OPERATIONS RESEARCH
Vol. 26, No. 3, May–June 1978

0030-364X/78/2603-0434 $01.25

Some Examples of Difficult Traveling Salesman Problems

C. H. PAPADIMITRIOU

Harvard University, Cambridge, Massachusetts

K. STEIGLITZ

Princeton University, Princeton, New Jersey

(Received September 1976; accepted May 1977)

We construct instances of the symmetric traveling salesman problem with $n=8k$ cities that have the following property: There is exactly one optimal tour with cost n, and there are $2^{k-1}(k-1)!$ tours that are next-best, have arbitrarily large cost, and cannot be improved by changing fewer than $3k$ edges. Thus, there are many local optima with arbitrarily high cost. It appears that local search algorithms are ineffective when applied to these problems. Even more catastrophic examples are available in the non-symmetric case.

MANY WORKERS, including Croes [5], Bock [2], Lin [8], Reiter and Sherman [11], and Lin and Kernighan [9], have reported the successful application of local search algorithms to the traveling salesman problem (TSP). Cook [4] and Karp [7], however, introduced a theory of complexity that shows that the TSP belongs to the class of NP-complete problems, which are seemingly of some inherent difficulty. More recently, Sahni and Gonzales [13] showed that the ϵ-approximate relaxation of the TSP is also NP-complete; and we [10] have shown that, unless $P=NP$, local search algorithms having polynomial time complexity per iteration cannot guarantee to solve the ϵ-approximate TSP. We are forced to conclude that the local search heuristics are not always as effective as they seem to be on "random" or "typical" test problems. The purpose of this paper is to construct instances of the TSP for which local search heuristics are ineffective. A review of work on the TSP is given in [1].

We use the terminology of Lin [8]. For any integer $k \geqq 2$, a *k-change* of a tour is another tour that differs from the given one in at most k edges. The neighborhood structure that assigns to each tour the set of its k-changes is called N_k. Local search algorithms using N_k are called *k-change search,* and local optima of these algorithms are referred to as *k-opts.* Lin used pseudo-random starting tours and obtained especially good computational results for 3-change search. Later, Lin and Kernighan [9] described what appears to be the best local search algorithm available today: They pursue successful transformations of a given tour to arbitrary depth, thus taking advantage of the problem data to define a good neighborhood of a given tour.

In Section 1 we construct instances of the symmetric TSP that are difficult for local search algorithms. The constructions are motivated by two very intuitive principles:

1. If an instance has a very large number of local optima with respect to some neighborhood structure N, and a unique global optimum that is much better, then this is a difficult instance with respect to local search using N.

2. If an instance of the TSP is difficult with respect to k-change search for large values of k (e.g., k comparable to n), then this instance is difficult for local search algorithms in general.

In Section 2 we examine the triangle inequality TSP and illustrate one aspect of the fact that this restriction of the TSP is considerably easier than the general case. In Section 3 we consider the non-symmetric TSP and give constructions for this problem that are analogous to the ones of Section 1. Finally, in Section 4 we describe the results of computational experiments that verify the difficulty of solving these problems with local search.

1. A CLASS OF PERVERSE TSP's

The following construction is suggested directly by the proof in [10] that a restricted hamiltonian path problem is NP-complete. We begin with the definition of a structural element called a *diamond*.

Definition 1. A *diamond* is the undirected graph with 8 vertices and 9 edges shown in Figure 1. It is understood that if a diamond is a subgraph of a graph $G=(V, E)$, then only the vertices N, E, S, W (north, east, south, west) can be incident to the other edges of G. The fundamental property of the diamond is expressed in:

LEMMA 1. *If a diamond D is a subgraph of a graph G with a hamiltonian circuit C, then G traverses D in exactly one of the two modes illustrated in Figure 2. That is, if a circuit C enters the diamond from the north, it must leave from the south; and similarly with respect to the east-west vertices.*

Proof. Assume the Hamilton circuit touches the vertex N in Figure 1. Then it must traverse the south-west path to vertex u, for otherwise it would never visit u again. It must then continue on to W, where it cannot leave the diamond since the remainder of the diamond could not then be part of a hamiltonian circuit with the restriction that only S and E can be incident with the rest of the graph. Hence the circuit must continue from W to y. It must then visit x, or x would be stranded, then E and v and S. The argument for the east-west path is symmetric.

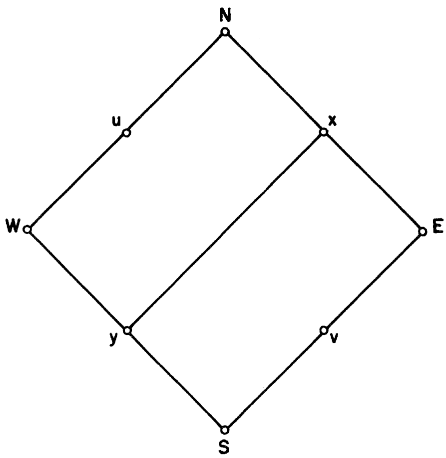

Figure 1. A diamond.

We now construct a family of graphs $G(k)$, with associated distance matrices, using k copies of the diamond (see Figure 3).

1. Make k copies of the diamond and call them D_i, $i=1, \cdots, k$. Call

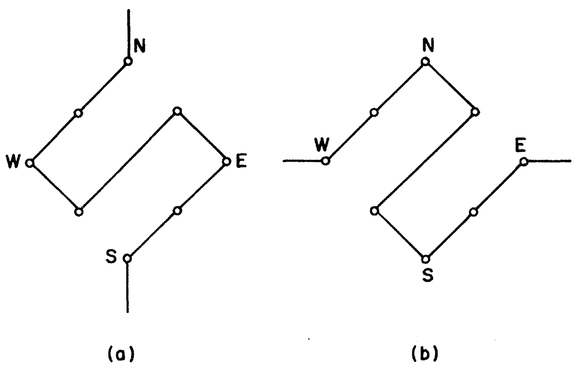

Figure 2. The two modes of traversing a diamond: (a) North-South mode; (b) East-West mode.

the north vertices of D_i, N_i, etc. Connect E_i to $W_{(i+1) \bmod k}$ with an edge, $i=1, \cdots, k$. This results in a graph with exactly one hamiltonian circuit. This circuit traverses each diamond in the east-west mode and we call it the *east-west circuit.* Assign to every edge on the east-west circuit a cost of one. Note that this leaves two edges in each diamond without an assigned cost; assign to them a cost of 0.

The idea is to next add many edges of 0 cost connecting the north and south vertices, but at the same time prevent any circuits that traverse diamonds in the north-south mode. This is accomplished by "isolating" a vertex, say N_1, by connecting it to other N_i and S_i only with edges of high cost.

2. Connect the set of $2k-1$ vertices $NS=\{N_2, \cdots, N_k, S_1, \cdots, S_k\}$ with $\binom{2k-1}{2}-k+1$ edges of cost 0, forming a complete subgraph from the vertices of NS omitting the edges (N_i, S_i), $i=2, \cdots, k$. Connect the remaining vertex N_1 to every $v \epsilon NS$ with an edge of cost M, an arbitrarily large positive integer.

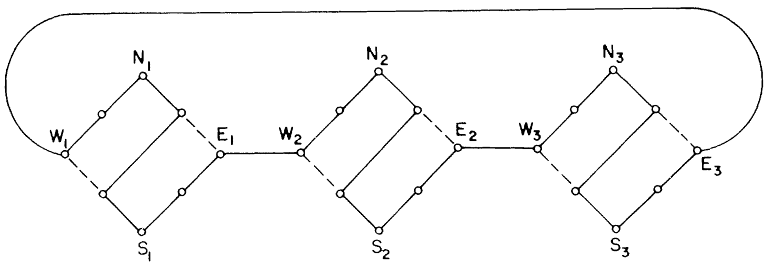

Figure 3. The East-West circuit of G(3), shown by solid edges.

3. To every pair of vertices of $G(k)$ not assigned an edge in 1 or 2 above, assign an edge with cost $2M$.

The essential property of the TSP defined by $G(k)$ is

LEMMA 2. *The instance of the TSP induced by the graph $G(k)$ has exactly one optimal tour of cost $n=8k$, given by the east-west circuit. The next best tours have cost $M+5k$, there are $2^{k-1}(k-1)!$ of them, and they differ from the optimal tour in exactly $3k$ edges.*

Proof. First consider the graph G' obtained from $G(k)$ by removing all edges of cost M or greater. There is but one hamiltonian circuit in G'—the east-west circuit. This follows because if any diamond is traversed in the north-south mode, they must all be, as specified in Lemma 1; and there is no edge in G' touching D_1 at N_1. Since M is arbitrarily large, the east-west circuit is uniquely optimal for $G(k)$, having no edges of cost greater than one.

Consider now the circuits with exactly one edge of cost M, and none of cost $2M$. These must traverse all D_i in the north-south mode and hence have cost $M+5k$. Each diamond in such circuits can be traversed in 1 of 2 orientations, and there are $(k-1)!$ orders in which they can be traversed.

Thus there are altogether $2^{k-1}(k-1)!$ distinct circuits of cost $M+5k$. Furthermore, these circuits must be next-best to optimal, since they have only one edge of cost M or greater and M can be chosen arbitrarily larger than any function of k.

Finally, observe that the optimal tour and any next-best tour have exactly $5k$ edges in common—the edges of D_i with cost one. They differ therefore in $3k$ edges, and the next-best tours are $(3k-1)-$opt; that is, they cannot be improved by changing fewer than $3k$ edges.

We have attempted to draw $G(4)$ in a transparent way, by first redrawing the diamond to bring vertices N and S to one side (Figure 4) and then arranging the diamonds in a circle (Figure 5).

Our construction of instances $G(k)$ satisfies the intuitive guidelines mentioned in the previous section. There is still some question, however,

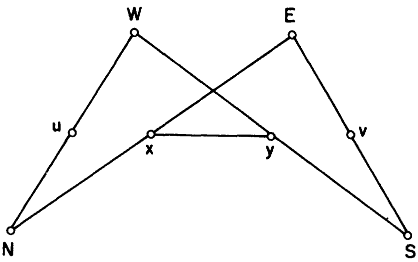

Figure 4. A redrawing of the diamond.

of just how well (or badly) local search behaves when confronted with such problems. Typically, a local search algorithm begins with a pseudo-random tour and pursues improvements found by searching in the neighborhood N. Each local optimum therefore has what might be termed a "region of attraction," from which it will be reached by the local search in question. It is conceivable that the single global optimum tour in $G(k)$ has a disproportionately large region of attraction—but this seems unlikely because it has all its edges of cost one, whereas the many next-best local optima have many edges ($3k-1$, to be precise) of cost 0.

2. THE TRIANGLE INEQUALITY TSP

When we restrict the problems under consideration to TSP's whose distance matrices satisfy the triangle inequality, there is an algorithm due to Christofides [3] that takes only polynomial time and at the same time guarantees solutions within 50% of optimal. It appears then that the triangle inequality TSP is considerably easier than the general case, and hence it becomes interesting to see whether the construction described above can be modified to work in this more restricted environment. We now present a simple argument to show that it cannot.

We define the *gap* of an instance of the TSP to be $g=(c_s-c_0)/c_0$, where $c_0>0$ is the cost of the optimal tour, and c_s is the cost of the second-best tour. An essential feature of the instance of the TSP induced by the graph $G(k)$ of the previous section is that it has an arbitrarily large gap. Nevertheless, the following theorem implies that for the triangle inequality TSP such a gap is unattainable.

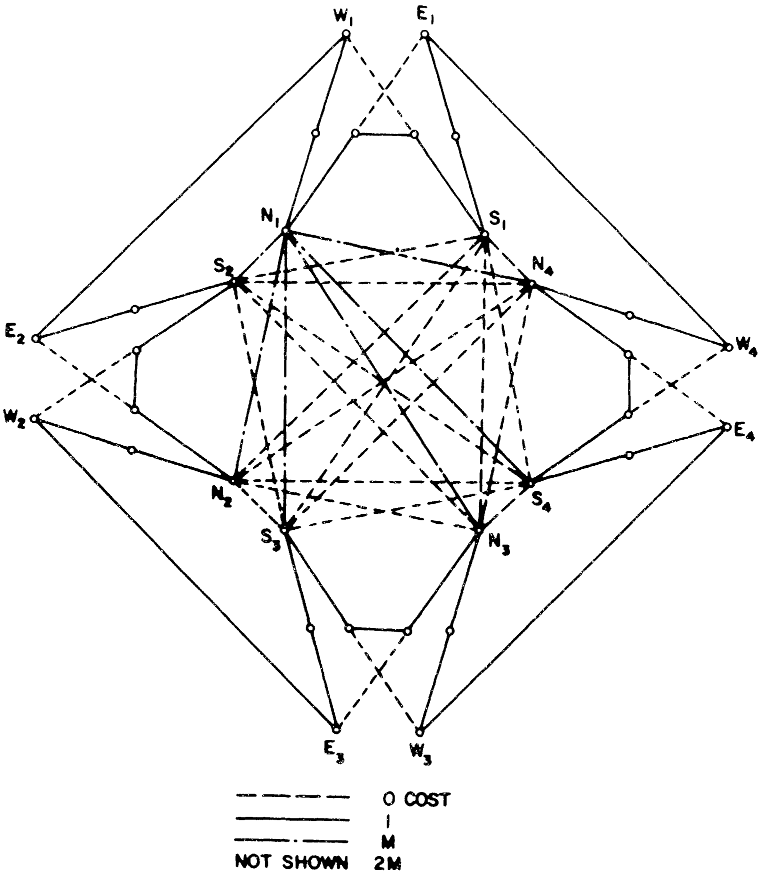

Figure 5. A drawing of G(4). The east-west tour is shown by solid edges.

THEOREM 1. *Let C be an instance of the triangle inequality TSP on n cities. Then the gap of C cannot be greater than $2/n$.*

Proof. Let T_0 be the optimal tour, and let b be the shortest edge in T_0. Then there is another tour T_1 using edges d, b, e in place of a, b, c (see Figure 6), where by the triangle inequality $d \leqq a+b$ and $e \leqq b+c$. (Here we let tours and edges stand for the respective costs.) Therefore, $T_1 - T_0 = d + e - a - c \leqq 2b$. Since b is the shortest edge of T_0, we have $T_0 \geqq nb$. Finally, if T_s is the second-best tour, we have a gap equal to

$$g = (T_s - T_0)/T_0 \leqq (T_1 - T_0)/T_0 \leqq 2b/nb = 2/n.$$

We cannot hope, therefore, to generate an infinite family of instances of the triangle inequality TSP with an arbitrarily large gap—in fact, with *any* constant gap. This is another bit of evidence pointing to the fact that the triangle inequality TSP is significantly easier than the general symmetric case. In Section 4 we present experimental evidence of yet another aspect of this fact.

3. THE NON-SYMMETRIC CASE

When the distance matrix is not restricted to be symmetric, we call the problem the *non-symmetric TSP*. Using the directed version of the

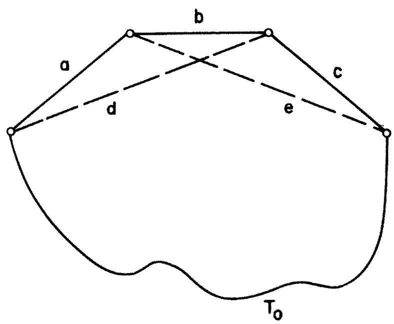

Figure 6. An optimal tour in a triangle inequality TSP.

diamond shown in Figure 7, an even more pathological example can be constructed in this general case as follows:

1. As before, make k copies of the directed diamond, calling them D_i, their vertices N_i, etc. Add the edges $(E_i, W_{(i+1) \bmod k})$. Put the cost of all these edges to 0. The east-west circuit thus has cost 0.

2. Put in the edges (N_j, S_i) for all $i = 1, \cdots, k$; and for $j = 2, \cdots, k$. These edges are also assigned a cost of 0.

3. Put in the edges (N_1, S_i), $i = 1, 2, \cdots, k$; and assign to these edges a cost of M.

4. An instance of the non-symmetric TSP is generated by setting the cost of all edges not mentioned above to $2M$. From the property of the directed diamond analogous to that of Lemma 1, we get an even more bizarre result than in the symmetric case.

LEMMA 3. *The instance of the non-symmetric TSP described above with $n=6k$ cities has an optimal tour with cost 0 and $(k-1)!$ next-best tours that are edge-disjoint from the global optimum and have (arbitrarily large) cost M.*

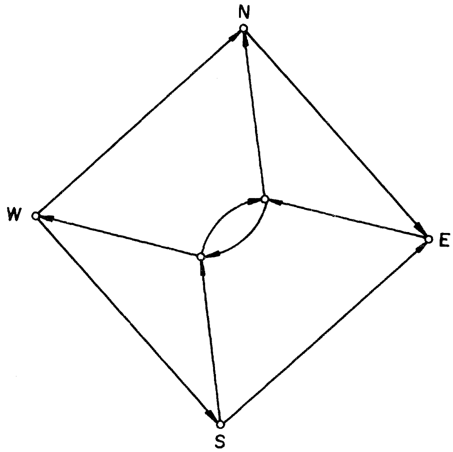

Figure 7. A directed diamond.

Proof. Analogous to that of Lemma 2.

4. COMPUTATIONAL EXPERIENCE

In the first set of results we applied the 3-opt algorithm proposed by Bock [2] and Lin [8] to the instances $G(k)$ of the TSP for the very moderate values of k shown in Table I. As indicated in the table, this algorithm failed to discover even the next-to-global optima of these instances.

TABLE I
RESULTS OF LOCAL SEARCH

	3-opt Lin [8]		Lin-Kernighan [9]		Triangle inequality, Lin-Kernighan [9]	
$k(n=8k$ cities)	3	5	3	5	3	5
Global optimum	0	0	0	0	24	3
Next-to-global optima	0	0	49	40	27	21
Other local optima	39	21	7	3	15	34
Total of trials	39	21	56	43	66	58

For the next set of results we programmed the local search algorithm described by Lin and Kernighan [9] with the following modifications:

1. The growth of the index i (see step 4 of [9]) was bounded by $6k$, which is ¾ of the total number n of cities. (This is by no means a severe restriction since a bound of n is implicit in the implementation of [9].)

2. The third part of the backtracking routine of step 6b in [9] was omitted in our implementation.

3. We omitted several other implementation details whose purpose is solely the reduction of computation time (and would not affect the quality of local optima).

Our implementation of the Lin-Kernighan algorithm was tested on a standard "tough" problem, the 8 by 8 knight's tour TSP [9]. The heuristic discovered the optimal solution in all 12 trials, thus equaling the impressive performance of the Lin-Kernighan implementation on this problem.

Next this algorithm was applied to the $G(k)$ instances of the TSP for the values of k shown. As indicated in the table, the algorithm failed to hit the global optimum even once, and very often ended up with the next-to-global optima. Since the Lin-Kernighan algorithm is perhaps the best local search algorithm known to date, the results of this experiment give us every reason to believe that the instances of the TSP constructed in Section 1 are indeed not susceptible to local search techniques.

A third set of experiments revealed to us another facet of the fact that the triangle inequality TSP is much easier than the general TSP. We constructed an instance of the $8k$-city TSP by setting the costs of the edges of $G(k)$ that were cheaper than M to 1 and finding the costs of all other edges by Floyd's [6] minimum distance algorithm. Naturally, since the resulting distance matrix satisfies the triangle inequality, by Theorem 1 we cannot expect any impressive gap from this instance. However, the existence and uniqueness of a global optimum remain unaffected. Hence it is interesting to determine whether application of local search algorithms to this instance will still result in suboptimal (though not as bad) solutions. We observed (see Table I) that our implementation of the Lin-Kernighan algorithm discovered the global optimum with non-vanishing frequency. This fact seems to indicate that, besides the uniformity property shown in Theorem 1 and the combinational properties exploited in the algorithms of Christofides [3] and Rosenkrantz et al. [12], the triangle inequality TSP possesses some positive properties related to local search that are missing from the general TSP. Discovering and exploiting such properties—possibly by designing a local search algorithm that works especially (and provably) well for the triangle inequality case—would be very interesting.

ACKNOWLEDGMENTS

This work was supported by NSF Grant GK-42048, U.S. Army Research Office-Durham under Grant DAAG29-75-G-0192, and an IBM Fellowship.

REFERENCES

1. M. BELLMORE AND G. L. NEMHAUSER, "The Traveling Salesman Problem: A Survey," *Opns. Res.* **16,** 538–558 (1968).
2. F. BOCK, "An Algorithm for Solving Traveling-Salesman and Related Network Optimization Problems," presented at the 14th National Meeting of the Operations Research Society of America, St. Louis, Oct. 24, 1958.
3. N. CHRISTOFIDES, "Worst Case Analysis of an Algorithm for the Traveling Salesman Problem," presented at the Carnegie-Mellon Conference on Algorithms and Complexity, Pittsburgh, April 7–9, 1976.
4. S. COOK, "The Complexity of Theorem Proving Procedures," *Proceedings of the 3rd SIGACT Symposium on the Theory of Computing*, pp. 151–158, 1971.
5. G. A. CROES, "A Method for Solving 'Traveling-Salesman' Problems," *Opns. Res.* **6,** 791–812, (1958).
6. R. W. FLOYD, "Algorithm 97: Shortest Path," *Commun. ACM* **5,** 345 (1962).
7. R. M. KARP, "Reducibility among Combinational Problems," in *Complexity of Computer Computations*, pp. 85–103, R. E. Miller and J. W. Thatcher (Eds.) Plenum Press, New York, 1972.
8. S. LIN, "Computer Solution of the Traveling Salesman Problem," *Bell System Tech. J.* **44,** 2245–2269 (1965).
9. S. LIN AND B. W. KERNIGHAN, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Opns. Res.* **21,** 498–516 (1973).
10. C. H. PAPADIMITRIOU AND K. STEIGLITZ, "The Complexity of Local Search for the Traveling Salesman Problem," *SIAM J. Comput.* **6,** 76–83 (1977).
11. S. REITER, AND G. SHERMAN, "Discrete Optimizing," *J. Soc. Indust. Appl. Math.* **13,** 864–889 (1965).
12. D. J. ROSENKRANTZ, R. E. STEARNS, AND P. M. LEWIS, "An Analysis of Several Heuristics for the Traveling Salesperson Problem," *SIAM J. Compt.* **6,** 563–581 (1977).
13. S. SAHNI, AND T. GONZLES, "*P*-Complete Approximation Problems," *J. Assoc. Comput. Machinery* **23,** 555–565 (1976).