

transportation network. Furthermore, when one considers the nature of traffic, one-way streets and the intricate layout of most roads, it is straightforward to see that, to some degree or another, real distance matrices are not symmetric. This degree of asymmetry cannot be easily estimated as it varies widely according to different factors. Long distances are likely to be more symmetric due to two-way highroads. However, connecting locations in the historical centers of some big cities is likely to return asymmetric distances.

The usage of Euclidean or orthodromic distances is simply motivated by the large cost and difficulty in obtaining the real distances matrix C . Even nowadays, one needs to calculate $n \times (n - 1)$ shortest paths, each one constituting an enormous effort as real transportation networks, for example inside a country, typically contain billions of nodes and arcs. Geographic Information Systems (GIS) and geo-spatial databases, along with their Advanced Programming Interfaces (API) facilitate, to some extent, this herculean task. In any case, this possibility is relatively recent as rich GIS systems capable of doing such calculations have only existed in the mainstream market since the mid 1990s. Before this date, and since the early 1970s, researchers have tried to calculate the real distance matrix indirectly from the Euclidean or orthodromic one. For example, some researchers tried to estimate real distances after multiplying the orthodromic matrix by a given factor (Christofides and Eilon, 1969). Other works developed some functions to estimate real distances (Love and Morris, 1972). This idea was further exploited by other authors that developed distance estimation functions depending not only on the zone where nodes are located, but also on total traveled distance (Daganzo, 1984). Many problems arise when using these functions. The proposed functions have to be adjusted mathematically and empirically (which more or less implies some validation, that in turn needs some real distance matrices). This adjustment process is objective function dependent and also depends on the precision desired. Other authors demonstrated that this adjustment process is also dependent on the territory and other characteristics like geometry of the zone, type of transportation network, orographic accidents, natural obstacles and the like (Love and Morris, 1988; Dubois and Semet, 1995). Therefore, distance estimation cannot be carried out over the basis of a single function or without a deep and careful study, including parameter adjustment. While we do not advocate that such functions are not useful in any environment (some strategic decisions with aggregated information might benefit from such functions, where some degree of approximation is accepted), we support the idea given in Love and Morris (1988) that such functions are not acceptable in real operational settings.

The fact is that the existing literature has usually been concerned more with the symmetric *TSP* than with its asymmetric counterpart. It is clear that the first is a simpler, and more basic, problem. The best known polynomial-time heuristic with a known quality guarantee for the Euclidean *TSP*, due to Arora, 1998, is able to guarantee a $1 + \varepsilon$ approximation factor with respect to the optimal solution, for any fixed error factor $\varepsilon > 0$. In contrast, the best known polynomial-time heuristic with a known quality guarantee for the *ATSP* (given in Asadpour et al., 2010)

can only guarantee a $\log n / \log \log n$ approximation factor, where n is the number of locations. It is evident that the *ATSP* is a much more difficult problem, and as commented, a significant portion of the *TSP* literature considers Euclidean distances without even raising the issue of real distances. There is also a rich literature on the *ATSP* generalization and formulations, as for example, the papers of Gouveia and Pires (1999) and Fischetti et al. (2003), among many others like Arora (1998), Bontoux et al. (2010) and Germs et al. (2012). However, authors do not actually study, to the best of our knowledge, in its full complexity, the effect that different degrees of asymmetry and factors affecting asymmetry have over solution methodologies. In order to cope with all these complexities, modern GIS systems must be employed (Goodchild and Kemp, 1990), together with a deep understanding of the effect of the asymmetry and other interesting factors over the calculation of real distance matrices and *TSP* resolution.

3 Studying the effect of asymmetry

As previously stated, we are interested in either confirming or refuting the following hypotheses: 1) Asymmetry strongly effects the effectiveness and efficiency of the main *TSP* and *ATSP* heuristics. 2) The location of the nodes in the real world generates different levels of asymmetry and therefore also conditions *TSP* methods. 3) It is not always feasible to reduce the *ATSP* to the *TSP* (Jonker and Volgenant, 1983) for solving real *ATSP* problems with *TSP* heuristics. 4) The size of the problem interacts with asymmetry and also affects *TSP* algorithms. In order to assess these hypotheses we carry out a complete comparative study of the different solutions provided by *TSP* methods, with real characteristics and dimensions as commented in Fischetti et al. (2003). A large set of *TSP* instances is generated to this end.

A full factorial experimental design is employed (Montgomery, 2009), where each generated problem instance is defined by a series of factors that are further described in the following.

3.1 Factors and instances generated

Territory: It is the geographical region where the instance is located. This region is bounded by a quadrant defined by two pairs of opposed geographical coordinates (latitude and longitude). This is a qualitative ordinal factor that has been tested at three variants, of increasing size, related with the Iberian peninsula (our area of interest), as shown in the leftmost picture at Figure 1. The three regions are referred to as short, medium and large distance, respectively.

In the short distance, locations are placed in the geographical area of influence of a big city. As a result, the minimum distances between pairs of nodes are conditioned by urban transportation networks (one-way streets, traffic circles, city center, etc.). Medium distance includes short distance plus larger distances entailing regional transportation through paths, regional roads, city communication rings, etc. Lastly, large distance territories are further conditioned by large distance roads, highways and inter-city communications.



Figure 1: Different territories in the Iberian peninsula (left). Example of an instance with locations following a radial distribution in a large distance territory (right).

Location: It is the placement of the nodes inside the territory. This can be random or might follow a given pattern. Three variants are defined for this nominal qualitative factor: random, grid and radial. Figure 2 shows some examples over a given territory. In the grid location distribution, the territory is divided into square zones. The node is placed at the center of each zone, albeit slightly displaced by a random vector. Radial distribution has a central location that services the remaining $n - 1$ nodes, which are radially distributed at an angular equidistance equal to $\alpha = 2\pi/(n - 1)$. Figure 1 (right) shows a map with 500 radially distributed locations in a large territory.

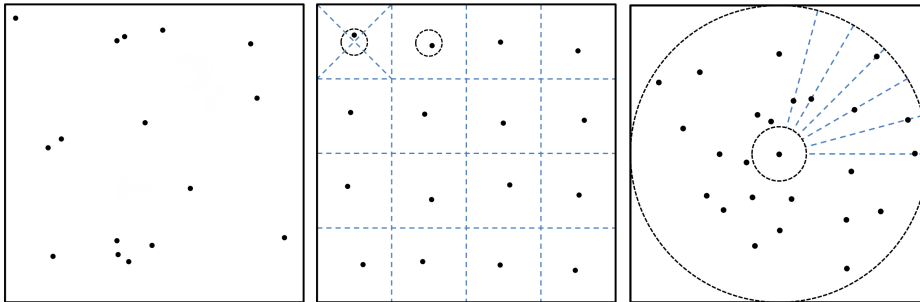


Figure 2: Examples of locations in random (left), grid (middle) and radial (right) distributions.

Number of nodes: This number n determines the size $n \times (n - 1)$ of the matrix or, $2n \times 2(n - 1)$ if it is transformed (the transformation process is detailed next). It is a quantitative factor with 10 levels: $n = \{50, 100, 150, \dots, 500\}$.

Symmetry: For each generated instance, the distance matrix C is calculated in different ways. This qualitative nominal factor considers strictly symmetric or asymmetric matrices with the following studied variants:

- Orthodromic: It is the symmetric matrix with great-circle distances.
- Asymmetric: Asymmetric matrix where the distances have been calculated with the aid of a GIS, i.e., distances are actually the shortest distances between locations as per the real network of roads and streets.
- Minimum arc from each pair: It is a symmetric matrix where distances have been extracted from the asymmetric matrix in a special way. Given any two distinct nodes o and d , the distance for the matrix is the minimum of the two ways, i.e., the distance satisfies $\min(c_{od}, c_{do})$. This results in a symmetric matrix.
- Maximum arc from each pair. Similar to the previous one but taking the maximum of the two ways: $\max(c_{od}, c_{do})$.
- Transformed: A symmetric matrix is constructed from the asymmetric one using a well known mathematical transformation due to Jonker and Volgenant (1983). This transformation is correlated with the number of nodes in the instance as the transformation multiplies the size of the distance matrix by a factor of four. Each location or node is split into two nodes, one real, and a second virtual node. The distance between a real node and its corresponding virtual sibling is set to a very small favorable cost (usually $-\infty$). This results in real and virtual nodes to be consecutively placed in the final *TSP* tour. The original asymmetric “from”-“to” ways are assigned to distances between real nodes in the transformed matrix whereas original asymmetric “to”-“from” distances (i.e., the way back distances) are assigned to the virtual nodes. All other possible distances are assigned a very unfavorable value ($+\infty$). A simple 3×3 symmetric matrix and its corresponding transformation are given in expression (1) below:

$$\begin{pmatrix} 0 & c_{12} & c_{13} \\ c_{21} & 0 & c_{23} \\ c_{31} & c_{32} & 0 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 0 & \infty & \infty & -\infty & c_{21} & c_{31} \\ \infty & 0 & \infty & c_{12} & -\infty & c_{32} \\ \infty & \infty & 0 & c_{13} & c_{23} & -\infty \\ -\infty & c_{12} & c_{13} & 0 & \infty & \infty \\ c_{21} & -\infty & c_{23} & \infty & 0 & \infty \\ c_{31} & c_{32} & -\infty & \infty & \infty & 0 \end{pmatrix} \quad (1)$$

All four factors, together with their corresponding levels or variants are gathered in Table 1.

As we can see, the last row of Table 1 contains the total number of levels or variants for each factor. Since we employ a full factorial experimental design, we have $3 \times 3 \times 10 \times 5 = 450$ treatments after combining all levels or variants. For each treatment, five different instances are generated, for a grand total of 2,250 *TSP/ATSP* instances. All these instances are publicly available at <http://soa.iti.es/problem-instances>.

Territory (T)	Location (L)	Number of nodes (n)	Symmetry (M)
Short distance	Random	50	Orthodromic (O)
Medium distance	Grid	100	Asymmetric (A)
Large distance	Radial	150	Minimum arc (P)
		...	Maximum arc (G)
		500	Transformed (T)
3	3	10	5

Table 1: Factors for the instances along with their levels and variants.

3.2 Response variables

As we will detail later, several state-of-the-art *TSP* methods are used for solving the proposed instances. Using Design of Experiments allows to study the effect that each considered factor (including the different algorithms) have over one or more response variables.

Solutions obtained after solving each instance are analyzed mainly at two levels: quantitative (mainly tour length) and qualitative (sequence of nodes or locations in the tour). As regards the last qualitative assessment, the literature is marred with papers that propose indicators for measuring the differences between solution objects, as for example Schiavinotto and Stützle (2007). In our case, measuring the differences between two *TSP* tours is commonly carried out by counting the number of $k - opt$ movements that are needed to transform one tour s into another s' . This needs a non-polynomial CPU time as a function of n . Therefore, we employ simpler measures of a distance d between two tours or $d(s, s')$:

Relative percentage deviation from the best solution found ΔS_i^* : It is the relative deviation (in percentage) of the tour length obtained after solving a given *TSP* instance i with algorithm A ($S_{i,A}$) from the lowest known tour length for that instance (S_i^*). It is calculated as follows:

$$\Delta S_i^* = \frac{S_{i,A} - S_i^*}{S_i^*} \cdot 100 \quad (2)$$

Hamming distance dH : It is a well known indicator that measures the differences between vectors, proposed by Hamming (1950). Basically, it takes two tours s and s' and adds 1 to the indicator counter each time a position in the tour is occupied by different nodes at both tours. For example, given $s = [2, 5, 3, 1, 4, 6]$ and $s' = [1, 2, 3, 4, 5, 6]$, the Hamming distance is 4. There is a problem as regards the *TSP* since the relative order of nodes in the tour is as important as their absolute positions. Take a second example $s = [6, 1, 2, 3, 4, 5]$ and $s' = [1, 2, 3, 4, 5, 6]$. In this case, the Hamming distance is 6, even though the route is almost the same (the only difference being the starting/ending node. However, this indicator is simple to calculate (it just requires $O(n)$ steps) it is easy to understand and to interpret.

Adjacency distance dA : Together with the Hamming distance, it makes sense to measure also the number of equal adjacent nodes between two routes s and s' , where the nodes need not

be located in the same absolute positions at the two tours. More specifically, this is achieved by checking if the arc between nodes e and $e + 1$ at solution $s -s(e, e + 1)-$ exists in any place of sequence s' . As a result, the adjacency distance counts the number of distinct arcs between two tours, with the maximum possible distance being $n + 1$. For example, given $s = [1, 2, 3, 4, 5, 6]$ and $s' = [1, 6, 2, 3, 4, 5]$, $dA = 3$, since arcs $(1, 6)$ $(6, 2)$ $(5, 1)$ of s' are not present at s . Using efficient data structures, dA can be calculated in $O(n)$ steps.

Note that in our measurements, both sequences are shifted so that vertex 1 is the first vertex in the sequence, in order to have a more precise measure of the distances (Hamming and Adjacency).

CPU time: It is the real elapsed CPU time that was needed when solving a given instance with an algorithm. This excludes input/output operations as well as all other system overheads, as detailed in Alba (2006).

Asymmetry in distance matrices: We are particularly interested in measuring the asymmetry degree of matrices. Strictly speaking, a matrix is asymmetric if it exists at least one pair o, d such that $c_{od} \neq c_{do}$, where $o, d \in n, o \neq d$. Furthermore, this is even true if $c_{od} = c_{do} + \varepsilon$, for any arbitrarily low value of ε . Obviously, this binary asymmetry indicator is not very informative and more precise indicators are needed. We employ the following alternatives: **Alfa** (α): It indicates the asymmetry degree by counting the number of asymmetric pairs of distances (pairs o, d that satisfy $c_{od} \neq c_{do}, o, d \in n, o \neq d$) over the total number of pairs n , using the $x_a(o, d)$ definition and expression (3) below, where a is a pair of arcs (od, do) . α takes values in the $[0\%, 100\%]$ interval. **Delta** (δ): It measures the asymmetry degree in more detail by actually looking at how different are asymmetric pairs (in distance). It is calculated with expression (4) below. **Weight:** It just sums all the distances of the C matrix, i.e., $\sum_{o=1}^n \sum_{d=1, o \neq d}^n c_{od}$. **Average weight** (\overline{Weight}): It relates the weight with the number of arcs.

$$x_a(o, d) = \begin{cases} 0 & \text{if } c_{od} = c_{do} \\ 1 & \text{if } c_{od} \neq c_{do} \end{cases}$$

$$\alpha = \frac{2 \sum_{a=1}^n (x_a)}{n^2 - n} \cdot 100 \quad (3)$$

$$\delta_a = \frac{|c_{od} - c_{do}|}{\min(c_{od}, c_{do})} \cdot 100 \quad \forall o, d \in n, \quad o \neq d \quad (4)$$

3.3 Solution process

For solving all instances, we employ a high performance computing cluster with 30 blades, each one containing 16 GBytes of RAM memory and two Intel XEON E5420 processors running at 2.5 GHz. Note that each processor has 4 physical computing cores (8 per blade). At this stage, it is worth mentioning the sheer computation effort needed for calculating real distance matrices (all instances where M=A as per Table 1), especially when compared against orthodromic matrices. 450 instances in the set of 2,250 contain real distances. These have been calculated by doing a

humongous number of shortest route requests between pairs of nodes to Google Maps. This took 196.5 single blade equivalent CPU days. This is in stark contrast with the 21 seconds needed for calculating the same matrices but with orthodromic distances.

A direct outcome of this computational effort is a large set of 450 *ATSP* instances where distances are actually real, corresponding to current transportation networks in Spain, following all previous factors already mentioned in earlier sections. This set is complementary to the well known TSPLIB95 dataset where only 19 synthetic *ATSP* instances can be found. These *ATSP* instances have random integer distances at each arc with n sizes between 17 and 443. As indicated, these instances are publicly available. Each instance is solved with a wide range of *TSP* heuristics:

- Nearest neighbor algorithm (A=NN) as described in Flood (1956). A simple heuristic, yet with reasonable performance.
- 2-Opt heuristic (Croes, 1958) (A=2O). A well known simple local search method.
- Concorde *TSP* solver¹ (A=CO). A very powerful state-of-the-art exact branch-and-cut algorithm for the *TSP*. It is described in Applegate et al. (2002). Parameters: *default options*.
- Lin-Kernighan heuristic of Lin and Kernighan (1973) (A=LK). One of the most well known powerful and well-known heuristics.
- Improved Lin-Kernighan of Helsgaun (2000) (A=HE). This is currently considered as one of the state-of-the-art methods for solving the *TSP*. Parameters: *author's recommendations*.
- Memetic algorithm of Nagata and Kobayashi (1997) and Nagata (2006) (A=NA). Also one of the most important and adaptive heuristics. Parameters: *10 trials, population size = 100, 30 children, 2 parents*.
- Branch-and-cut method of Fischetti et al. (2003) (A=FI). In our experiments, only for problem size $n = 50$. Parameters: *optimized compilation, internal limit of 150,000 branching nodes*.
- Improved *GKS/TBCOP* heuristics of Goldengorin et al. (2006) (A=GO), based on the Helsgaun code. Parameters: *author's recommendations*.

As we can see, the selection of *TSP* heuristics is motivated either by simplicity, asymmetry adaptation or by current state-of-the-art performance. Note that not all studied heuristics are capable of working over asymmetric matrices. For example, the LK and CO methods are specifically designed for the *TSP* and not for the *ATSP* (Applegate et al., 2006). In these cases, the transformed matrix (M=T) is employed instead of the real asymmetric one. This results in 2,250 instances \times 8 algorithms – 450 asymmetric matrices \times 2 non-*ATSP* heuristics (LK and CO)

¹<http://www.tsp.gatech.edu/concorde.html>

–450 (M=T unsupported on GO) –2,025 (instances $n > 50$ size on FI) = 14,625 computational experiments. All these experiments needed 830 single blade equivalent CPU hours. No CPU time limit was imposed to any algorithm. The algorithms were implemented and ran following the instructions of their respective authors.

4 Analysis of results

All results are supported by statistical analyses. We mainly use the multifactor Analysis of Variance (ANOVA) technique where we control all studied factors. Three different groups of response variables are considered: CPU times needed by the algorithms, quantitative and qualitative comparison of symmetric (*TSP*) and asymmetric (*ATSP*) tours. All results are detailed in the following sections. Since the ANOVA is a parametric technique, one needs to check the three main hypotheses which are normality, homoscedasticity and independence of the residuals. The residuals resulting from the experiment were analyzed and no serious deviations were observed.

4.1 CPU times

Some of the most interesting results are observed when analyzing the CPU times needed by the algorithms capable of solving *ATSP* problems. The resulting ANOVA table is given in Table 2.

Source	Sum of squares	Degrees of freedom	Mean square	F-Ratio	p-Value
Main Effects					
A:Territory	2900.6	2	1450.3	17.7	0.0000
B:Location	1028.8	2	514.4	6.3	0.0019
C:Symmetry	2502.2	3	834.1	10.2	0.0000
D:n	753030	8	94128.8	1148.7	0.0000
E:Algorithm	1.546E6	6	257723	3145.2	0.0000
Interactions					
AB	351.6	4	87.9	1.1	0.3681
AC	1221.9	6	203.7	2.5	0.0210
AD	2664.9	16	166.6	2.1	0.0086
AE	11786.4	12	982.2	12	0.0000
BC	236.5	6	39.4	0.5	0.8229
BD	1788.7	16	111.8	1.4	0.1491
BE	4835	12	402.9	4.9	0.0000
CD	3348.8	24	139.5	1.7	0.0174
DE	1.038E6	48	21638.7	264.1	0.0000
Residual	849253	10364	81.9		
Total (corrected)	4.269E6	10529			

Table 2: Analisis of Variance (ANOVA) for CPU time response variable and *ATSP* algorithms (M≠T, A≠CO and A≠LK).

At a 95% confidence level ($\alpha = 0.05$), all single factors and 6 double factor interactions are statistically significant. Among significant factors, importance is observed by the magnitude of the F -Ratio. For example, the F -Ratio of the factor “Algorithm” is no less than 3,145.2. This means that the differences among the different algorithms generate 3,145.2 more variance than the variance obtained within each algorithm. Therefore, the type of algorithm has a very strong and statistically significant influence over the CPU time.

The ANOVA technique mainly points out statistical significance. For a further understanding of the behavior of any studied factor, we need descriptive plots. We have included plots with points and smoothed lines for comparing the CPU time as a function of the size of the matrices for all combinations of Symmetry (types of matrices) and Algorithms factors. All these plots are shown in Figure 3 where the X-axis gives the size of the matrix (n) and the Y-axis the CPU time in seconds, with a logarithmic scale. Each row in the plot corresponds to a type of matrix and each column to an algorithm. Note that there are no plots for algorithms LK and CO for asymmetric matrices, and GO for transformed matrices ($M=T$).

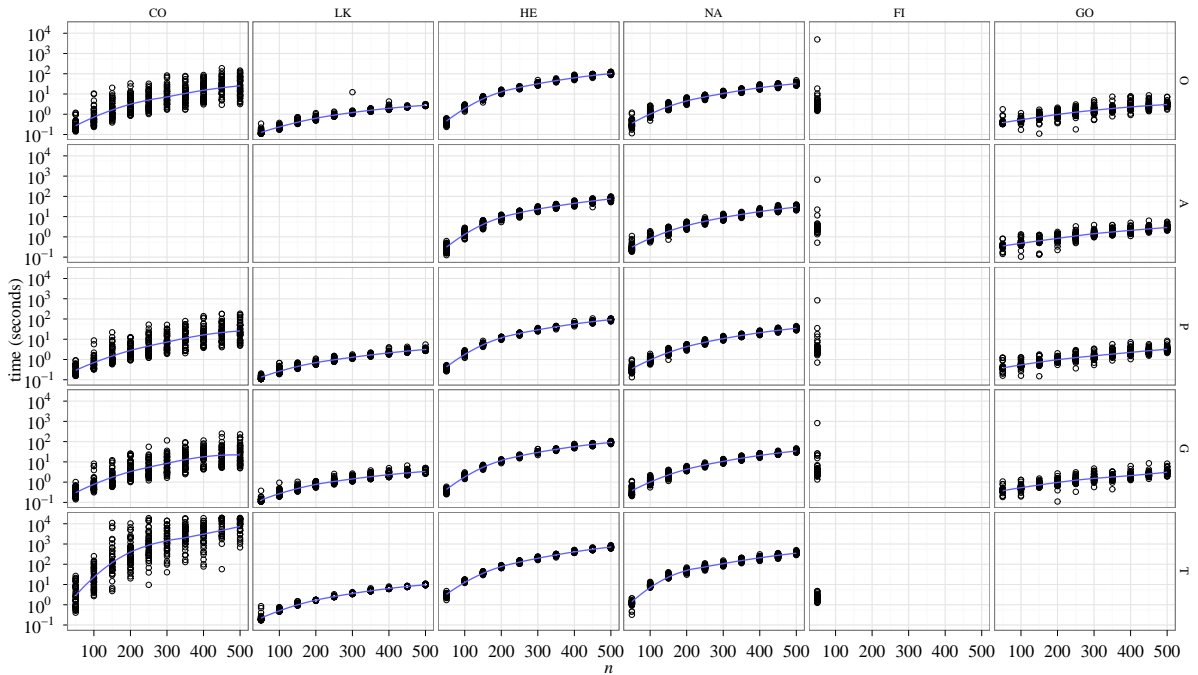


Figure 3: CPU time behavior for algorithms (CO, LK, HE, NA, FI, GO), types and sizes of matrices.

We confirm that the type of algorithm has a strong influence over CPU time. As expected, the improved Lin-Kernighan method of Helsgaun (HE) and Concorde (CO) are the most computationally demanding algorithms. The size of the matrix also affects CPU time directly and exponentially and this is the case for all types of matrices, symmetric and asymmetric and for all algorithms. Also expected is the matrix transformation process of matrices ($M=T$), which results

in enormous CPU time increases. This is a logical result that validates the whole experiment, as the size of the original asymmetric matrix is multiplied by two in the transformation process. However, and as shown in Figure 3 with symmetric matrices, HE is actually slower than CO (about three times slower). This is an unexpected result as CO is an exact procedure and HE, albeit extremely effective, cannot guarantee optimality. The matrix transformation ($M=T$) affects much more CO than HE as the CPU time increases approximately by a factor of 7. The problem is that CO only works with symmetric matrices and the transformation is the only possible way of dealing with asymmetric problems. The corollary is that CO is far more sensible to the size of the *TSP* to solve.

As regards the other studied factors, the location affects the CPU time for all methods. The means plot of Figure 4 contains the interaction between Algorithm and Location factors for asymmetric matrices only. The means are plotted in the middle of Tukey’s Honest Significant Difference (HSD) 95% confidence intervals. Overlapping intervals denote that the means contained within them are not statistically different. Grid locations result in slightly higher CPU times for all methods compared to the Random and Radial locations; except for the FI algorithm as it consumes more CPU time for Random location ($n = 50$). In the case of symmetric matrices (not shown in the figure), CPU times increase sharply (20%) for the CO method for Grid locations and this difference is statistically significant. To the best of our knowledge, there are no reported studies that analyze how the distribution of the nodes or clients and road transportation networks affect the CPU times of state-of-the-art methods.

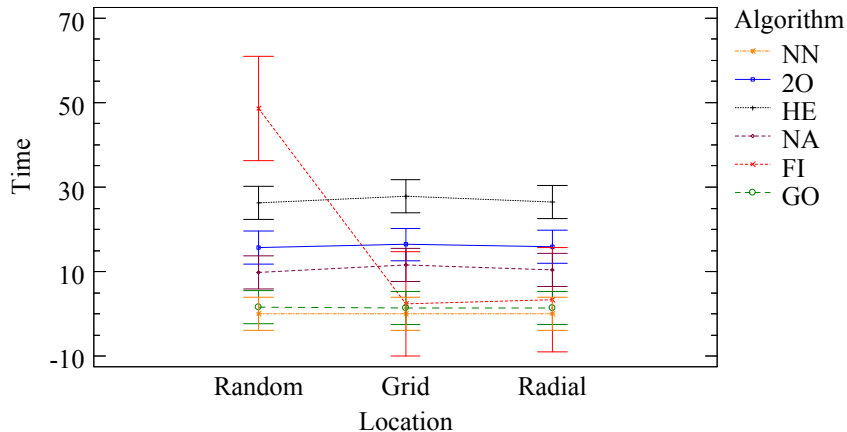


Figure 4: Means plot with Tukey’s Honest Significant Difference (HSD) 95% confidence intervals for the interaction between Location and Algorithm factors, where the response variable is CPU time ($M=A$).

Other interesting findings affect the Territory factor. Figure 5 shows the interaction between Algorithms and the Territory factor for asymmetric matrices only. In general, and especially for FI, there is a preference of algorithms for short distances territories. A possible explanation is that in short distances there is more variability in the distances between nodes and possibly this helps in the process of finding a solution. In case of symmetric matrices (not shown in the figure), CPU

time increases up to 37% for CO in cases of long and medium distances compared with the short distances territories. A contribution of this work is to demonstrate that the different degree of asymmetry (and differences between distances) of the road transportation network affect studied methods accordingly.

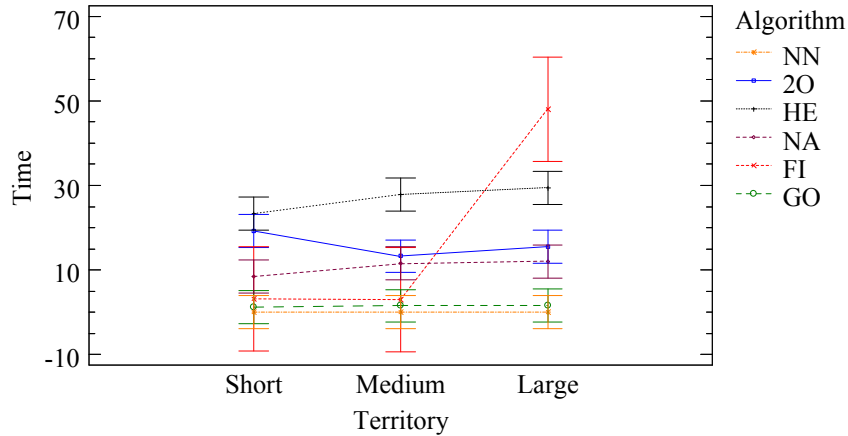


Figure 5: Means plot with Tukey’s Honest Significant Difference (HSD) 95% confidence intervals for the interaction between Territory and Algorithm factors, where the response variable is CPU time (M=A).

Finally, CPU time is affected, on average, by the Symmetry factor (type of distance matrix). If we remove the transformed matrix (M=T) which we have already seen increases CPU times by orders of magnitude, the result obtained is shown at Figure 6. We see that there are no statistically significant differences between the M=P and M=G matrices. Recall that these represent symmetric matrices where the distances are the minimum and maximum distances, respectively, between the from-to and to-from asymmetric distances in the matrix. This means that the differences in CPU time cannot be attributed to the magnitude of the distances, but rather to the differences in the distances themselves. We also observe how asymmetric matrices (M=A) need significantly more CPU time than regular orthodromic matrices (M=O).

4.2 Quality of solutions

It has to be reminded that the objective at this step is not to measure which algorithm, among the tested ones, is the best. The focus is rather on studying how the considered factors affect the quality of the solutions provided by the algorithms. Table 3 provides the number of times that each algorithm provides the best solution ($N. S^*$), and the corresponding rate ($\% S^*$) under three different settings. The second and third columns indicate matrices M=(O, P, G) (1, 350 experiments per algorithm). The fourth, fifth, sixth and seventh columns indicate asymmetric (M=A) and transformed (M=T) cases, respectively, with 450 experiments per algorithm. Note that in the case of A=GO there is no data for M=T, and for A=FI there is only data for $n = 50$ problem size.

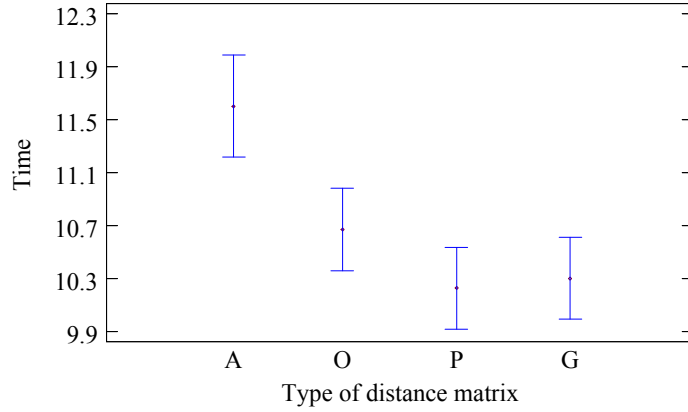


Figure 6: Means plot with Tukey's Honest Significant Difference (HSD) 95% confidence intervals for the Symmetry factor (type of matrix), where the response variable is CPU time ($M \neq T$, $A \neq CO$ and $A \neq LK$).

Algorithm	M=(O,P,G)		M=A		M=T	
	N. S^*	% S^*	N. S^*	% S^*	N. S^*	% S^*
NN	0	0.0%	0	0.0%	0	0.0%
2O	3	0.2%	0	0.0%	0	0.0%
LK	627	46.4%	-	-	70	15.6%
CO	1350	100%	-	-	391	86.9%
HE	930	68.9%	414	92.0%	33	7.3%
NA	463	34.3%	155	34.4%	30	6.7%
FI	83	61.5%	45	100%	0	0.0%
GO	252	18.7%	66	14.7%	-	-

Table 3: Number of best solutions and success rates for the studied algorithms and types of matrices.

As expected, CO always produces the optimum solution for the 1,350 symmetric instances. Expectedly, HE's success rate is high at almost a 69% and much better than LK's at 46%. The simple heuristics NN and 2O are rarely successful. For asymmetric matrices (M=A), FI is clearly dominant on small sized problems ($n = 50$). For other asymmetric cases, HE is clearly dominant and no further results can be drawn from our comparison since LK and CO do not accept asymmetric matrices. Most surprisingly, CO does not obtain the optimum solutions in all cases for M=T. An obvious explanation is that in the transformation process, some values in the matrix are $-\infty$ or $+\infty$ and this creates numerical instability problems inside CO that result in small deviations from the true optimum solution. These ∞ values were also the reason for not experimenting with M=T for the GO method. With these results, we can now conclude that not only CO needs an exponentially greater CPU time for transformed matrices, but also that the results cannot be trusted. Naturally, with modifications inside the CO code, there is the possibility that transformed matrices could be considered without glitches. Another interesting outcome is that for transformed matrices, NA and LK outperform HE. However, the transformation process is actually not needed for HE or NA and we cannot conclude that LK or NA are preferred over HE for transformed matrices.

Numerous statistical analyses were performed in order to check the influence of the studied factors over the quality of the solutions. Multiple ANOVA experiments were performed, which are not fully detailed due to space restrictions. It has to be noted that since no maximum CPU time was given to all tested algorithms, the effect of the different studied factors over solution quality is about 1% or less (contrary to the previous observed effects on CPU time). While this might be seen as a marginal effect, it has to be reminded that in the *TSP* state-of-the-art literature, publications and new results are often disputed with improvements of less than 2% in solution quality (Helsgaun, 2000). However, almost all factors resulted statistically significant in all tests carried out. Table 4 shows average ΔS_i^* values, defined by expression (2) in previous sections, for the different tested algorithms as a function of the type of matrix (symmetry factor). Again, we see the large deterioration in FI, HE, NA and CO with transformed matrices (M=T). For large asymmetric matrices, HE, NA and GO are good choices. Whereas for all other matrices, either CO or HE clearly dominate. Once again we see that for transformed matrices, it is even better to use LK than HE.

After studying the different matrices' asymmetry degree and all other studied factors, we found that the asymmetry degree of a matrix (δ) and the Territory factor are strongly related. In our experiments we have observed and demonstrated that the asymmetry of the arcs on short distances territories is much higher than medium or large distances territories, which is relevant and interesting for the case of city logistics problems. It is logical to think that these differences in the degree of asymmetry affect the behavior of algorithms, as shown at Table 5.

It is relevant and interesting to observe how depending on the type of algorithm, there are preferences for more symmetrical or asymmetrical environments, or what is equivalent: long

	O	A	T	P	G	Average
NN	23.78	23.63	22.88	22.12	21.10	22.70
2O	8.34	17.11	631.33	6.49	5.93	133.84
LK	0.04	–	1.56	0.05	0.05	0.42
CO	0.00	–	1.19	0.00	0.00	0.30
HE	0.02	0.01	9.10	0.02	0.02	1.83
NA	0.04	0.01	7.07	0.04	0.04	1.45
FI	0.18	0.00	25.84	0.07	0.03	5.23
GO	0.45	0.26	–	0.38	0.36	0.36
Average	4.11	6.84	99.85	3.64	3.44	

Table 4: Average ΔS_i^* values according to Algorithm and Symmetry (type of matrix) factors.

	Short	Medium	Large	Average
NN	24.28	21.88	21.95	22.70
2O	119.02	136.98	145.51	133.84
LK	0.42	0.41	0.44	0.42
CO	0.41	0.30	0.18	0.30
HE	3.40	1.09	1.00	1.83
NA	2.92	0.76	0.65	1.45
FI	4.79	5.50	5.39	5.23
GO	0.31	0.38	0.40	0.36
Average	19.44	20.91	21.93	

Table 5: Average ΔS_i^* values according to Algorithm and Territory factors.

distances (CO, HE, NA) and medium distances (LK) versus short distances (FI, GO, NN, 2O). Furthermore, an although not shown, this effect is observed for all matrix sizes and specially for transformed matrices. We now analyze the behavior of the different algorithms against the Location factor in Table 6. As shown, no overly strong effects are observed (albeit all differences are statistically significant for Location factor). It is interesting to note that although the road network in the Iberian Peninsula is characterized by a radial structure centered in the capital Madrid, the degree of asymmetry increases slightly in such locations. Other road networks in other countries could be an important relationship between location and asymmetry of the arcs that conditions more the behavior of algorithms.

	Random	Grid	Radial	Average
NN	23.49	22.52	22.10	22.70
2O	150.47	124.90	126.14	133.84
LK	0.45	0.31	0.50	0.42
CO	0.25	0.53	0.12	0.30
HE	1.98	1.52	1.99	1.83
NA	1.54	1.06	1.73	1.45
FI	5.56	6.73	3.38	5.23
GO	0.44	0.27	0.39	0.36
Average	23.02	19.73	19.54	

Table 6: Average ΔS_i^* values according to Algorithm and Location factors.

Lastly, it is worth mentioning that matrix size has a very small impact on ΔS_i^* values. The effect is less than 0.06% in the worst case. Figure 7 shows the averages of the Symmetry and Algorithms factors (excluding M=T). The horizontal axis shows the size of the matrix n and the vertical axis (different scale for each symmetry factor) shows the percentage deviation ΔS_i^* over the best solution. Note the vertical scale and values for M=T matrices, where the effect of this type of matrix and size n on ΔS_i^* is higher.

4.3 Quantitative and qualitative assessment

After studying the different matrices' asymmetry degree and all other studied factors, we found that the asymmetry degree of a matrix (δ) and the average weight (\overline{Weight}) of the different distances in the matrix are strongly related. If there is a relation between symmetric and asymmetric matrices in the form of an increased average weight, it is logical to think that the symmetric solution of the *TSP* could be "augmented" in order to carefully estimate the real *ATSP* solution (as regards the total tour length). This is needed since, as we have already stated, the *TSP* tour length is a loose lower bound of the real *ATSP* tour length. Similarly, it is important to check the tour length of the *TSP* solution, when calculated with the *ATSP* matrix and viceversa. In order to check all these questions we use the following indicators, which are strongly based on the previously defined response variables.

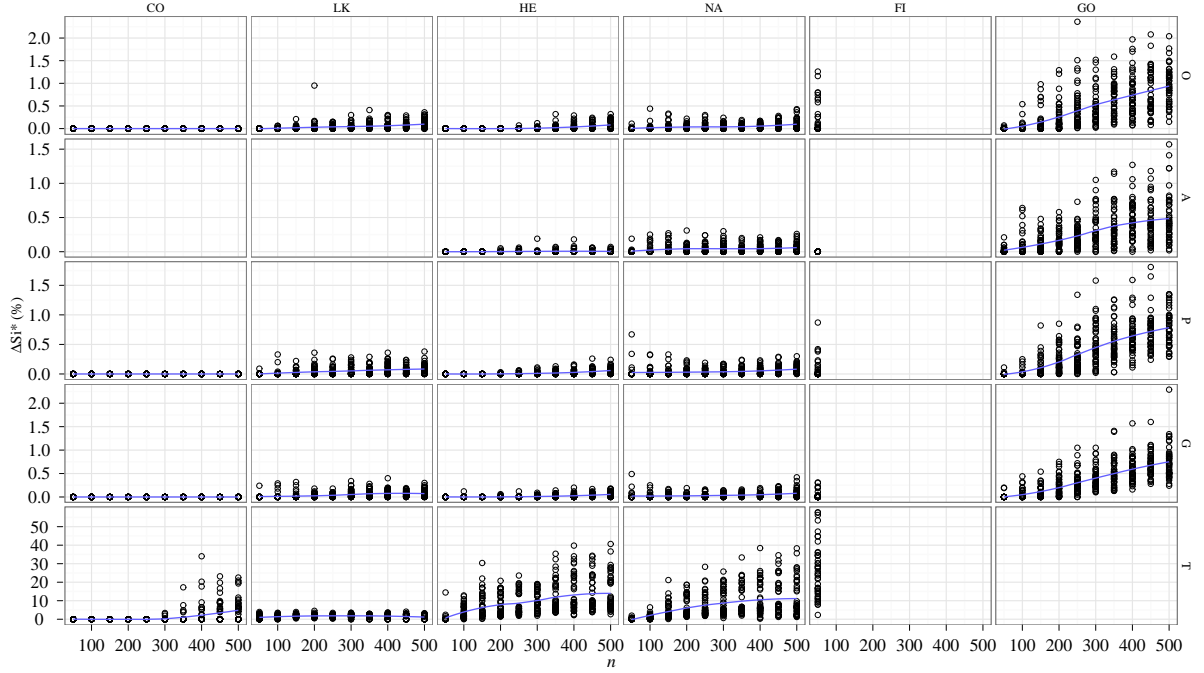


Figure 7: Average ΔS_i^* values according to the size of the matrix n , algorithms (CO, LK, HE, NA, FI, GO) and Symmetry factors.

$\Delta ATSP$: It is the percentage increase of the tour length of the $ATSP$ solution as regards the TSP solution:

$$\Delta ATSP = \frac{ATSP - TSP}{TSP} \cdot 100 \quad (5)$$

TSP is the tour length of the symmetric problem, calculated with symmetric orthodromic matrices. $ATSP$ is the asymmetric problem tour length, calculated with real distance matrices ($M=A$). Note that algorithms LK and CO cannot solve the $ATSP$. In these cases the transformed matrix ($M=T$) is used instead.

TSP_A : The TSP solution is calculated with asymmetric matrices. i.e., we take the solution of a TSP problem and recalculate it with the real distances. Obviously, the tour length will increase ($TSP_A \geq TSP$).

$ATSP_O$: It is the opposite case as TSP_A . The $ATSP$ solution is calculated with the symmetric matrix.

ΔTSP_A : It is the percentage increase of TSP_A against $ATSP$. It could be positive or negative.

$\Delta ATSP_O$: It is the percentage increase or decrease of $ATSP_O$ against TSP .

ΔdH : It is the percentage of differences in the TSP solution against $ATSP$. Values close to 100% qualitatively indicate that the TSP solution is very different from the $ATSP$. It is

based on the previously defined Hamming distance dH :

$$\Delta dH = \frac{dH}{n} \cdot 100 \quad (6)$$

ΔdA : It is the percentage of different arcs, over the total arcs, that the TSP sequence has over the $ATSP$ solution. It is based on the previous adjacency distance dA :

$$\Delta dA = \frac{dA}{n+1} \cdot 100 \quad (7)$$

We calculate all previous indicators for all experiments, namely 3 territories \times 3 locations \times 10 different matrix sizes \times 8 different algorithms \times 5 replicates which results in 3,195 data. The results are good. The average $\Delta ATSP$ indicator reaches a value of 80.1%. This indicates a huge difference between the $ATSP$ and TSP solutions. Note that the minimum observed value for this indicator is an already large 32.9% (the maximum being an impressive 196.9%). The frequency distributions of the $\Delta ATSP$ values are given as an histogram in Figure 8 (left). It is observed that in a large percentage of the cases, the increase is between 50% and 100%. Figure 8 (right) shows a second histogram, this time for ΔTSP_A . The distribution is clearly skewed towards positive values, with an average of 13.6%. Exactly, 39.8% of the cases show differences equal or larger than 10%.

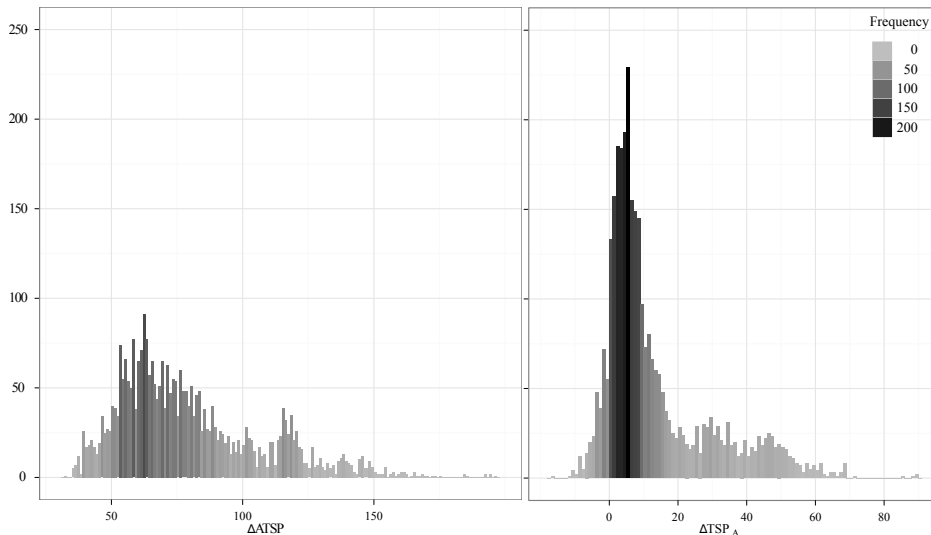


Figure 8: Histograms with the $\Delta ATSP$ (left) and ΔTSP_A (right) frequency distributions.

A very strong result, especially with the second histogram, is that there is a large difference between solving symmetric and asymmetric problems. The idea that orthodromic or Euclidean distances for solving the TSP are valid (seen in some commercial routing software) and that after all, one can later calculate real distances with the solution is utterly wrong. As shown, on average, there are differences that amount to a 13.6%. Note that these differences are extremely

large for *TSP* problems where algorithms fight in the state-of-the-art segment for less than 2% improvements. As regards $\Delta ATSP_O$, the distribution (not shown) is again positive, with an average of 14.3%. In a 50.02% of the cases, the differences are equal or greater than 10%. Once again, it is shown that it is not the same solving an asymmetric *ATSP* and measuring the tour length of the sequence with symmetric distances.

In our opinion, this is the main contribution of this research work. From our perspective, the commonly accepted assumption of considering Euclidean distances does not hold when solutions are calculated with real asymmetric distances. The *TSP* solutions deteriorate enormously when calculated with real matrices. Furthermore, there is little guarantee that good algorithms for the *TSP* will work equally good for the *ATSP*. Often, the degradation in performance will be significantly greater than the observed differences between competing methods. Another significant result is to qualitatively observe the big differences between the sequences obtained with symmetric *TSP* and asymmetric *ATSP* problems. The average ΔdA , with a value of 71.0% indicates that the symmetric sequences are almost entirely different from the asymmetric ones. Values are even greater if one uses the Hamming distance ΔdH . The minimum value for the ΔdA value is as high as 6%. This means that, in the best case, a full 6% of the sequence is different. Figure 9 (left) shows the frequency distribution of ΔdH . Note how it is heavily skewed towards values very close to 100%. Figure 9 (right) shows the corresponding histogram for ΔdA .

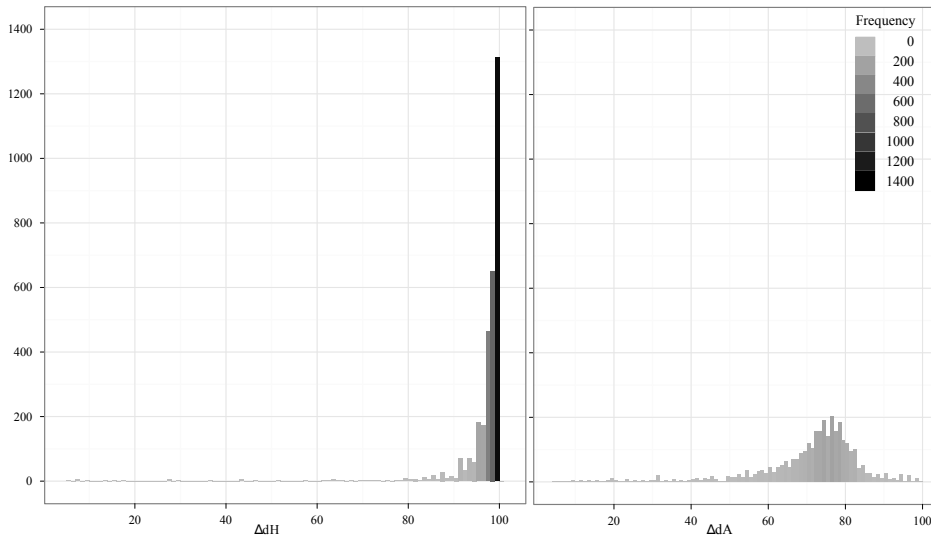


Figure 9: Histograms with the ΔdH (left) and ΔdA (right) frequency distributions.

4.4 Some examples

In this section we depict two examples. The purpose is to graphically show the large differences between symmetric and asymmetric solutions. Only two examples have been randomly selected due to obvious space constraints.

CC-0503HE. The first example corresponds to instance CC-0503. A total of 50 nodes are grid located in a short distance territory. We compare the best solution obtained with algorithm HE. There are huge differences between the symmetric optimal tour length of 244.2 km. versus the 423.3 km. of the different optimal tour length calculated with the asymmetric distances. Therefore, $\Delta ATSP = 73.32\%$. Additionally, we have that $\Delta dA = 90.20\%$. Figure 10 (left) has the two optimum solutions superimposed. The symmetric in blue and the real one in red.

MR-0504CO. In this second example we show the results of the instance MR-0504 solved with CO algorithm. We have again 50 nodes radially distributed in a medium distance territory. The total tour length in the symmetric tour is 1,369.7 km. versus 2,097.9 km. for the asymmetric tour length. This results in a $\Delta ATSP$ of 53.17%, with a ΔdA of 98.04%. Figure 10 (right) shows the details.

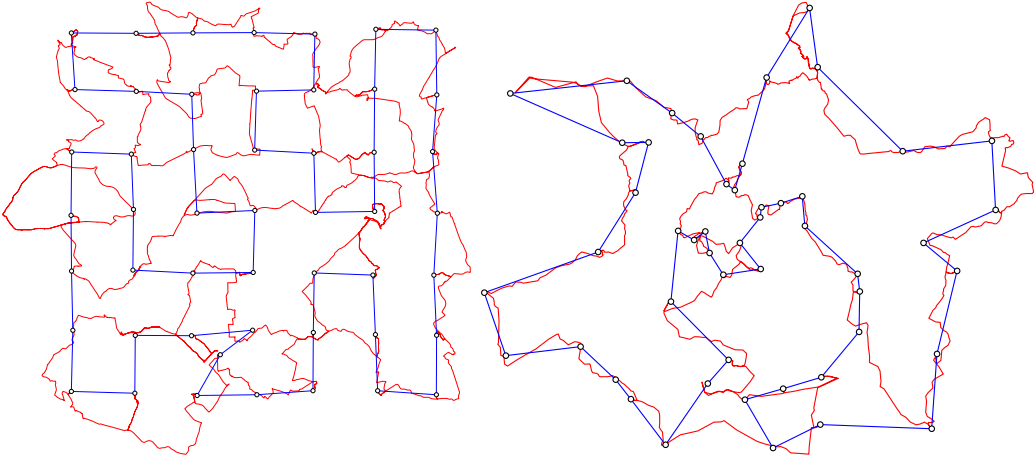


Figure 10: *TSP* (blue) and *ATSP* (red) solutions for A=HE on instance CC-0503 (left) and A=CO on instance MR-0504 (right) .

5 Conclusions and further work

In this work we have studied the effect that the asymmetry has over the solution process of the *TSP/ATSP*. We have shown that solving the *TSP* differs much from solving the *ATSP* at so many levels: tour length, adjacency of nodes, hamming distance and CPU time. Furthermore, all these differences are strongly affected by other and new studied factors: degree of symmetry, territory and location of the nodes in the road transportation network. During the research, thousands of instances have been solved with some of the best well known (including some state-of-the-art) algorithms for the *TSP* and *ATSP*.

We have been able to confirm, as expected, that the algorithm used strongly affects the CPU time. HE and CO are the most computationally demanding methods, much more than the other simplistic heuristics or modern meta-heuristics. We have also shown that the Territory, Location

and size of the problem factors all affect the different methods in a sound and statistically significant way. As is known, the problem size increases the CPU time exponentially. An interesting and novel contribution of this paper is the study of the effect on CPU time and quality of solutions due to the factors of territory, location and their relationship with the asymmetry degree of the road transportation network. Particularly for symmetric matrices, locations in grid have a significant effect on CPU times, resulting in harder to solve problems. This effect is amplified with transformed matrices. It has been demonstrated that the Territory has an impact on CPU time, especially for short distance territories. For symmetric transformed matrices, the Territory factor is relevant as regards the quality of the solution. This effect is observed for all matrix sizes. Another conclusion of this study is that the transformation process has a profound effect over CPU times. Furthermore, this transformation process has shown not to be entirely feasible for algorithms HE and, especially, for CO. If one closely compares HE and CO for symmetric matrices, we observe that HE is about three times slower than CO. This is unexpected, since CO is an exact branch-and-cut method and HE a (powerful) heuristic. However, the transformation process increases the computation time more than sevenfold, and affects CO much more. The result is that asymmetry has a deep impact over algorithms, either from the quality of the solutions standpoint, or from the CPU time whenever matrix transformation is needed.

Comprehensive statistical experiments further demonstrate that there is an inverse relation between the Territory factor and the average $\Delta ATSP$ indicator value. This confirms that there are quantitative differences between the *TSP* and *ATSP* solutions. The differences between the two solutions are smaller for large distances than for short distances. Furthermore, we found relations between the Location factor and the $\Delta ATSP$ indicator. In this case, the existing differences are greater as the size of the problems grows and they are greater for radially and randomly placed locations than for grid ones. The size of the matrix also conditions the differences between the *TSP* and *ATSP* solutions. In any case, solving a *TSP* and later calculating the real tour length with real distances is not a viable solution process.

The asymmetry is not just a binary condition of the problem. There are different degrees of asymmetry as the territory and the location of the nodes in the road transportation network. This results in different degrees of complexity, and the effect on CPU time and solution quality. Some algorithms (HE, FI, NA, GO) are better equipped to solve problems with a high degree of asymmetry or short distances, and in certain location patterns. Others (LK, CO, HE, NA) provide better performance in symmetrical environments or long distances. This paper hopes to inspire future research on the development and testing of new and improved algorithms, not only taking into account the asymmetry condition, but also new factors studied here. New instances have been made available to the scientific community. Further work stems from the possibility of providing effective methods for calculating real asymmetric travel matrices, as this imposes today a clear entry barrier for those researchers not willing to use the typical Euclidean matrices. Extending this study to more complex problems, like for the Capacitated Vehicle Routing

Problem *CVRP* or for the Heterogeneous Fleet variant (*HFCVRP*) is of interest to see if more realistic routing problems are equally affected by asymmetry.

Acknowledgements

The authors would like to warmly thanks to: Prof. Keld Helsgaun, Prof. Yuichi Nagata, Profs. Boris Goldengorin and Gerold Jäger, and especially Prof. Matteo Fischetti, for facilitating the code of their great algorithms. This work is partially funded by the Spanish Ministry of Science and Innovation, under the project “SMPA - Advanced Parallel Multiobjective Sequencing: Practical and Theoretical Advances” with reference DPI2008-03511/DPI. The authors should also thank the IMPIVA - Institute for the Small and Medium Valencian Enterprise, for the project OSC with references IMIDIC/2008/137, IMIDIC/2009/198 and IMIDIC/2010/175 and the Polytechnic University of Valencia, for the project PPAR with reference 3147.

References

- Alba, E. (2006). *Parallel metaheuristics. A New Class of Algorithms*. Wiley, New York.
- Applegate, D. L., Bixby, R. E., Chvátal, V., and Cook, W. J. (2006). *The Traveling Salesman Problem: A computational study*. Princeton University Press, New Jersey.
- Applegate, D. L., Cook, W. J., Dash, S., and Rohe, A. (2002). Solution of a min-max vehicle routing problem. *INFORMS Journal on computing*, 14(2):132–143.
- Arora, S. (1998). Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782.
- Asadpour, A., Goemans, M. X., Madry, A., Gharan, S. O., and Saberi, A. (2010). An $O(\log n/\log \log n)$ -approximation Algorithm for the Asymmetric Traveling Salesman Problem. In *Proc. of 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, volume 135 of *Proceedings in Applied Mathematics*, pages 379–389, Philadelphia. ACM; SIAM Activ Grp Discrete Math; SIGACT, SIAM.
- Bontoux, B., Artigues, C., and Feillet, D. (2010). A Memetic Algorithm with a large neighborhood crossover operator for the Generalized Traveling Salesman Problem. *Computers & Operations Research*, 37(11):1844 – 1852. Metaheuristics for Logistics and Vehicle Routing.
- Christofides, N. and Eilon, S. (1969). Expected distances in distribution problems. *Operations Research*, 20(4):437–443.
- Clarke, G. and Wright, J. (1964). Scheduling of vehicles from central depot to number of delivery points. *Operations Research*, 12(4):568–581.
- Croes, G. (1958). A method for solving Traveling-Salesman Problems. *Operations Research*, 6(6):791–812.
- Daganzo, C. (1984). The length of tours in zones of different shapes. *Transportation Research Part B*, 18(2):135–145.
- Dubois, N. and Semet, F. (1995). Estimation and determination of shortest-path length in a road network with obstacles. *European Journal of Operational Research*, 83(1):105–116.
- Fischetti, M., Lodi, A., and Toth, P. (2003). Solving real-world ATSP instances by branch-and-cut. In Junger, M. and Reinelt, G. and Rinaldi, G., editor, *Combinatorial Optimization - Eureka - Papers dedicated to Jack Edmonds*, volume 2570 of *Lecture notes in Computer Science*, pages 64–77.

- Flood, M. (1956). The TSP. *Operations Research*, 4(1):61–75.
- Gerns, R., Goldengorin, B., and Turkensteen, M. (2012). Lower tolerance-based Branch and Bound algorithms for the ATSP. *Computers & Operations Research*, 39(2):291–298.
- Goldengorin, B., Jaeger, G., and Molitor, P. (2006). Tolerance based Contract-or-Patch heuristic for the asymmetric TSP. In Erlebach, T, editor, *Combinatorial and Algorithmic aspects of Networking*, volume 4235 of *Lecture notes in Computer Science*, pages 86–97, Chester, England.
- Goodchild, M. and Kemp, K. (1990). *NCGIA Core Curriculum in GIS*. National Center for Geographic Information and Analysis, University of California., Santa Barbara CA.
- Gouveia, L. and Pires, J. (1999). The asymmetric travelling salesman problem and a reformulation of the Miller-Tucker-Zemlin constraints. *European Journal of Operational Research*, 112(1):134–146.
- Gutin, G. and Punnen, A. (2002). *The Traveling Salesman Problem and Its Variations*, volume 12. Kluwer, Dordrecht, Dordrecht, Boston, London.
- Hamming, R. (1950). Error detecting and error correcting codes. *Bell System Technology Journal*, 29(2):147–160.
- Helsgaun, K. (2000). An effective implementation of the Lin-Kernighan Traveling Salesman Heuristic. *European Journal of Operational Research*, 126(1):106–130.
- Jonker, R. and Volgenant, T. (1983). Transforming asymmetric into symmetric traveling salesman problems. *Operations Research Letters*, 2(4):161–163.
- Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., and Shmoys, D. B. (1985). *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*. John Wiley & Sons Ltd., New York.
- Lin, S. and Kernighan, B. (1973). Effective heuristic algorithm for traveling-salesman problem. *Operations Research*, 21(2):498–516.
- Love, R. and Morris, J. (1972). Modeling inter-city road distances by mathematical functions. *Operational Research Quarterly*, 23(1):61–71.
- Love, R. and Morris, J. (1988). On estimating road distances by mathematical functions. *European Journal of Operational Research*, 36(2):251–253.
- Montgomery, D. C. (2009). *Design and Analysis of Experiments*. Wiley, New York, seventh edition.
- Nagata, Y. (2006). New EAX Crossover for Large TSP Instances. In Runarsson, T. P., Beyer, H.-G., Burke, E. K., Guervós, J. J. M., Whitley, L. D., and Yao, X., editors, *PPSN*, volume 4193 of *Lecture Notes in Computer Science*, pages 372–381. Springer.
- Nagata, Y. and Kobayashi, S. (1997). Edge assembly crossover: A high-power genetic algorithm for the travelling salesman problem. In *ICGA*, pages 450–457.
- Schiavinotto, T. and Stützle, T. (2007). A review of metrics on permutations for search landscape analysis. *Computers & Operations Research*, 34(10):3143–3153.
- Toth, P., Vigo, D., for Industrial, S., and Mathematics, A. (2001). *An overview of vehicle routing problems*. SIAM - Society for Industrial and Applied Mathematics, Philadelphia.