PART I – INFORMATION SHEET

Originating BD/BU/Product Line	Development Project Reference	Date
CTO - R & I	IMS Converging Applications	29/08/2006

TITLE (10 words maximum)

Process for calculating some solutions to the Traveling Salesman Problem

Main INVENTOR

Employee	Full Name	Employee No. / UPI	Phone	Subsidiary Unit
☒ Yes ☐ No	LARVET Philippe	AG0005694	2111 1120	CTO - R&I
Position	**Citizenship**	**E-mail Address**	**Supervisor Name, Phone No.**	
Engineer	French	Philippe.Larvet@alcatel.fr	Guillaume DORBES, 2111 1351	
Home Address		**City, State, Zip Code**		**Country**
13 rue du Jeu de Paume		91470 Forges les Bains		France

FRAME of the STUDY

COMPANY FUNDING:

☒ Corporate funding, or

☐ Direct funding by Business Unit - Please identify the Business Unit:

What is the value of the invention with / without standardization?

No standardization.

Business Division(s) WITH POTENTIAL INTEREST (Include contact name, if available):

ESD, MSD, FSD

PART II - TECHNICAL DESCRIPTION

What is the technical problem that the author of the FIT (= the inventor) had to solve?

The Traveling Salesman Problem (TSP for short) is an old mathematical problem. Given a collection of cities (or points) and the cost of travel between each pair of them, the TSP is to find the cheapest (the shortest) way of visiting all of the cities and returning to the starting point.

What is the best existing solution of this problem, to the knowledge of the author AND why this solution is not good enough? (drawings illustrating this existing solution are welcome)

Numerous solutions exist to this problem, since 1954 (solution for 49 cities) to 2004 (solution for 24978 cities) – see for example http://www.tsp.gatech.edu/history/milestone.html - but the computation of these solutions needs multi-processor devices : for example, a 110 parallel-processor system was necessary for solving the solution for 15112 cities in Germany, in 2001, and the total computer time used in the computation, scaled to a Compaq EV6 Alpha processor running at 500 MHz, was 22.6 years.
Then, some more simple solutions are needed, using simple standard mono-processor computers, and demanding a reasonable computation time.

Basic idea of the author's solution

The basic idea is to determine, for each point M of the set of n points, the two "metaproximate" points M1 and M2, i.e. the points that are close to M ***and*** located on the shortest way that connects all the points. The notion of "metaproximate point" is a concept we want to introduce in the scope of this invention. M1 and M2 are "close" to M, but are not necessary, of course, the ***closest*** points to M.
For example, in the Fig.1 below, B and C are the closest points to A, but the "metaproximates" of A are B and G, because from the point A, the two ways located on the global shortest way (in blue) are AB and AG.

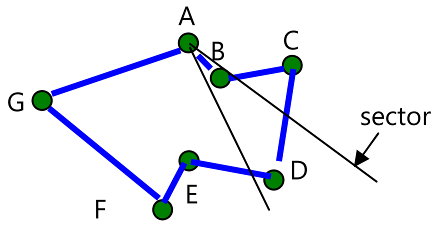


Fig.1 Illustrating "metaproximate" and "horizon" points

The determination of the "metaproximates" for a given point M is made by examinating the "horizon points" for the point M. The notion of "horizon point" is another concept we want to propose in the scope of this invention. An "horizon point" for a given point M is the closest point to M situated in a given space sector around M.
This notion comes from our observations of numerous solutions to the TSP, mainly observations concerning the modification of the track of the shortest way when occur some modifications of the positions of some points. If the number of points is large enough (more than 20, for example), an "horizon effect" appears: this means that modifying some positions of points at one end of the set does not imply a modification of the shortest way at the other end of the set, as schematized in Fig.2. below.

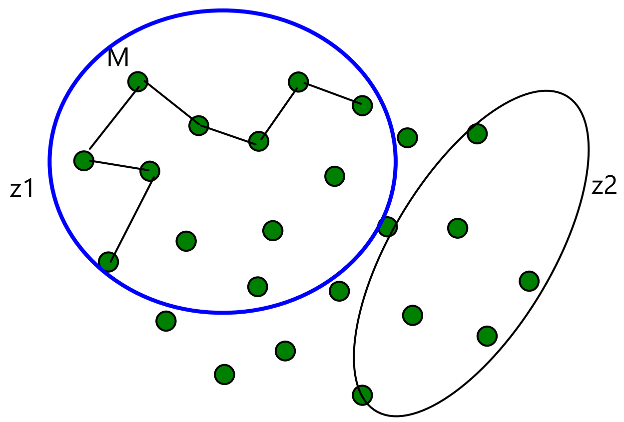


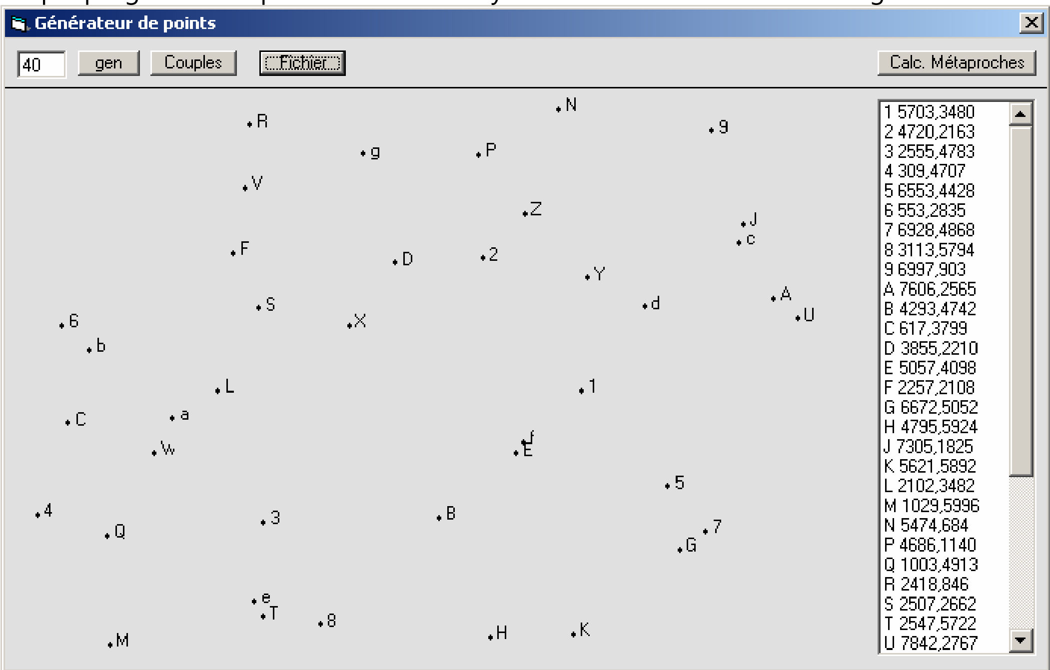
Fig.2 The "horizon effect": modifying positions of some points in area z2
has no effect on the shortest way determined around point M in area z1

This "horizon effect" allows us to determine the "horizons" for a given point.
For example, we want to determine the "horizons" for the point A in Fig.1 above. We imagine a radar situated on A, turning on 360° by discrete steps, splitting the space into a given number of virtual sectors. For our own samples, we have chosen a splitting in 16 sectors of 22.5° each, but a splitting in 32 sectors of 11.25° each is also possible, depending on the density of the set of points to process. For a given sector, we decide to keep only the closest point to A situated in the considered sector. For example, in the sector shown in Fig.1, two points are "visible" from A: B and D – but we decide to keep only the closest, B.
So, after a complete 360° turn around A, this process allows to determine all the "horizons" of A : C, B, E and G.
We propose to express formally the "horizons" for the considered point under the form of a vector, that we consider as a "sectorial signature" of the considered point, and that contains, for each sector, the identifier of the point which is kept for this sector. Of course, some sectors can be empty; an empty sector is marked by a dot "." in the "sectorial signature". The virtual sectors are clock-wise numbered from 1 to 16, the #1 being located to the North.

For example, the "sectorial signature" for A is: `"....C.BE..G....."`
From the "sectorial signatures", containing the "horizons" for all the points, the determination of the "metaproximates" of each point is based on a process of "swapping by insertion", which is explained below.

Description of at least one embodiment of the solution accompanied by appropriate drawings

The solution comprises 3 steps: determination of Horizons, determination of Metaproximates and final connection, determining the shortest way. Let us describe the solution through a 40-points sample, randomly generated by a simple program. Each point is described by its coordinates X and Y - see Fig.3 here :



Before determining the Horizons, all the (40*39)/2=780 distances between the points are calculated.
"1f=7687" means the distance between point "1" and point "f" is equal to 7687.
Here are some of the distances for our 40-points sample :

```
1f 7687
1E 8940
1d 10398
1Y 11204
15 12733
12 16434
1G 18467
1Z 18485
17 18513
1B 18923
1A 21115
1c 21435
1D 22423
1U 22547
1J 23034
1X 23954
1K 24134
1P 25514
1H 26072
1N 28054
19 28836
1g 31940
1S 32990
13 34070
18 34731
1L 36010
1F 37091
1e 38557
1T 38713
1V 38942
```

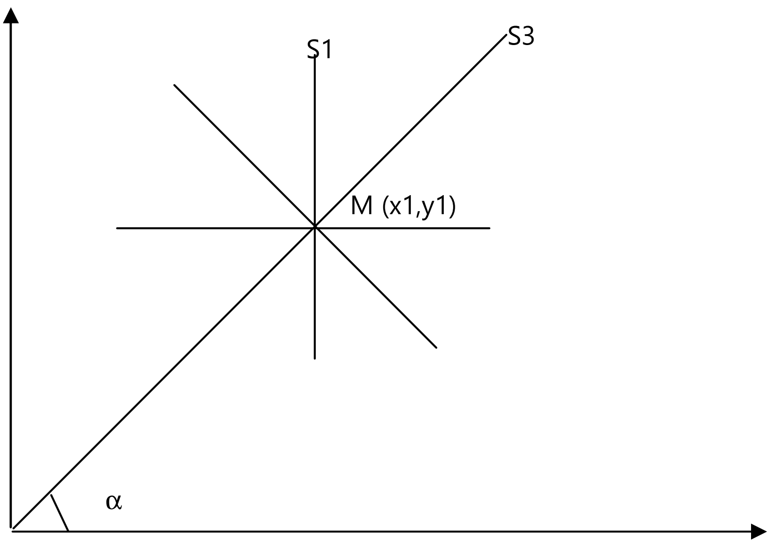


Fig.4 Calculating the sectors for the M point

```
1a 40620
1R 42106
1W 42769
1b 48910
1Q 49136
1C 50960
...
```

1) Determining the Horizons

This computation is made for each point, regarding the (n-1) other points. The space around the point is divided into 16 sectors, delimited by straight lines whose the equation has the general form : $y-y1 = tg\alpha(x-x1)$, where x1 and y1 are the coordinates of M, as shown in Fig.4.

The "sectorial signatures" of all points are then obtained, and the processing of their structure (i.e. the positions of kept points and the positions of empty sectors) allow to determine the order in which the "swapping-insertion" will be processed to determine the Metaproximates.

Here are some of these signatures for our sample, after the structure-processing :

```
1 : Yd5f2
2 : PZY1D
3 : B8eQWL
4 : CWQM
5 : AU7Gf1
6 : VFSBC
...
F : VSL6
G : 7K5
H : K8B
...
```

2) Determining the Metaproximates

For each reference point, the horizon signature is now processed, by using an algorithm of "swapping-insertion":

- all the points of the signature are swapped, using optimized permutations. For example, if the points of the signature are 1, 2 and 3, the number of useful optimized permutations is 3, because in the scope of this very problem, the way "123" is the same than "321". In general, for h horizons, the number of useful permutations is (h!)/2.
- the reference point is "inserted" inside the permutations; for example, if horizons of point A are 1,2 and 3, the following permutations are used: 123, 132 and 213. After insertion of A, the obtained arrangements are: 1A23, 12A3, 1A32, 13A2, etc.
- for each arrangement, the corresponding way is calculated; for example, for the arrangement 12A3, the corresponding way is calculating by adding the following distances: 1 to 2, 2 to A and A to 3.
- all the resulting ways are sorted, and only the shortest one is kept.
- the Metaproximates of the reference point are the 2 points on right and left of the ref.point in the arrangement corresponding to the shortest way; for example, if the shortest way is 13A2 for ref. point A, the Metaproximates of A are 3 and 2.

As an example, here are some of Metaproximates for the 40-points sample:

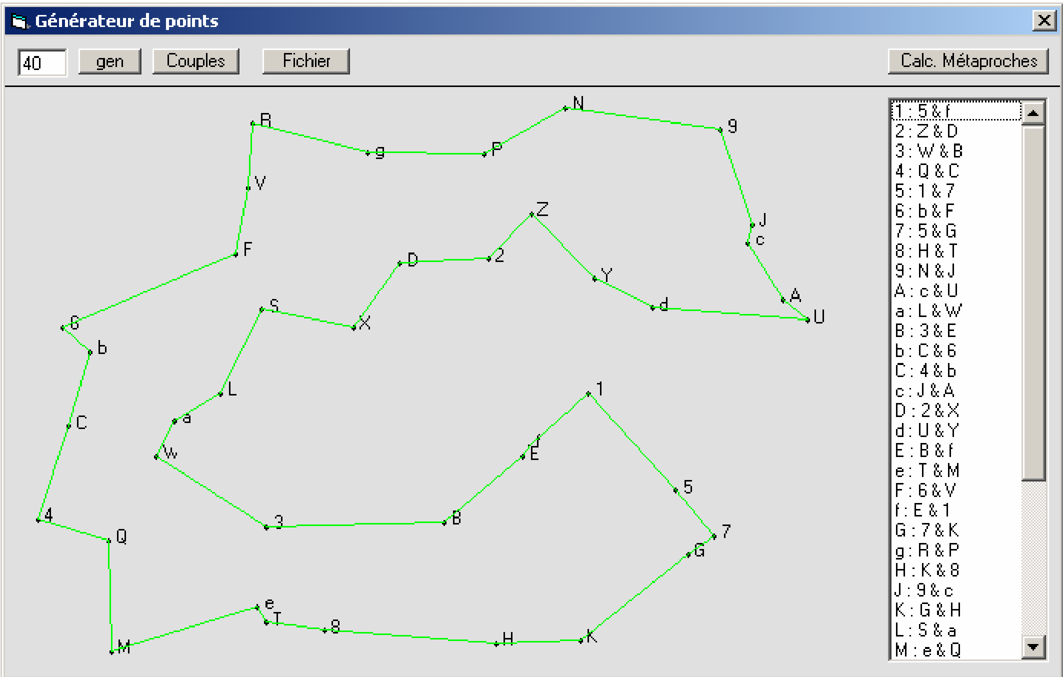
```
1 : 5 & f
2 : Z & D
3 : W & B
4 : Q & C
5 : 1 & 7
6 : b & F
7 : 5 & G
8 : H & T
9 : N & J
A : c & U
B : 3 & E
...
```

3) Final connection

The couples of metaproximate points are finally connected to their reference point, and the track showing the shortest way can be finally obtained. The value of the shortest way is calculated at this step.

For our 40-points sample, here is the final solution, taking into account the processing of the 3 steps:

```
Solution_40_points
 shortest way = 351076
 track : 157GKH8TeMQ4Cb6FVRgPN9JcAUdYZ2DXSLaW3BEf1
 -------------------
 other solutions :
 358822 - 157GKH8TeMQ4CWab6FVRgPN9JcAUdYZ2DXSL3BEf1
 367862 - 1dYZ2DXSLab6FVRgPN9JcAU75GKH8TeMQ4CW3BEf1
 368287 - 157GKH8TeMQ4C6FVRgPN9JcAUdYZ2DXSLbaW3BEf1
 370114 - 1dYZ2DXSb6FVRgPN9JcAU75GKH8TeMQ4CWaL3BEf1
--------------------
computation begin : 22:42:19
computation end : 22:42:21
--------------------
```



Advantage(s) of the new solution over the best existing solution (if possible, quantify)

The main advantage of the new solution is the possibility to calculate the shortest way in a reasonable computation time, on a standard monoprocessor (for example less than 6 seconds of CPU time for 100 points, on a 800 MHz processor).

Calculating all the "horizons" through their signatures, and then processing them through an "swapping-insertion" process is very fast and depends only on the number n of points. The corresponding computation time grows as n(n-1) instead of n! for the "blind" solutions that are based on the computation of the total number of possible solutions.

Another advantage is the simplicity of the 3 main algorithms of the solution.

Drawback(s) of the new solution (if possible, quantify)

No drawback.

Is there any reason to believe that the new solution is of particular interest to competitors? If so, which competitors and for what particular reasons?

This solution is of particular interest for dimensioning networks, for optimizing the transfer of packets through a TRANSPAC-like system, for optimizing the use of power cables to connect energy-distributing devices, to optimize the industrial use of raw material, etc. This solution is of particular interest to all competitors who have to dimension networks, to transport data packets through a network or who want to optimize power cable installation – to mention only some few examples of use of this solution.

Could the solution be implemented on products of other Business Units or other products?

The solution could be implemented on products of MSD and FSD.

Other useful information

General useful information on TSP, its history, solutions and applications can be found at
http://www.tsp.gatech.edu/

A lot of web sites give useful informations about TSP and strategies for its solution. See for example:
http://fr.wikipedia.org/wiki/Probl%C3%A8me_du_voyageur_de_commerce
http://www.polytech-lille.fr/~vmagnin/coursag/voyageur/voyageur.html
http://www.crdp.ac-grenoble.fr/imel/jlj/pvc/infogene.htm