RAIRO. RECHERCHE OPÉRATIONNELLE

J. CARLIER
P. VILLON

A new heuristic for the traveling salesman problem

RAIRO. Recherche opérationnelle, tome 24, n° 3 (1990), p. 245-253

<http://www.numdam.org/item?id=RO_1990__24_3_245_0>

Recherche opérationnelle/Operations Research
(vol. 24, n° 3, 1990, p. 245 à 253)

A NEW HEURISTIC FOR THE TRAVELING SALESMAN PROBLEM (*)

by J. CARLIER (1) and P. VILLON (1)

Abstract. – *In this paper we present a new heuristic for the Traveling Salesman Problem. This heuristic is based on a dynamic programming method computing the best tour among an exponential number of tours in polynomial time. Computational results illustrating the efficiency of the method are presented.*

Keywords : Traveling Salesman Problem; Dynamic Programming Method; Heuristic; Neighborhood Relation.

Résumé. – *On présente dans cet article une nouvelle heuristique pour le problème du voyageur de commerce. Cette heuristique est basée sur une méthode de programmation dynamique qui permet de calculer le meilleur tour parmi un nombre exponentiel. Les résultats de calcul qui sont rapportés illustrent l'efficacité de la méthode.*

Mots clés : Problème du voyageur de commerce; Méthode de Programmation Dynamique; Heuristique; Relation de Voisinage.

1. INTRODUCTION

Exact methods for solving the Traveling Salesman Problem (TSP), like polyhedral or branch and bound ones, can solve large-size problems if good suboptimal tours are used for initialization [7, 9, 10]. So good heuristics to provide these tours are necessary.

This paper presents a new heuristic for the TSP in the symmetric case. First, a neighborhood relation between permutations of a set of n elements is defined. Each permutation is associated with a tour and is valued by its cost. Using a dynamic programming method, the best neighbor of a given permutation is computed in $O(n^2)$. The neighborhood relation is then extended to tours and an $O(n^3)$ algorithm to compute the best neighbor of a

(*) Received in February 1990, revised in March 1990.
(1) U.R.A. C.N.R.S. HEUDIASYC, Université de Technologie de Compiègne, B.P. n° 649, 60206 Compiègne, France.

Recherche opérationnelle/Operations Research, 0399-0559/90/03 245 09/$ 2.90

given tour is obtained. Our heuristic consists in iteratively applying this algorithm. Numerical results presented show the efficiency of the method. Finally, more general neighborhood relations are considered and the previous algorithm is generalized to compute the corresponding best neighbor.

This paper is organized as follows. In Section 2 and 3 we define neighborhood relations and discuss their properties. Section 4 presents the heuristic and its results. In Section 5, some generalizations are presented. Finally, some ideas for future work are given in Section 6.

2. A NEIGHBORHOOD RELATION BETWEEN PERMUTATIONS

INTRODUCTION : With a permutation $s=(s(1), s(2), \ldots, s(n))$ on a set I of n elements is associated the tour $T_s=(s(1), s(2), \ldots, s(n), s(1))$. So to a relation between permutations corresponds a relation between tours. We introduce in this section a neighborhood relation having some nice properties presented below.

DEFINITION: *A permutation* σ is a neighbor of a permutation s if there exists an integer p such that:

1. $0 \leqq p \leqq n$,
2. $\sigma(1)=s(i_1), \ldots, \sigma(p)=s(i_p)$ with $i_1<i_2<\ldots<i_p$ and
3. $\sigma(p+1)=s(j_1), \ldots, \sigma(n)=s(j_{n-p})$ with $j_1>j_2>\ldots>j_{n-p}$.

This relation is not symmetric, but it is reflexive. The subset $K=\{s(i_1), s(i_2), \ldots, s(i_p)\}$ of I permits to associate σ with s. We will say that K transforms s to σ. K and p are not uniquely defined but it can be easily verified that either $i_p=n$ or $j_1=n$. Moreover, K and $KU\{s(n)\}$ are the only subsets transforming s to the same permutation σ.

PROPOSITION 1: *The number of neighbors of a permutation s is* 2^{n-1}.

Proof: This number is equal to the number of subsets of $I-\{s(n)\}$.

Q.E.D.

PROPOSITION 2: *The diameter of the neighbor graph is smaller than* $[\log_2 n]$, *where* $[x]$ *means the upper integer part of the real number* x.

Proof: By induction on n, we will prove that, for any couple of permutations s and σ, there exists a sequence of $k=[\log_2 n]$ subsets of I transforming s to σ.

Let us suppose it is true for any integer $1 \leqq m \leqq n-1$ and set: $n' = [n/2]$, $J = \{ \sigma(1), \sigma(2), \ldots, \sigma(n') \}$, $L = \{ \sigma(n'+1), \sigma(n'+2), \ldots, \sigma(n) \}$, $\tau = \sigma/J$ (the restriction of σ to the set J), $\omega = \sigma/L$.

Let us denote ω^* the mirror permutation of ω. By the inductive assumption, there exists a sequence of $k-1$ subsets $J_1, J_2, \ldots, J_{k-1}$ of J (respectively $L_1, L_2, \ldots, L_{k-1}$ of L) transforming s/J to τ (respectively s/L to ω^*). Let us denote $I_1 = J_1 \cup L_1, I_2 = J_2 \cup L_2, \ldots, I_{k-1} = J_{k-1} \cup L_{k-1}$ and $I_k = J$. Since the intersection of J and L is empty, the sequence $I_1, \ldots, I_{k-1}$ transforms s to a permutation σ' such that $\sigma'/J = \tau$ and $\sigma'/L = \omega^*$. Consequently, the sequence $I_1, I_2, \ldots, I_k$ transforms s to σ.

Q.E.D.

GEOMETRICAL INTERPRETATION: In order to simplify the presentation of the algorithm, we suppose, without loss of generality, that $I = \{ 1, 2, \ldots n \}$ and $s = (1, 2, \ldots, n)$. An immediate geometrical interpretation of the neighborhood relation is that, when the nodes $1, 2, \ldots, n$ are symbolically placed, in this order on an horizontal line, any vertical line meets the tour T_σ either at 0 or

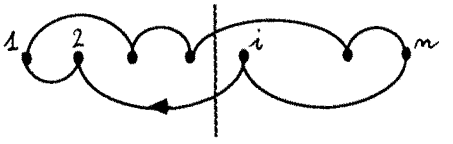

Figure 1. – Symbolical representation.

2 points (*fig.* 1). In particular, the vertical line going through $i (1 < i < n)$ meets the tour at another point also. We will say that T_σ is a 2-links tour of T_s. Consequently, for any i, T_σ is obtained by connecting two hamiltonian paths starting from i and going respectively through $\{ 1, \ldots, i \}$ and $\{ i, i+1, \ldots, n \}$ (*fig.* 1). This simple remark was used to build the procedure BEL, presented below, that computes the best neighbor of the permutation s. This method is known as the pyramidal method [7]. In this procedure, $S_{ij} (i < j)$ is the value of the best 2-links hamiltonian path going from i to j through $\{ 1, 2, \ldots, j \}$.

```
PROCEDURE BEL :
{ * We suppose that the initial permutation is (1, 2, ..., n) * }
Begin
  S11 := 0; S12 := v12;
  For j := 3 to n - 1 do
  Begin
    For i := 1 to j - 2 do Sij := Sij-1 + vj-1, j
    { * V is the valuation matrix of the edges * };
```

$S_{j-1,j} := \text{Min}\{S_{k,j-1} + v_{jk};\ k = 1, \ldots, j-2\}$;
End
$S_{nn} := \text{Min}\{S_{k,n-1} + v_{n-1,n} + v_{nk};\ k = 1, \ldots, n-2\}$;
Compute the best permutation by a backward procedure
End

This procedure is valid in the symmetric case. It consists in looking for an optimal path in a graph among the 2^{n-3} ones that correspond to permutations starting from 1 and ending to n.

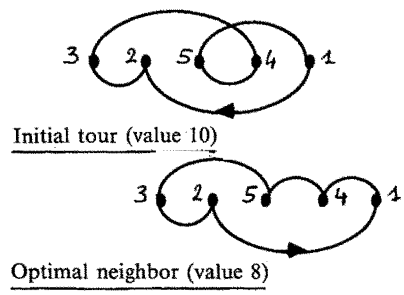

Figure 2. — Example (physical representation).

An example is given in figure 2. Five nodes 1, 2, 3, 4, 5 are on an axis with respective abscissae 4, 1, 0, 3, 2. Let us remark that we have here a physical representation of the problem which is different from a symbolical one (see *fig.* 1).

The optimal path for this example was computed by the procedure BEL. Figure 3 presents the associated graph, each node of which represents a hamiltonian path, as it is explained for nodes (2, 3), (2, 4) and (3, 4).

SPACE ALLOCATION AND COMPLEXITY OF THE PROCEDURE BEL: If distances are stored in a matrix V, the space complexity is $O(n^2)$. Otherwise, distances have to be computed and the space complexity is $O(n)$ since we need only two arrays for the procedure BEL: Pred (n) for predecessors and Pot (n) for potentials. Pred is constructed during the forward step and used during the backward step. We can restrict the dimension of Pred to n because only n nodes in the graph have more than one predecessor.

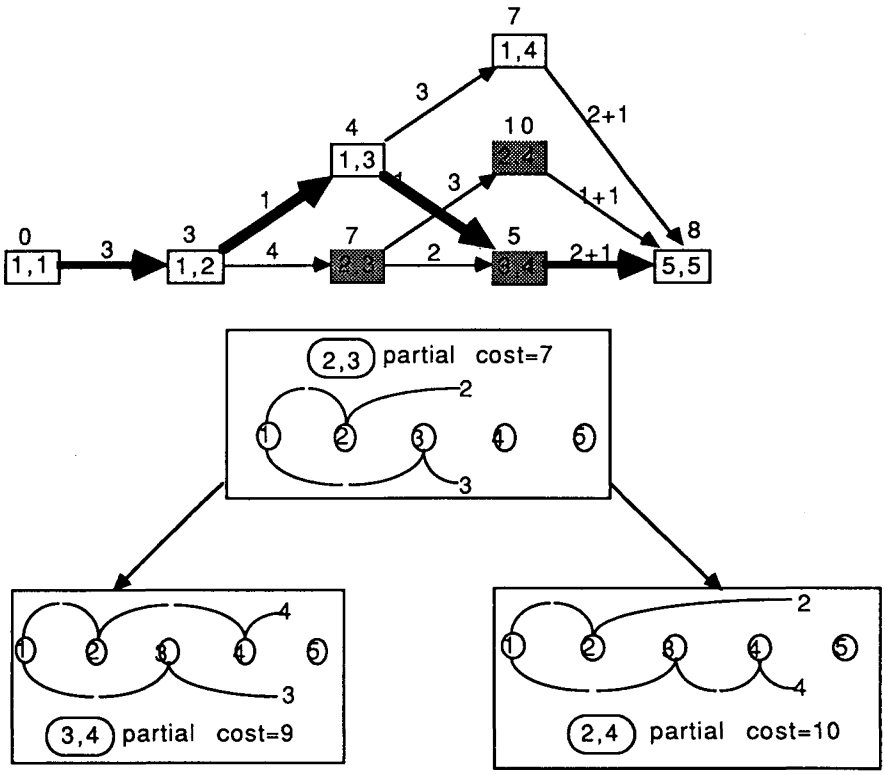

Figure 3. – Procedure BEL applied to the example (symbolical representation).

3. THE NEIGHBORHOOD RELATION BETWEEN TOURS

INTRODUCTION: A drawback of the previous neighborhood relation is that edges $[1, 2]$ and $[1, n]$ belong to all of the tours. To avoid this, we define below a neighborhood relation between tours by considering the n permutations associated with a tour.

DEFINITION: *A tour* T is a neighbor of a tour T' if there exist two permutations s and σ such that:

1. $T = T_\sigma$, $T' = T_s$;
2. σ is a neighbor of s.

The procedure BESTNEIGHBOR computes σ such that T_σ is the best neigbor of T_s.

```
  σ = FUNCTION BESTNEIGHBOR(s)
Begin
  opt := value (s);
  σ := s;
  For count = 1 to n do
  Begin
    neworder := BEL (s);
    If value (neworder) < opt then
    Begin
      σ := neworder;
      opt := value (neworder)
    End
    s := circular permutation (s)
    { * the circular permutation is a unitary shift * }
  End
End
```

The following proposition is easily proved.

PROPOSITION 3: *If a tour T is a local minimum for the procedure BESTNEIGHBOR, it is a local minimum for* 2-*OPT* [8].

We recall that 2-OPT consists in exchanging two edges of a tour for other two ones (see *fig.* 4).

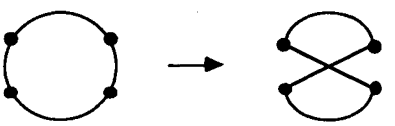

Figure 4. – 2-OPT exchanges.

The k-OPT procedure is based on k sequential exchanges [8].

4. THE HEURISTIC AND ITS RESULTS

Our heuristic consists in iteratively applying the algorithm computing the best neighbor: it is a steepest descent. We denote BELPERM this heuristic.

```
  PROCEDURE BELPERM
Begin
  initialize s; seuil := + ∞;
  while value (s) < seuil do
  Begin
    seuil := min (seuil, value (s));
    s := BESTNEIGHBOR (s);
  End
End
```

Though simulated annealing [4] gives good results, it seems that the best heuristic for the symmetric TSP is the one of Lin and Kerninghan based on k-OPT procedure [8]. So we compared our heuristic to the latter one.

Given an initial tour T_0, the value of the final tour obtained with BELPERM is slightly better than the one obtained by Lin and Kerninghan heuristic. However, the former computational times are much greater than the latter ones. So we suggest to use BELPERM not alone but coupled with the Lin and Kerninghan procedure to unlock local minima.

The table 1 reports computational results. The examples come from Padberg and Hong [9] with 60 cities randomly distributed. The first column reports the best cost obtained by a sophisticated version of Lin and Kerninghan heuristic (in particular several initial tours are used). The corresponding tour is generally optimal [9]. The second column reports the cost obtained with the basic procedure described in [8]. The third column reports the cost obtained with the BELPERM procedure.

TABLE I

Best k-OPT	Basic k-OPT	BELPERM
6,330	6,463	6,330
6,559	6,559	6,665
6,344	6,470	6,402
6,324	6,575	6,412
6,101	6,328	6,207
5,960	6,058	5,976
6,439	6,530	6,577
6,191	6,361	6,257
6,484	6,613	6,689
6,235	6,390	6,578

It appears that BELPERM is a very good heuristic but it needs more refinements to be as efficient as the better methods. A cutting rule permits to improve BELPERM. It consists in eliminating Bellman graph nodes (see *fig.* 3) with a lower bound greater than the value of the best known tour. The lower bound of a node is obtained by adding the number of unvisited cities multiplied by the value of the smallest unused edge to its potential. This improvement permits to obtain empirically $O(n^3)$ for BELPERM (against $O(n^{2.2})$ for k-OPT [8]).

5. GENERALIZATION

Up to now, we have considered 2-links tours and proposed a method to compute the best 2-links tour in $O(n^3)$. This method can be generalized to compute $2p$-links tours in $O(n^{p+1})$. Of course, for $2p \geqq n$, these results permit

to obtain an exact method for the TSP, but which cannot be applied without pruning many nodes of the corresponding dynamic programming state-graph.

More promising is the case $p=2$ because the complexity then remains manageable: $O(n^4)$ for the dynamic programming method.

The nodes of the k-th step of the corresponding state graph are of three types:

- type 1: (k, i_1), (i_2, i_3) with $i_2 < i_3 < k$ and $i_1 < k$;
- type 2: (i_1), (i_2, i_3) with $i_2 < i_3 \leqq k$ and $i_1 \leqq k$ and $k \in \{i_1, i_2, i_3\}$;
- type 3: (i_1, i_2) with $i_1 < i_2 \leqq k$.

For instance, in type 1, (k, i_1) and (i_2, i_3) correspond to two paths going through $\{1, 2, \ldots, k\}$ exactly once.

When $p=2$, we have the following impressive proposition:

PROPOSITION 4: *The diameter of the corresponding* 4*-links graph is less than* $[[\log_2 n]/2]$.

Proof: This follows from the fact that two nodes of the 2-links graph joined by a path of length 2 are neighbors in the 4-links graph.

Q.E.D.

Preliminary numerical experiments show that this method is very time consuming but always better than basic k-OPT.

6. CONCLUSION AND FUTURE WORK

Our method has very nice theoretical properties. In particular the diameter of the neighbor graph is $[\log_2 n]$. From a practical point of view it works very well. However the computational time is important. So this method can be used to unlock local minima of other heuristics.

At the moment the best version of Lin and Kerninghan gives better results, but many improvements are possible. For instance cutting the Bellman graph by using lower bounds, setting a subset of edges, using generalizations, etc. Moreover this method is easily generalized to the asymmetric case for which heuristics are not so good. Consequently this approach is very promising for the asymmetric case.

REFERENCES

1. R. BELLMAN, Dynamic Programming Treatment of The Traveling Salesman Problem, *J.A.C.M.*, 1962, *9*, pp. 61-63.

2. R. BELLMAN, Dynamic Programming, Princeton University, New-Jersey, 1965.

3. C. BERGE, Graphs and Hypergraphs, *North Holland*, Amsterdam, 1973.

4. E. BONOMI and J. L. LUTTON, The *N*-city Traveling Salesman Problem: "Statistical Mechanics and Metropolis Algorithm", *S.I.A.M. rev.*, 1984, *26*, n° 4.

5. J. CARLIER and P. VILLON, A Well Solved Case of the Traveling Salesman Problem, Research Report, Université de Technologie de Compiègne, *8*, July 1987.

6. J. EDMONDS and E. L. JOHNSON, Matching, Euler Tours and the Chinese Postman, *Math. Program.*, 1973, *5*, pp. 88-124.

7. E. L. LAWLER, J. K. LENSTRA, A. H. G. RINNOOY KAN, and D. B. SHMOYS, The Traveling Salesman Problem, *Wiley*, Chichester, 1985.

8. S. LIN and B. W. KERNINGHAN, An Effective Heuristic Algorithm for the Traveling Salesman Problem, *Oper. Res.*, 1973, *21*, pp. 498-516.

9. M. W. PADBERG and S. HONG, On the Symmetric Traveling Salesman Problem: a Computational Study, *Math. Program. Stud.*, 1980, *12*, pp. 78-107.

10. M. W. PADBERG and G. RINALDI, Optimization of a 532-city Symmetric Traveling Salesman Problem, A.F.C.E.T. Combinatorial group 20th anniversary, *A.F.C.E.T. Publisher*, 1986.

11. H. D. RATLIFF and A. S. ROSENTHAL, Order Picking in a Rectangular Warehouse: a Solvable Case of the Traveling Salesman Problem, *Oper. Res.*, 1983, *31*, pp. 507-521.

12. J. FONLUPT and A. NACHEF, Dynamic Programming and the Graphical Traveling Salesman Problem. Research report, ARTEMIS Laboratory, Grenoble, 1989.