

## Desarrollo de un programa en lenguaje de alto nivel para resolver el problema de Múltiples TSP

David de la Fuente<sup>1</sup>, Jesús Lozano<sup>1</sup>, Raúl Pino<sup>1</sup>, Paolo Priore<sup>1</sup>

<sup>1</sup> Dpto. de Administración de Empresas de la Universidad de Oviedo. Escuela Politécnica de Ingeniería de Gijón. Campus de Viesques, s/n, 33204 Gijón  
[david@uniovi.es](mailto:david@uniovi.es); [lozano@uniovi.es](mailto:lozano@uniovi.es) ; [pino@uniovi.es](mailto:pino@uniovi.es) , [priore@uniovi.es](mailto:priore@uniovi.es)

**Palabras clave:** Multiple TSP, Heurísticas, Perl, Zonificación

### 1. Introducción.

Son frecuentes en la práctica que se presenten este tipo de problemas, donde hay varios viajantes, varias rutas simultáneas de camiones o autobuses, etc. donde se trata de minimizar la distancia recorrida\*. Tras varias experimentaciones, el problema que denominaremos MTSP resulta particularmente difícil porque puede proporcionar soluciones no equilibradas, es decir soluciones en las que unas rutas tengan más puntos de parada o más distancia que la media. En la literatura, se ha abordado este problema aplicando variados métodos heurísticos: metaheurísticas (Petersen and Madsen, 2009), colonias de hormigas (Bell and McMullen, 2004; Donati et al., 2008), algoritmos genéticos (Tao, 2008), redes neuronales artificiales (Masutti and De Castro, 2009), cúmulos de partículas (Marinakis and Marinaki, 2010), algoritmos meméticos (Duan and Yu, 2007), programación por objetivos (Hong. and Park, 1999), etc.

En orden a resolver este problema, nosotros hemos desarrollado un método heurístico que se ha dividido el algoritmo en dos fases principales.

En la primera fase se trata de zonificar los puntos de destino agrupándolos en zonas que serán asignadas a un solo vendedor o camión. De los algoritmos disponibles para resolver este problema y aplicándolo a problemas de tamaño mediano (200 puntos) hemos optado por una heurística simple de intercambio aleatorio de pares entre zonas. Se parte de un número conocido de rutas y se asignan inicialmente todos los puntos a una zona. Se produce un intercambio de un punto de una zona a otra y se evalúa la suma total de las distancias cruzadas entre todos los puntos de una zona y de todas las zonas. De esta manera se evita el tener que calcular centroides. Si la distancia total es inferior a la anterior el cambio permanece y si no, el intercambio es deshecho. Así, repitiendo iterativamente el proceso se llega con una

---

\* Este trabajo se deriva de la participación de sus autores en un proyecto de investigación financiado por el Ministerio de Fomento, con referencia MFOM-08-E12/08, titulado "Análisis, Desarrollo y Evaluación de Sistemas Inteligentes de Transporte de Mercancías en un Entorno Intermodal".

cierta rapidez a una zonificación satisfactoria. El inconveniente de este método es que en principio ese bucle es infinito y se trata de meterle un límite de iteraciones sin mejora que rompa el bucle y pase a la siguiente fase.

En la segunda fase nos encontramos con que todas las zonas están pobladas y más o menos equilibradas, pero el orden de los puntos no responde a ningún criterio especial. Entonces, se pueden desarrollar estrategias de TSP simple para cada zona, como podría ser el intercambio aleatorio de pares o método de los ahorros (en nuestro caso este es el que se ha desarrollado), en que la ordenación varía por pares que se deshacen si se produce un incremento de la distancia de la ruta circular de una zona. Si bien los TSP simples se pueden desarrollar en paralelo, al ser separables, también se puede seguir un orden rotatorio (que es el que hemos utilizado) que está basado en la igualdad en el número de puntos de cada zona.

## 2. Descripción del problema

Aplicaremos esta metodología a la resolución de un problema real propuesto por una empresa logística. Este operador logístico recibe en las 7 campas repartidas por todo el estado (Ciempozuelos, Tarragona, Santander, ...) vehículos procedentes de fabricantes e importadores, y tiene que distribuirlos a los más de 3000 concesionarios existentes en España y Portugal. El transporte se realizará por carretera en la península y en barco hacia las islas Baleares o Canarias.

Las restricciones principales del problema son las siguientes:

- Inicialmente se supondrá un número infinito de camiones, aunque en una fase posterior del proyecto, se tendrá en cuenta la flota real disponible en cada momento.
- Existen varios tipos de camiones y medios de transporte, con diferente capacidad, si se establece una capacidad estándar del 100% para un camión de tamaño medio. En la tabla 1 se puede observar la diferente capacidad de los distintos camiones o medios de transporte en relación al camión medio.

**Tabla 1.** Capacidad de los camiones y medios de transporte disponibles.

<b>Tipo Camión</b>	<b>Porcentaje de tamaño</b>
C/R MIXTO	100
C/R PILARES	100
C/R TIJERAS	100
CANGURO CAMETAL	110
EUROLOHR 21	110
EUROLOHR 22	110
EUROLOHR 22 SIN MARQUESINA	100
EUROLOHR 22 Tijeras Tras	110
EUROLOHR 23	110
FORMULA 2000	110
GABARRA CON MODIFICACIÓN	85
GABARRA SIN MARQUESINA	80
GABARRA CERRADA	50
GRUA 2 VEHICULOS	20

GRUA SIMPLE	40
METAGO/SUPERTRANS/VARIOTRANS	100
RIGIDO Pilares 5 vhs cable	50
RIGIDO Tijeras 4 vhs cable	40
RIGIDO Tijeras 5 vhs	50
RIGIDO Tijeras 5 vhs cable	50

- De la misma manera, el tamaño de los vehículos a transportar es muy variable, desde el micro-coche hasta el mini camión. En la tabla 2 se muestra el número de unidades de cada tamaño que caben en un camión estándar.

**Tabla 2.** Capacidad de un camión estandar en función del tipo de vehículo transportado.

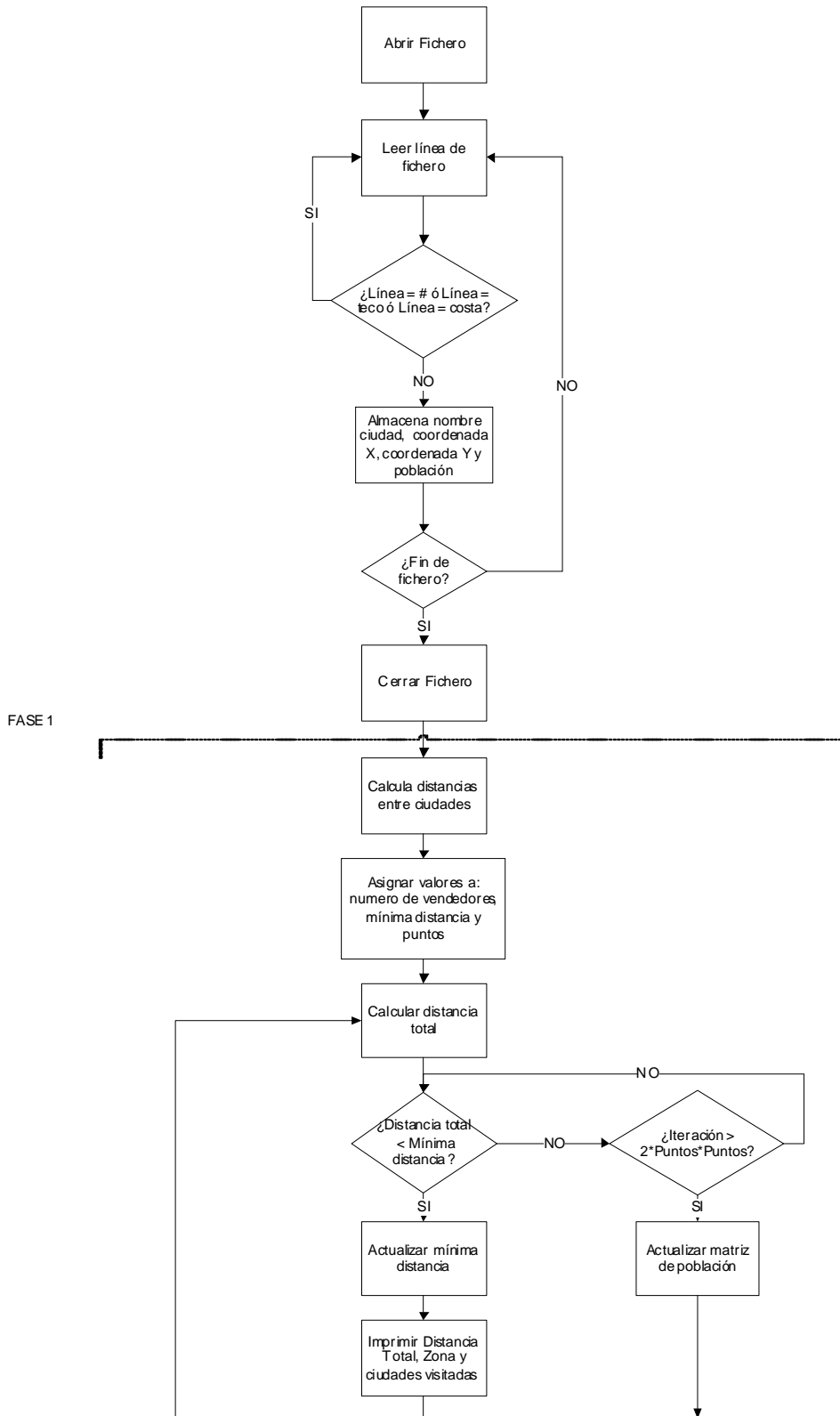
<b>Tipo de vehículo</b>	<b>Máximo número de unidades</b>
Micro coche	11
Turismo pequeño	10
Turismo mediano	9
Turismo grande	8
Gran turismo	7
Monovolumen pequeño. Furgoneta rendida	7
Turismo especialmente grande	6
Monovolumen Grande. Todo terreno pequeño, furgoneta	6
Gran monovolumen. Todo terreno grande	5
Furgoneta chasis / pequeña	4
Furgon	3
Maxi	2
Mini Camión	1
Micro Coche (CC 12)	12
Micro Coche (CC 13)	13

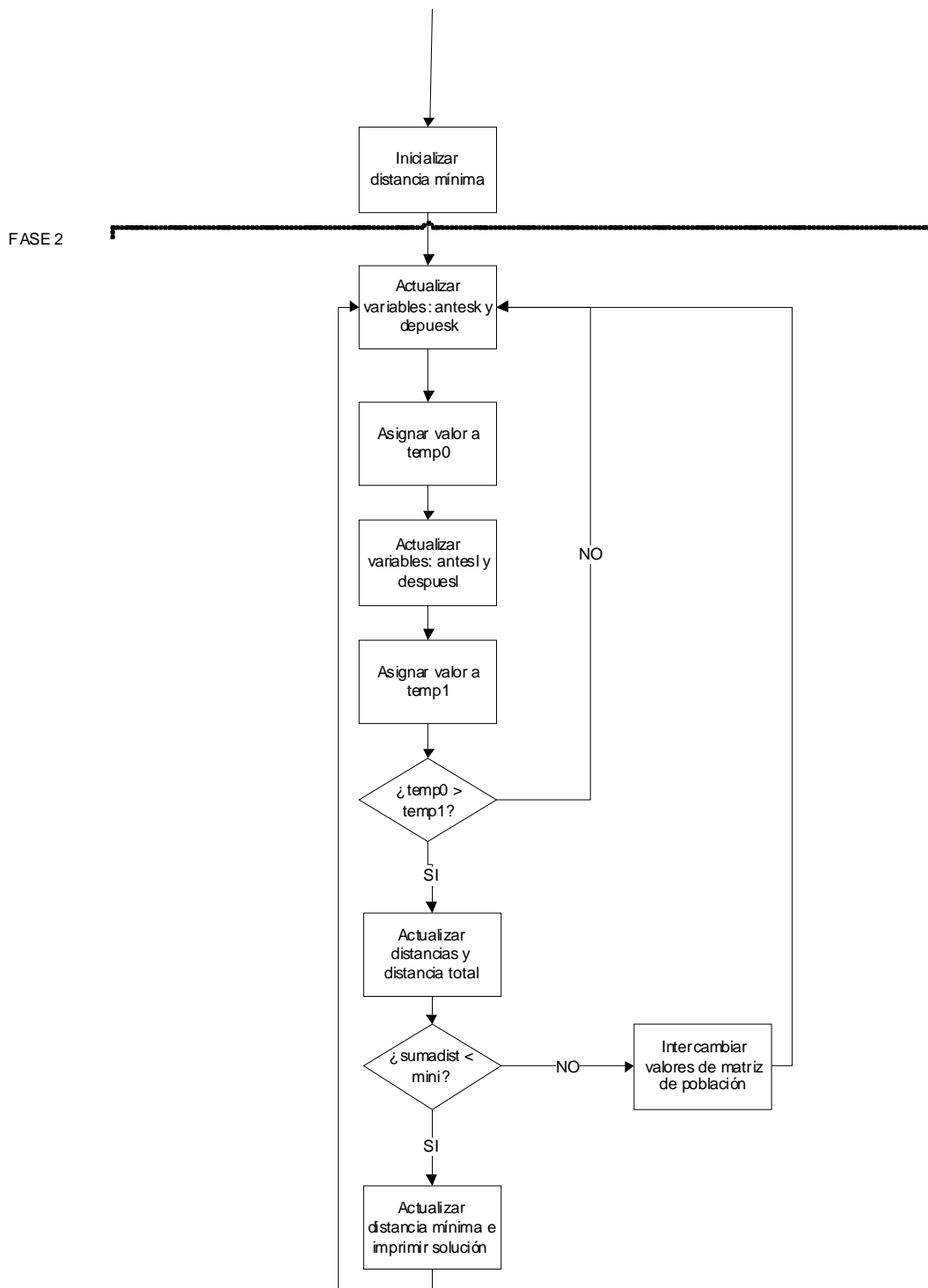
Como datos de partida, se dispone de la lista de vehículos a distribuir. Para cada caso, se tiene información de: número de pedido, tipo de vehículo (marca y modelo), zona de origen, origen, zona de destino, destino (concesionario), fecha máxima de entrega.

Asimismo, se dispone de una base de datos con las distancias entre todos los distintos puntos de origen y destino, para que el algoritmo pueda optimizar las distancias recorridas.

### **3. Flujograma**

El flujograma propuesto se puede ver en la siguiente figura.





En dicho flujograma se presentan los siguientes elementos:

En la primera fase, de zonificación, se realiza un intercambio aleatorio entre puntos de distintas zonas, siendo el criterio de aceptación la suma de todas las distancias cruzadas de puntos de una misma zona. En el momento en que dicha suma sea mínima y transcurran una serie de iteraciones sin mejora, se consideran las zonas formadas y se pasa a la siguiente fase.

En la segunda fase, de múltiples TSP en cada zona, se va iterando por cada zona y en cada una se eligen dos puntos distintos al azar A y B. Se computa:

$$\text{tmp0} = \text{dist}(A, \text{prevA}) + \text{dist}(A, \text{postA}) + \text{dist}(B, \text{prevB}) + \text{dist}(B, \text{postB})$$

y se compara con:

$$\text{tmp1} = \text{dist}(B, \text{prevA}) + \text{dist}(B, \text{postA}) + \text{dist}(A, \text{prevB}) + \text{dist}(A, \text{postB})$$

si tmp1 es menor o igual a tmp0 se procede a intercambiar los puntos:

swap(A,B)

De esta forma por el método de los sucesivos ahorros nos acercamos heurísticamente a la solución óptima. El programa puede darse por finalizado tras una serie de iteraciones sin mejora y se presenta un listado de zonas (viajantes) con los puntos ordenados en la secuencia resultante. Será irrelevante el punto de comienzo, posteriormente se sigue la secuencia de la lista, considerando que ésta es circular.

En los anexos se puede ver el listado del programa en Perl (Anexo 1), un ejemplo de los datos de las ciudades (Anexo 2), y un ejemplo de la salida que proporciona el programa (Anexo 3).

#### **4. Conclusiones**

En este trabajo, se ha desarrollado un método heurístico para la resolución del conocido problema del viajante múltiple (MTSP). Se ha aplicado a un caso real de distribución de vehículos desde las campas de almacenamiento a los concesionarios, teniendo en cuenta las distintas restricciones de tipos de vehículos, tamaños de los mismos y tipos de camiones (capacidades distintas).

#### **Referencias**

Bell, J.E. and McMullen, P.R. (2004): "Ant colony optimization techniques for the vehicle routing problem". *Advanced Engineering Informatics* 18, pp. 41–48.

Donati, A.V.; Montemanni, R.; Casagrande, N.; Rizzoli, A.E.; Gambardella, L.M. (2008): "Time dependent vehicle routing problem with a multi ant colony system". *European Journal of Operational Research*, 185, pp. 1174–1191

Duan, H. and Yu, X. (2007): "Hybrid Ant Colony Optimization Using Memetic Algorithm for Traveling Salesman Problem". *Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL 2007)*, pp. 92-95.

Hong, S.C. and Park, Y.B. (1999): "A heuristic for bi-objective vehicle routing with time window constraints". *International Journal of Production Economics*, 62, pp. 249-258.

Marinakis, Y. and Marinaki, M (2010): "A Hybrid Multi-Swarm Particle Swarm Optimization algorithm for the Probabilistic Traveling Salesman Problem". *Computers & Operations Research*, 37 pp.432-442.

Masutti, T.A.S. and De Castro, L.N. (2009): "Neuro-immune approach to solve routing problems". *Neurocomputing*, 72, pp. 2189–2197.

Petersen, H.L. and Madsen, O.B.G. (2009): "The double travelling salesman problem with multiple stacks - Formulation and heuristic solution approaches". *European Journal of Operational Research* 198, pp.139–147

Tao, Z. (2008): "TSP Problem solution based on improved Genetic Algorithm". *Proceedings of Fourth International Conference on Natural Computation*, pp. 686-690.

## ANEXO 1. Código fuente en Perl.

```
#!/usr/local/bin/perl -w

open (IN, "<", "coordenadas") || die "Cannot open coordenadas:$!";
while (<IN>){
    next if (/^#/);
    next if (/^teco/);
    next if (/^costa/);
    chomp;
    $n++;
    @linea = split /\s+/, $_;
    $nombre[$n]=$linea[0];
    $x[$n]=$linea[1];
    $y[$n]=$linea[2];
}
close IN;

for $i (1..$n){
    for $j (1..$i){
        $d[$i][$j]=sqrt( ($x[$i]-$x[$j])**2 + ($y[$i]-$y[$j])**2 );
        $d[$j][$i]=$d[$i][$j];
    }
}

$N = 5;          # NUMBER OF SALESMEN!
$PUNTOS = $n;

for $i (1..$PUNTOS){
    for $j (1..$N){
        if ($i <= $j*$PUNTOS/$N && $i > ($j-1)*$PUNTOS/$N){
            $p[$j][$i]=$i;
            $h[$i]=$j;
        }else{
            $p[$j][$i]=-1;
        }
    }
}

# $ind=9e99;
$MIN=9e99;
$sqmin=9e99;
for (;){
    $iteracion++;
    @r=();
    $suma=0;
    for $j (1..$N){
        for $i (1..$PUNTOS){
            next if ($h[$i]!=$j);
            next unless ($p[$j][$i]>0);
            for ($z=$i-1;$z>=0;$z--){
                next unless (defined $h[$z]);
            }
        }
    }
}
```

```

                if ($h[$z]==$j && $p[$j][$z]>0){
                    $r[$j] += $d[ $p[$j][$i] ][ $p[$j][$z] ];
                    last;
                }
            }
        }
        $suma += $r[$j];
    }
    $media = $suma / $NN;
    $sqerr=0;
    for $j (1..$NN){
        $sqerr += ($r[$j]-$media)**2;
    }
#   $pct_suma = (($suma - $MIN)/$MIN);
#   $pct_sqerr = (($sqerr - $sqmin) / $sqmin);
    if ($suma<=$MIN){
#       if ($pct_suma+$pct_sqerr<=$ind){
#       if ($sqerr<=$sqmin){
#           $k++;
#           $ind = $pct_suma+$pct_sqerr;
#           $MIN=$suma;
#           $sqmin=$sqerr;
#           print "\n$0: Problema de $PUNTOS puntos, con $NN vendedores\n";
#           print "Iteracion $iteracion, solucion $k, ";
#           printf "distancia %.3f, ", $suma;
#           printf "error %.3f\n", $sqerr;
#           for $j (1..$NN){
#               print "Viajante $j distancia ";
#               printf "%.3f: ", $r[$j];
#               for $i (1..$PUNTOS){
#                   if ($h[$i]==$j){ print "$i "; }
#                   if ($p[$j][$i]>0) { print "$p[$j][$i] "; }
#               }
#               print "\n";
#           }
#       }
#####
loop:

    $i = 1+ int rand($PUNTOS);
    $j = 1+ int rand($PUNTOS);
    if ($j==$i) { goto loop; }

    $ii= $h[$i];
    $jj= $h[$j];
    $z=$p[$ii][$i];
    $m=$p[$jj][$j];
    unless ($z>0 && $m>0) { goto loop; }
#   if ($z==0 && $m==0) { goto loop; }      # ||?

    $primero=$ultimo=$antesz=$despuesz=$ok=0;
    for $p (1..$PUNTOS){
        if ($h[$p]==$ii){
            if ($p[$ii][$p]>0){
                if ($antesz==0){ $primero = $p; }
                if ($p!=$i && $ok==0){ $antesz = $p; }
                if ($p!=$i && $ok==1 && $despuesz==0){ $despuesz = $p; }
                if ($p==$i) { $ok=1; }
                $ultimo = $p;
            }
        }
    }
    $p[$ii][0]= $p[$ii][$ultimo];
    if ($antesz==0){ $antesz=$ultimo; }
    if ($despuesz==0){ $despuesz=$primero; }
    $d1 = $d[$z][ $p[$ii][$antesz] ] + $d[$z][ $p[$ii][$despuesz] ];

    $primero=$ultimo=$antesm=$despuesm=$ok=0;
    for $p (1..$PUNTOS){
        if ($h[$p]==$jj){
            if ($p[$jj][$p]>0){
                if ($antesm==0){ $primero = $p; }
                if ($p!=$j && $ok==0){ $antesm = $p; }
                if ($p!=$j && $ok==1 && $despuesm==0){ $despuesm = $p; }
                if ($p==$j) { $ok=1; }
                $ultimo = $p;
            }
        }
    }

```



```

}
$p[$jj][0]=$p[$jj][$ultimo];
if ($antesm==0){ $antesm=$ultimo; }
if ($despuesm==0){ $despuesm=$primero; }
$d2 = $d[$m][$p[$jj][$antesm]] + $d[$m][$p[$jj][$despuesm]];
$temp0 = $d1+$d2;

$d1 = $d[$m][$p[$ii][$antesz]] + $d[$m][$p[$ii][$despuesz]];
$d2 = $d[$z][$p[$jj][$antesm]] + $d[$z][$p[$jj][$despuesm]];
$temp1 = $d1+$d2;

if ($temp0>=$temp1){
    $p[$ii][$i]=$m;
    $h[$m]=$ii;
    $p[$jj][$j]=$z;
    $h[$z]=$jj;
} else { goto loop; }
}
__END__

```

**ANEXO 2.** Fichero de coordenadas, aquí se presentan 8 ciudades de las más de 200 que tiene el fichero:

Adeje	-1825	-1365
Aguilas	184	-333
Albacete	158	-157
Alcala_Guadaira	-193	-338
Alcala_Henares	27	6
Alcantarilla	215	-270
Alcazar_S_Juan	41	-114
Alcobendas	5	13
...	...	...

### ANEXO 3. Salida del programa:

FASE 1

```

Distancia total 32121443.7260842581
ZONA 1: 1 2 3 4 5 6 7 8 9 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142
143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165
166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188
189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211
212 213 214 215 216 217 218 219 220 221 222 223
ZONA 2:
ZONA 3:
ZONA 4: 13
ZONA 5:
ZONA 6:
ZONA 7:
ZONA 8:

...

Distancia total 720709.42168516047
ZONA 1: 18 23 30 33 36 38 63 69 76 81 87 89 99 105 123 141 147 156 167 169 176 179 182 197 202
218 223
ZONA 2: 4 11 17 20 34 42 48 50 51 57 61 62 71 74 86 91 97 104 107 109 112 116 118 122 128 149
150 153 159 161 166 175 180 198 204 208
ZONA 3: 29 31 32 37 45 49 54 68 70 73 78 79 84 88 101 102 115 117 126 127 133 157 158 162 163
170 171 172 173 174 183 190 191 201 209 211 212 213
ZONA 4: 2 3 6 14 16 44 52 53 67 83 90 103 106 119 125 129 131 136 151 160 195 200 203 205

```

ZONA 5: 1 22 24 25 93 95 108 137 152 184 185 186 192  
ZONA 6: 5 7 8 9 19 21 26 28 47 56 58 60 75 80 85 94 98 111 113 120 130 142 144 148 165 178 187  
188 189 194 196 215  
ZONA 7: 27 43 55 72 82 92 100 110 121 132 135 138 139 145 146 155 168 181 207 210 214 222  
ZONA 8: 10 12 13 15 35 39 40 41 46 59 64 65 66 77 96 114 124 134 140 143 154 164 177 193 199  
206 216 217 219 220 221

FASE 2

```
./mtsp6: Iteración 2, Ahorro 468.37928190758366
Solución 1: Distancia total = 25687.709640827312
Viajante 1 -> distancia 3163.38505697927401: 18 23 30 33 36 38 63 69 76 81 87 89 99 105 123
141 147 156 167 169 176 179 182 197 202 218 223
Viajante 2 -> distancia 3868.80416821578694: 208 11 17 20 34 42 48 50 51 57 61 62 71 74 86 91
97 104 107 109 112 116 118 122 128 149 150 153 159 161 166 175 180 198 204 4
Viajante 3 -> distancia 2141.68825596431194: 29 31 32 37 45 49 54 68 70 73 78 79 84 88 101 102
115 117 126 127 133 157 158 162 163 170 171 172 173 174 183 190 191 201 209 211 212 213
Viajante 4 -> distancia 4650.45500018915745: 2 3 6 14 16 44 52 53 67 83 90 103 106 119 125 129
131 136 151 160 195 200 203 205
Viajante 5 -> distancia 999.746330035976856: 1 22 24 25 93 95 108 137 152 184 185 186 192
Viajante 6 -> distancia 3668.66185920540492: 5 7 8 9 19 21 26 28 47 56 58 60 75 80 85 94 98
111 113 120 130 142 144 148 165 178 187 188 189 194 196 215
Viajante 7 -> distancia 3388.62061289063628: 27 43 55 72 82 92 100 110 121 132 135 138 139 145
146 155 168 181 207 210 214 222
Viajante 8 -> distancia 3806.34835734676363: 10 12 13 15 35 39 40 41 46 59 64 65 66 77 96 114
124 134 140 143 154 164 177 193 199 206 216 217 219 220 221
```

...

```
./mtsp6: Iteración 55731, Ahorro 4.85668057792678116
Solución 455: Distancia total = 10330.7389331237672
Viajante 1 -> distancia 1293.0958836313539: 182 105 218 36 30 99 69 33 23 202 223 87 141 89
156 176 63 76 81 167 197 169 147 179 123 38 18
Viajante 2 -> distancia 1367.8787864096379: 198 118 208 17 150 128 204 107 61 4 180 42 112 86
166 175 51 50 11 104 62 57 109 116 71 159 20 153 48 161 149 91 97 122 74 34
Viajante 3 -> distancia 799.757091971762763: 78 211 68 32 170 183 29 126 84 209 133 73 79 37
117 102 54 173 88 115 191 163 158 127 49 31 162 172 70 174 171 213 101 201 157 190 212 45
Viajante 4 -> distancia 1649.99679448040837: 205 3 195 53 151 16 83 129 119 67 160 14 2 106 44
6 131 200 136 125 52 203 90 103
Viajante 5 -> distancia 942.96171962676496: 22 1 25 93 192 24 137 108 152 95 184 185 186
Viajante 6 -> distancia 1495.9422408920559: 5 85 58 7 194 60 120 28 47 144 165 189 19 21 196
187 56 188 8 111 148 9 130 98 80 142 75 113 94 215 26 178
Viajante 7 -> distancia 1213.66169394086054: 72 132 110 135 145 121 92 138 27 82 181 100 139
207 222 155 210 146 214 168 43 55
Viajante 8 -> distancia 1567.44472217092285: 10 59 77 206 124 143 154 220 199 134 66 65 13 177
35 64 114 140 41 164 39 217 221 193 46 216 40 96 12 15 219
```