REVUE FRANÇAISE D'AUTOMATIQUE, D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE. RECHERCHE OPÉRATIONNELLE

P. MILIOTIS
G. LAPORTE
Y. NOBERT

Computational comparison of two methods for finding the shortest complete cycle or circuit in a graph

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, tome 15, nº 3 (1981), p. 233-239.

<http://www.numdam.org/item?id=RO_1981__15_3_233_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 15, n° 3, août 1981, p. 233 à 239)

COMPUTATIONAL COMPARISON OF TWO METHODS FOR FINDING THE SHORTEST COMPLETE CYCLE OR CIRCUIT IN A GRAPH (*) (1)

by P. MILIOTIS (2), G. LAPORTE (3) and Y. NOBERT (4)

Abstract. — *Two methods for finding the shortest complete cycle or circuit in a graph are compared. The first method which is well known transforms the problem into a travelling salesman problem. Under the second approach, the problem is formulated directly as an integer linear program and then solved by relaxing most of its constraints. The results show the superiority of the second method.*

Keywords: Cycles, circuits, travelling salesman problem, integer programming.

Résumé. — *Cet article compare deux méthodes pour la découverte du cycle ou circuit complet le plus court dans un graphe. La première de ces méthodes est bien connue et consiste à transformer le problème en un problème du voyageur de commerce. Avec la seconde approche, on formule le problème directement comme un programme linéaire en nombres entiers; ce programme est résolu par un algorithme de relaxation de contraintes. Les résultats démontrent la supériorité de la seconde méthode.*

Mots clés : Cycles, circuits, problème du voyageur de commerce, programmation en nombres entiers.

1. INTRODUCTION

The *Travelling Salesman Problem* (TSP) is well known. The practical problem to which the TSP is generally related consists in finding the shortest route for a salesman wishing to visit n cities once and only once. To each pair (i, j) of cities, one associates a distance c_{ij}. The problem is said to be symmetrical whenever $c_{ij}=c_{ji}$ $(i \neq j)$ and asymmetrical otherwise. It is Euclidean if $c_{ik}+c_{kj} \geqq c_{ij}$ for $i, j, k=1, \ldots, n$.

The problem can also be defined in terms of graph theory. We consider the symmetrical case first. Let $G=(N, E)$ be a graph consisting of a set N of nodes and of a set E of edges. A Hamiltonian cycle is defined as a cycle which goes

(*) Received March 1980.
(1) The authors wish to thank the Canadian Natural Sciences and Engineering Research Council (Grant A4747) and the Quebec Government (F.C.A.C. program) for their financial support.

(2) National Energy Council, Athens, Greece.
(3) École des Hautes Études commerciales de Montréal, 5255 avenue Decelles, Montréal, Québec, Canada.
(4) Université du Québec à Montréal, Québec, Canada.

through each node of G exactly once. The TSP consists in finding the shortest Hamiltonian cycle in G. In asymmetrical problems, E is a set of arcs (directed edges) and the TSP consists in finding the shortest Hamiltonian circuit in G.

The TSP is a special case of the more general problem which consists in finding the *shortest complete cycle* (*circuit*) in G, i.e. the shortest cycle (circuit) going through each node of G at least once. This problem will be referred to as the CCP.

When G is not complete (i.e. when not all possible edges or arcs are defined), there does not necessarily exists a Hamiltonian cycle (circuit) in G and one may wish, in some instances, to find the shortest complete cycle (circuit) in G. Even when G is complete, the TSP solution does not always yield the shortest complete cycle (circuit) in G.

This paper compares two algorithms for the CCP.

2. METHOD 1: TRANSFORMING THE CCP INTO A TSP

There exists a close relationship between the TSP and the CCP. If C has the Euclidean property and if G is complete, there is a solution to the CCP which is a Hamiltonian cycle (circuit) [1]. Using this property, Hardgrave and Nemhauser [6] have shown that a CCP solution can be obtained by solving a TSP, even if G is not complete, with the Euclidean distance matrix $C'=(c'_{ij})$ derived from C by replacing each c_{ij} by the length of the shortest path between i and j.

This method may be described in three basic steps.

Step 1: Transform G into G' by finding all the shortest paths in G (*see* [10]).

Step 2: Solve the TSP associated with G'.

Step 3: Identify the solution to the original problem from the solution obtained in step 2. If $(i, k, \ldots, l, j)$ is the shortest path from i to j in G and if (i, j) is contained in the optimal Hamiltonian cycle found in step 2, then the optimal complete cycle or circuit in G contains the sequence $(i, k, \ldots, l, j)$.

3. METHOD 2: SOLVING THE CCP DIRECTLY

There has been, to our knowledge, no serious attempt to solve the CCP directly, without first transforming it into a TSP. Our aim is to show that it is more efficient to use a more direct approach.

We suggest the following formulation for the CCP. Let us first eliminate from G all edges or arcs (i, j) which are not themselves the shortest path between i

and j. Such edges or arcs will indeed never be used in the optimal CCP solution. Let G'' be the resulting graph. Then, the CCP can be formulated as (P1) or (P2) according to whether the problem is symmetrical or not.

(P1) *Symmetrical problems:*

$$\text{Minimize} \sum_{i<j} c_{ij} x_{ij},$$

subjetc to:

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} - 2 y_k = 2 \qquad (k=1, \ldots, n), \tag{1}$$

$$\sum_{\substack{i\in S,\, j\in \bar{S} \\ \text{or } j\in S,\, i\in \bar{S}}} x_{ij} \geqq 2 \qquad (2 \leqq |S| \leqq n-2,\ S \subseteqq \{1, \ldots, n\}), \tag{2}$$

$$\left.\begin{array}{r} x_{ij}=0,\ 1 \text{ or } 2, \\ y_k \text{ non negative and integer} \end{array}\right\} \qquad \begin{array}{r}(3\,a)\\(3\,b)\end{array}$$

In this formulation, variables x_{ij} are only defined for the edges of $G''=(N, E'')$ and for $i<j$. Consider the graph $G^*=(N, E^*)$ where E^* is the set of edges obtained by taking each (i, j) of E'' x_{ij} times. G^* is meaningful only if $(3\,a)$ is satisfied; if (2) is also satisfied, G^* is connected [3], and if (1) and $(3\,b)$ are satisfied, the degree of each node is even. Therefore, from [5], G^* possesses an Euler cycle which is also a complete cycle.

At this point, it is worth noticing that we imposed constraints:

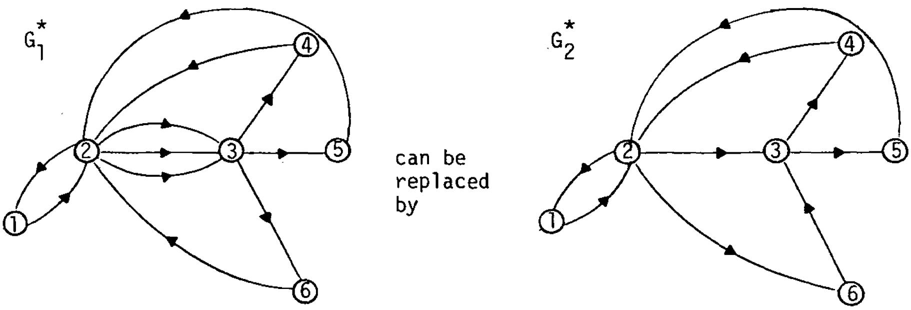
$$\sum_{\substack{i\in S,\, j\in \bar{S} \\ \text{or } j\in S,\, i\in \bar{S}}} x_{ij} \geqq 2 \qquad (2 \leqq |S| \leqq n-2,\ S \subseteqq \{1, \ldots, n\}), \tag{2}$$

in order to ensure that G^* is connected. The following constraints would have been adequate:

$$\sum_{\substack{i\in S,\, j\in \bar{S} \\ \text{or } j\in S,\, i\in \bar{S}}} x_{ij} \geqq 1 \qquad (2 \leqq |S| \leqq n-2,\ S \subseteqq \{1, \ldots, n\}), \tag{2'}$$

but are weaker than (2). Note that constraints (2′) can never be satisfied as equalities whenever (1) and (3) are imposed, hence (2) can be used instead of (2′).

It is interesting to note that in (P1), an optimal solution exists in which variables x_{ij} only take the values 0, 1 or 2. Although the formal proof of this property is not very complex, it is rather tedious because many cases have to be considered. In simple terms, it can be understood as follows: if a feasible solution contains a value of $x_{ij} \geqq 3$, for given i and j, then a solution at least as good with $x'_{ij} = x_{ij} - 2$ can be found. This is illustrated in the following figure: $c_{ij} = 1$ for all arcs shwon of the diagrams; c_{ij} is arbitrarily large otherwise. Arrows have been included only to facilitate the reading of the solution. One could reverse all arrows in figure since the graph is symmetrical.



The formulation for the asymmetrical case is similar to that of (P1).

(P2) *Asymmetrical problem:*

$$\text{Minimize} \sum_{i,j} c_{ij} x_{ij}$$

subject to:

$$\left.\begin{aligned} \sum_i x_{ik} - y_k = 1 \qquad (k=1, \ldots, n), \\ \sum_j x_{kj} - y_k = 1 \qquad (k=1, \ldots, n), \end{aligned}\right\} \tag{1}$$

$$\sum_{i \in S, j \in \bar{S}} x_{ij} \geqq 1 \qquad (2 \leqq |S| \leqq n-2, S \subseteqq \{1, \ldots, n\}), \tag{2}$$

$$x_{ij} \text{ non negative and integer,} \tag{3 a}$$
$$y_k \text{ non negative and integer.} \tag{3 b}$$

In this formulation, the y_k variables ensure that at each node, the incoming flow is equal to the outgoing flow. Merely imposing:

$$\sum_i x_{ik} \geqq 1 \qquad \text{and} \qquad \sum_j x_{kj} \geqq 1,$$

would not be sufficient. As in (P1) connectedness is garanteed by constraints (2) and, as in the previous case, the optimal solution possesses a complete circuit. Moreover, variables x_{ij} can this time take any non negative integer values.

Using these formulations, we suggest the following procedure for solving the CCP.

Step 1: Reduce G to G'' by dropping every edge or arc (i, j) which is not the shortest path between i and j.

Step 2: Solve the CCP associated with G'' by using P1 or P2. As in the case of the TSP [8, 9], we suggest that (P1) or (P2) be solved by first relaxing constraints (2) and (3) which should only be introduced as they are found to be violated. Integer solutions are obtained either by using Gomory cutting planes modified for integer arithmetic [9] or a branch and bound procedure [8]. As was shown in [9], the integrality and connectedness tests may be carried out in any order; however, it is more efficient to test connectedness first when using a cutting planes algorithm. With the branch and bound algorithm, tests for illegal subtours are only made once an integer solution is obtained.

Step 3: Identify the solution to the original problem: this is done by finding an Euler cycle or circuit associated with the optimal solution of step 2. (Here, one may use the algorithm proposed by Edmonds and Johnson [4].)

4. COMPUTATIONAL RESULTS

In order to compare the relative efficiency of the two methods, symmetrical and asymmetrical test problems were randomly generated by taking the c_{ij}'s uniformly on the interval]0, 100[. All distance matrices were complete. Five problems of each type (symmetrical and asymmetrical) and of each size ($n=20$, 30, 40, ...) were attempted by each of four possible algorithms:

- method 1, cutting planes;
- method 1, branch and bound;
- method 2, cutting planes;
- method 2, branch and bound.

In order to make the comparison between the two methods as fair as possible, the same type of algorithm was used in both cases: Miliotis' algorithms [8, 9] were applied directly in the case of method 1 and adjusted wherever appropriate to take into account the particular structure of the problems to be solved by method 2. (Christofides [2] advocates the use of LP based methods for symmetrical TSPs; for asymmetrical problems, the choice is less obvious.) In all cases, the same policies were used for branching, generating cuts, "purging" [7] ineffective constraints, etc. The computer used was the University of Montreal Cyber 173. The main results are presented in the following table.

Some attempts were unsuccessful. Failures occurred for two main reasons: either the preset time limit of 500 seconds was reached (this happened mainly with the branch and bound algorithm) or there was a lack of computer memory

TABLE

Computational results

n	Method 1 Transforming the CCP into a TSP					Method 2 Solving the CCP directly					
	Number of problems (1)	Maximum size of the inverse (2), (3)	Number of variables (3)	Number of simplex iterations (3)	Time (seconds) (3)	Number of problems (1)	Maximum size of the inverse (2), (3)	Number of variables (3)	Number of simplex iterations (3)	Time (seconds) (3)	Percentage of variables eliminated by method 2 (3)
Type of problem/algorithm: symmetrical/cutting planes											
20	5	25	190	65	2.2	5	30	60	40	0.9	68.4
30	5	40	435	118	10.7	5	47	94	53	3.1	78.4
40	5	49	780	150	20.1	5	60	138	93	6.1	82.3
50	4	62	1,225	184	40.9	4	76	180	125	13.0	85.3
60	3	73	1,770	215	66.6	4	94	222	166	25.8	87.5
70	3	87	2,415	346	161.1	3	106	252	192	39.2	89.6
80	1	93	3,160	279	134.8	2	118	308	212	51.8	90.3
90	–	–	–	–	–	2	135	355	253	74.7	90.9
100	–	–	–	–	–	2	148	386	409	312.3	92.2
Type of problem/algorithm: symmetrical/branch and bound											
20	5	24	190	74	2.2	5	40	60	76	1.9	68.4
30	4	37	435	282	23.8	5	58	94	99	2.6	78.4
40	4	47	780	538	77.6	5	82	138	212	12.1	82.3
50	4	60	1,225	383	118.5	5	102	181	228	14.1	85.2
60	4	72	1,770	694	356.8	5	121	223	320	27.2	87.4
70	–	–	–	–	–	4	103	252	977	111.7	89.6
80	–	–	–	–	–	3	116	307	1,212	152.8	90.3
90	–	–	–	–	–	1	131	365	2,449	441.0	90.9
Type of problem/algorithm: asymmetrical/cutting planes											
20	5	43	380	73	5.3	5	58	96	60	2.0	74.7
30	5	59	870	92	13.2	5	58	162	98	5.4	81.4
40	5	82	1,560	157	44.4	5	82	245	153	16.0	84.3
50	5	107	2,450	221	120.3	5	103	302	195	27.8	87.7
60	5	125	3,540	307	194.8	4	120	360	227	41.9	89.8
70	–	–	–	–	–	5	139	444	273	64.0	90.8
80	–	–	–	–	–	3	160	515	330	103.3	91.9
Type of problem/algorithm: asymmetrical/branch and bound											
20	5	42	380	83	6.1	5	30	96	46	0.7	74.7
30	5	58	870	94	10.3	5	45	162	111	3.4	81.4
40	5	82	1,560	177	41.8	5	60	245	134	5.4	84.3
50	5	105	2,450	367	179.3	5	75	302	233	13.5	87.7
60	4	124	3,540	281	143.7	5	87	363	798	68.9	89.6
70	–	–	–	–	–	5	139	444	273	18.6	90.8
80	–	–	–	–	–	5	160	518	302	26.7	91.8
90	–	–	–	–	–	5	179	576	396	36.9	92.8
100	–	–	–	–	–	4	200	637	486	51.5	93.6
110	–	–	–	–	–	4	219	729	527	63.0	93.9

(1) Number of successful problems out of 5.
(2) Maximum number of effective constraints during the course of the algorithm. (For further details, *see* Land and Powell [7]).
(3) Average value.

(mainly with the cutting planes algorithm). In some odd cases, the problems were badly conditioned (for example, the determinant of the inverse basis became too large, resulting in very weak cuts and in practically non convergence).

A simple examination of table confirms the superiority of method 2 over method 1: computation times are much smaller with method 2, less computer space is needed, larger problems can be solved and, on the whole, fewer failures occur.

The reason for this success lies in the relatively small number of variables contained in problems solved by method 2, and this, in spite of the fact that all distance matrices were complete (when G is not complete, the passage to G' actually increases the number of variables with method 1, thus making the comparison even more favourable to method 2). The last column of table shows that when $n \geqq 50$, at least 85% of the variables are eliminated by method 2. The percentage of eliminated variables grows up steadily as n increases.

A smaller number of variable helps reducing the number of simplex iterations; its major effect however, is on the amount of memory needed to store the contraints and on the average time per iteration.

Finally, the branch and bound algorithm appears to be more reliable than the cutting planes algorithms. This is especially true in the case of asymmetrical problems.

REFERENCES

1. M. BELLMORE and G. L. NEMHAUSER, *The Travelling Salesman Problem: a Survey*, Operations Research, Vol. 16, 1968, pp. 538-558.
2. N. CHRISTOFIDES, *The Travelling Salesman Problem*, published in "Combinatorial Optimisation" by CHRISTOFIDES *et al.*, Wiley, 1979, pp. 131-149.
3. G. B. DANTZIG, D. R. FULKERSON and S. M. JOHNSON, *Solution of a Larger Scale Traveling-Salesman Problem*, Operations Research, Vol. 2, 1954, pp. 393-419.
4. J. EDMONDS and E. JOHNSON, *Matching, Euler Tours and the Chinese Postman*, Mathematical Programming, Vol. 5, 1973, pp. 88-124.
5. L. EULER, *Commentationes Arithmeticae Collectae*, Saint-Petersbourg, 1766, pp. 337-338.
6. W. W. HARDGRAVE and G. L. NEMHAUSER, *On the Relation between the Travelling Salesman Problem and the Longest Path Problem*, Operations Research, Vol. 10, 1962, pp. 647-657.
7. A. H. LAND and S. POWELL, *Fortran Codes for Mathematical Programming*, Wiley, 1973.
8. P. MILIOTIS, *Integer Programming Approaches to the Travelling Salesman Problem*, Mathematical Programming, Vol. 10, 1976, pp. 367-378.
9. P. MILIOTIS, *Using Cutting Planes to Solve the Symmetric Travelling Salesman Problem*, Mathematical Programming, Vol. 15, 1978, pp. 117-188.
10. J. D. MURCHLAND, *A Fixed Matrix Method for all Shortest Distances in a Directed Graph and for the Inverse Problem*, Ph. D. Dissertation, Karlsruhe, 1970.