UNIV. BEOGRAD. PUBL. ELEKTROTEHN. FAK.
Ser. Mat. 5 (1994), 35–42.

THE TRAVELING SALESMAN PROBLEM ON A CHAINED DIGRAPH

Dragoš Cvetković, Vladimir Dimitrijević, Milan Milosavljević

The traveling salesman problem (TSP) on a chained digraph is studied. An algorithm based on the ideas of dynamic programming is presented. A previously known reduction to a kind of the shortest path problem is extended to a special multi-TSP.

1. INTRODUCTION

Suppose that a salesman, starting from his home city, is to visit exactly once each city on a given list of cities and then to return home. It is reasonable for him to select the order in which he visits the cities so that the total of the distances travelled in his tour is as small as possible. This problem is called the *traveling salesman problem* (TSP) (cf., e.g. monograph [**13**] and a recent expository article [**12**]).

Finding the traveling salesman's shortest route to pass n cities in such a way that each city is visited exactly once represents the traditional formulation of TSP. It is assumed that non-negative distances $c(i,j) = c_{ij}$ between the cities i, j $(1 \le i < j \le n)$ are given and also that traveling salesman starts his trip from an arbitrary city. If the traveling salesman does not return to the starting city then the minimal traversed route is called an *open route* or a *path.*

We mention some graph theoretical interpretations. For standard graph theoretical notions see, e.g., [**11**], [**3**].

Let G be a *weighted* digraph on n vertices. To each arc (i,j) of G a *length* (*weight* or *cost*) c_{ij} is assigned. In particular, in the TSP c_{ij} is the distance between the cities i and j. We define $c_{ij} = +\infty$ if the arc (i,j) does not exist in G. The

[0] 1991 Mathematics Subject Classification: 05C85

matrix $C = \| c_{ij} \|_1^n$ is called the *distance* (*weight* or *cost*) *matrix* of G. The *length* (*weight* or *cost*) $l(H)$ of a subgraph H of G is defined to be the sum of lengths of arcs of H. In particular, the length of a path is the sum of lengths of arcs from which it consists. A path (cycle) consisting from k arcs is called a *k-path* (*k-cycle*).

A cycle (path) passing through each vertex of G exactly once is called a *Hamiltonian cycle* (*path*). Of course, a Hamiltonian cycle is an n-cycle and a Hamiltonian path is an $(n-1)$-path. A salesman's tour (or *route*) is now a Hamiltonian cycle, or path. TSP can be reformulated in the following way:

In a weighted digraph (graph) find a Hamiltonian path (cycle) of a minimal length.

We recall general ideas for finding the best solution of the TSP and many other problems of combinatorial optimization. The first general algorithmic approach is *branch and bound*. The second one is the *polyhedral* approach. In the recent years, the polyhedral approach combined with branch and bound has proved to be an indispensable tool in computational solution of many hard large-scale real-world combinatorial problems.

Solutions to the TSP, which are not optimal but are "good" in some sense are called suboptimal solutions. "Good" suboptimal solutions are found by "clever" heuristics. There are many heuristics for the TSP described in the literature. One can also formulate the problem of finding the k-th best (k-best) solution(s) of a combinatorial optimization problem (see, e.g., [**8**]).

2. A CHAINED TSP

The TSP on a digraph with a chained structure is studied in this paper. Section 2.1 contains some basic notions while Section 2.2 provides an algorithm for solving TSP which is based on the dynamic programming (the DP-algorithm). In Section 2.3 the problem is reduced to the problem of finding a shortest path in an auxilary weighted digraph (the SP-algorithm). A part of these results have been announced in [**7**].

2.1. Preliminaries

Let G be a digraph whose adjacency matrix has the form

$$A = \left\| \begin{array}{ccccc} A_1 & B_1 & O & \dots & O \\ O & A_2 & B_2 & & O \\ \vdots & & & & \\ O & O & O & & B_{m-1} \\ O & O & O & & A_m \end{array} \right\|,$$

where $A_1, A_2, \dots, A_m$ are square blocks of order n. Subgraphs of G with adjacency matrices $A_1, A_2, \dots, A_m$ are denoted by $G_1, G_2, \dots, G_m$ respectively. Digraph G is

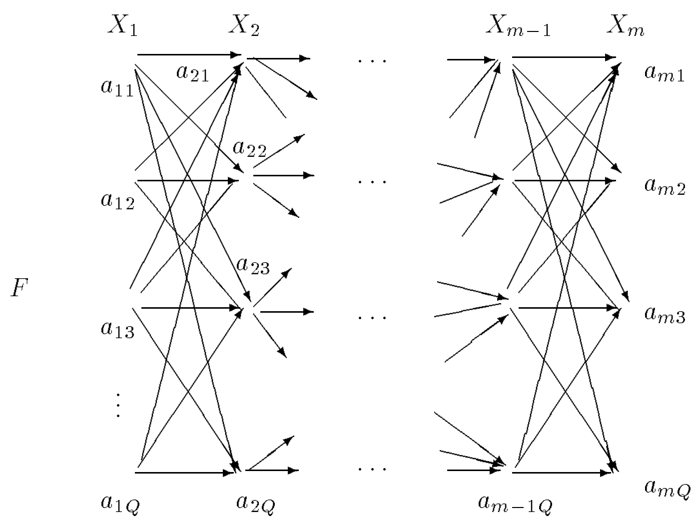

Fig. 1

called a *chained* digraph. If arcs of G carry some weights the corresponding TSP is called a *chained* TSP. The weight matrix C of G can be represented in the form

$$C = \left\| \begin{array}{ccccc} D_1 & E_1 & & & \\ & D_2 & E_2 & & \\ & & & & \\ & & & & E_{m-1} \\ & & & & D_m \end{array} \right\|,$$

where $D_1, D_2, \ldots, D_m$ are weight matrices of $G_1, G_2, \ldots, G_m$ and $E_1, E_2, \ldots, E_{m-1}$ contain weights of the interconnecting arcs.

A dynamic programming algorithm for the chained TSP is given in [**6**] (see Section 2.2).

We can form an auxiliary weighted digraph F (see Fig. 1) with weights on both vertices and arcs in the following way. The vertex set of F is the union of disjoints sets $X_1, X_2, \ldots, X_m$. Elements (vertices of F) from X_i are in a 1-1 correspondence with Hamiltonian paths of the digraph G_i $(i = 1, 2, \ldots, m)$. The weight of a vertex from X_i is the length of the corresponding Hamiltonian path in G_i. For any $i = 1, 2, \ldots, m-1$ we have arcs going from each vertex from X_i to each vertex of X_{i+1}. The weight of the arc between $x \in X_i$ and $y \in X_{i+1}$ is equal to the (p, q)-entry of E_i if p is the ending vertex of the Hamiltonian path in G_i which corresponds to x and q is the starting vertex of the Hamiltonian path in G_{i+1} which corresponds to y.

A chained TSP is reduced in [**14**], [**15**] to the problem of finding a shortest path in F which starts at one of the vertices from X_1 and terminates at one of the vertices of X_m, where the length of a path is the sum of weights of all vertices and arcs from the path. Some efficient procedures for finding such a path and k-best solutions are given in [**14**], [**15**] as well (see Section 2.3).

2.2. A dynamic programming algorithm

Following [**6**] we describe an algorithm for solving TSP on a digraph G having a chained structure. The algorithm is based on the ideas of dynamic programming and therefore we call it a DP-*algorithm*.

For a vertex x in G_i $(i = 1, 2, \ldots, m)$ let $d(x)$ be the length of a shortest path starting from x and passing through all vertices of $G_i, G_{i+1}, \ldots, G_m$. Such a path contains a Hamiltonian path in each of digraphs $G_i, G_{i+1}, \ldots, G_m$. Let $h(x)$ be the length of a shortest Hamiltonian path in G_i which starts at the vertex x. Finally, let $s(x)$ be the length of a shortest path starting at the x and passing through all the vertices of the digraphs $G_{i+1}, \ldots, G_m$.

We determine the quantity $d(x)$ first for vertices x from G_m, then for those from G_{m-1} and so on, back to the digraph G_1. The minimal value of $d(x)$ for vertices x from G_1 yields the solution of the TSP on G. Minimizing $d(x)$ also provides an optimal path as will be seen soon.

For vertices x from G_m we have $d(x) = h(x)$. The quantity $h(x)$ and the corresponding Hamiltonian path in G_m can be determined by any standard algorithm for the TSP.

Suppose we have determined $d(x)$ for vertices x from G_{i+1} and we proceed to determining $d(x)$ for vertices from G_i.

First we determine $s(x)$ for vertices x from G_i. Let $c(x, y)$ be the length of the arc joining vertex x with vertex y. Obviously, we have

$$s(x) = \min_{y \in G_{i+1}} (c(x, y) + d(y)).$$

Further, we form an auxiliary digraph H_i by extending digraph G_i with a new vertex z_i. From each of vertices x of G_i an arc goes toward z_i. The length of this arc is defined to be $c(x, z_i) = s(x)$. It is obvious now that $d(x)$ is equal to the length of a shortest Hamiltonian path in H_i starting at x and terminating in z_i.

2.3. Reduction to a kind of the shortest path problem

Following the paper [**14**] by M. Milosavljević and M. Obradović, we describe how the TSP on a chained digraph can be reduced to the problem of finding a shortest path in an auxiliary weighted digraph. The corresponding algorithm is called an SP-algorithm.

The length $l(H_G)$ of any (global) Hamiltonian path H_G in G is the sum of lengths $l(H_{G_i})$ of (partial) Hamiltonian paths H_{G_i} in G_i, $i = 1, 2, \ldots, m$, plus the sum of weights of the interconnecting arcs. Accordingly, the criterion function for the minimal (global) Hamiltonian path can be expressed in the form

$$(1) \qquad J(H_G) = \min l(H_G) = \min\{\sum_{i=1}^{m} l(H_{G_i}) + \sum_{i=1}^{m-1} e_{pq}^{(i)}\},$$

where $e_{pq}^{(i)}$ is an entry of E_i, with p the ending vertex of H_{G_i} and q the starting vertex of $H_{G_{i+1}}$.

As explained in Section 2.1, minimization of (1) leads to a convenient layered structure representation (Fig. 1), where this task is accomplished by selecting a minimal lenght path through the digraph F starting from X_1 and ending at X_m.

Let us introduce notation (see Fig. 1) as follows: vertex a_{ij} corresponds to the j-th Hamiltonian path in G_i, $i = 1, 2, \ldots, m$; $j = 1, \ldots, Q$, $Q = n!$, where the numeration of Hamiltonian paths in each X_i is arbitrary. The weight $c(a_{ij})$ of vertex $a_{ij} \in X_i$ is equal to the lenght of the j-th Hamiltonian path in G_i. The weight of the arc $(a_{il}, a_{i+1,k})$, denoted by $c(a_{il}, a_{i+1,k})$, is equal to the corresponding entry of matrix E_i as mentioned earlier. Without loss of generality we suppose $c(a_{ij}) > 0$, $i = 1, 2, \ldots, m$, $j = 1, \ldots, Q$, $c(a_{il}, a_{i+1,k}) > 0$, $i = 1, 2, \ldots, m-1$, $l, k = 1, \ldots, Q$. If the set of vertices of F is denoted by X and if U is the set of arcs, we shall write $F = F(X, U)$.

We consider the minimization of the criterion function

$$J(j_1, j_2, \ldots, j_m) = \sum_{i=1}^{m} \Big(c(a_{ij_i}) + c(a_{ij_i}, a_{i+1,j_{i+1}}) \Big),$$

(with $j_1, j_2, \ldots, j_m$ a path of length $m - 1$ through digraph F and $c(a_{mj_m}, a_{m+1,j_{m+1}}) = 0$), which is an equivalent of criterion (1).

In order to construct an efficient algorithm for minimization of criterion (1), we transform the weighted digraph F into another weighted digraph F' in the following way. Consider a set of mappings $j \to n(i,j)$, $i = 1, \ldots, m$; $j = 1, \ldots, Q$, actually m permutations on Q elements, under condition that

$$\begin{aligned} c(b_{ij}) &= c(a_{in(i,j)}),\ i = 1, \ldots, m,\ j = 1, \ldots, Q; \\ c(b_{ij}, b_{i+1,l}) &= c(a_{in(i,j)}, a_{i+1,n(i+1,l)}),\ i = 1, \ldots, m,\ l = 1, \ldots, Q; \\ c(b_{ij}) &\leq c(b_{ik}),\ j < k,\ i = 1, \ldots, m,\ j, k = 1, \ldots, Q; \end{aligned}$$

where $\{b_{ij}\}$ is the set of vertices of F'. In other words, we have rearranged each subset in such a way that $\{c(b_{ij})\}_{j=1}^{Q}$ forms a non-decreasing sequence, keeping length of all arcs the same as in F.

Of course, if $J(j_1^*, j_2^*, \ldots, j_m^*) = \min J(j_1, j_2, \ldots, j_m)$ on the digraph F, and $J^*(j_1^{**}, j_2^{**}, \ldots, j_m^{**}) = \min J(j_1, j_2, \ldots, j_m)$ on the digraph F', it is obvious that $J^*(j_1^{**}, j_2^{**}, \ldots, j_m^{**}) = J(j_1^*, j_2^*, \ldots, j_m^*)$, $j_i^{**} = n(i, j_i^*)$.

Observe that the shortest path could be expected to pass the reordered digraph F' through its "upper" part. Hence, the "lower" part of F' is irrelevant for a procedure of finding the shortest path and consequently need not be memorized and processed. The subgraph which is sufficient for processing is given by the following theorem from [**14**].

Theorem 1. *Let $j_1^*, j_2^*, \ldots, j_m^*$ be the shortest path in the subgraph $F'(X_S, U_S)$ of the reordered digraph $F'(X, U)$ defined by*

$$X_S = \{b_{ij} \mid i = 1, \ldots, m;\ j = 1, \ldots, j_i^{**}\},$$

where U_S is the corrresponding set of arcs. The indices j_i^{**}, $i = 1, \ldots, m$ are obtained as minimal indices j for which

$$c(b_{ij}) \geq c(b_{ij_i^*}) + c(b_{ij_i^*}, b_{i+1,j_{i+1}^*}) - \min_{l,k \in \{1,\ldots,Q\}} \{c(b_{il}, b_{i+1,k})\}.$$

Then, the problem of finding an $(m-1)$-path of minimal length in $F'(X,U)$ has the same solution as the same problem in subgraph $F'_S(X_S, U_S)$.

In [**14**] an algorithm for finding k-best solution of the TSP on a chained digraph (i.e. of the shortest path problem on F) is described.

3. A SPECIAL MULTI-TSP

In this section we shall consider the following multi-traveling salesman problem.

Given a weighted digraph G on $n = km$ (k, m integers) vertices, determine m disjoint $(k-1)$- paths with a minimal total length.

Note that this problem is quite different from the standard multi-TSP in which the number of cities in tours of salesmen are not prescribed. The standard problem is usually solved by a transformation which reduces it to the ordinary TSP [**2**]. This transformation cannot be applied successfully to our problem.

A problem similar to our problem is the so called *clover leaf problem* [**9**], [**1**]. In the clover leaf problem each salesman starts from a specified city and after visiting a fixed number of cities returns to the starting city. Note that our problem can be reduced to the clover leaf problem by introducing an additional city which is equidistant to all other cities.

Some branch and bound and dynamic programming algorithms for our problem have been described in reports [**4**], [**5**], [**6**]. Note that in [**10**] the problem has been formulated in terms of integer programming.

Let G be a chained digraph, consisting of digraphs $G_1, G_2, \ldots, G_k$. We shall consider m-TSP on digraph G in the presence of some aditional constraints [**6**].

Each of digraphs $G_1, G_2, \ldots, G_k$ has n vertices, where n is divisible by m. Feasible colections of m disjoint $(k\frac{n}{m}-1)$-paths have the additional property of inducing a collection of m disjoint $(n-1)$-paths on each of digraphs $G_1, G_2, \ldots, G_k$. A collection of m disjoint paths of equal lengths will be called an *m-collection.*

In order to get a feasible m-collection for G we start with an m-collection in G_1, continue it with an m-collection in G_2, and so on up to G_k. Terminal vertices (m in number) of paths from an m-collection in G_i are joined by arcs with m starting vertices of paths from an m-collection in G_{i+1}. There are $m!$ ways of joining the two m-collections by m arcs. In order to get an optimal join we have to solve an assignment problem of dimension m.

Now we can generalize the SP-algorithm described in Section 2.3, to the multi-TSP. We form the auxilary digraph F in such a way that vertices of F correspond now to m-collections (instead of Hamiltonian paths) from digraphs G_i. The weight of a vertex in F is equal to the length of the corresponding m-collection from a G_i.

The weight of an arc in F is the sum of the weights of the arcs joining m-collections corresponding to the starting and the terminal vertex of the arc, i.e. to the length of an optimal assignment solved when joining two m-collections by arcs. The algorithms of M. Milosavljević and M. Obradović can be now applied, without difficulties, for finding both an optimal solution and best suboptimal solutions.

REFERENCES

1. J. R. Araque: *Contributions to the polyhedral approach to vehicle routing.* PhD thesis, State University of New York at Stony Brook, Stony Brook, NY, 1989.
2. M. Bellmore and S. Hong: *Transformation of the multisalesmen problem to the standard travelling salesman problem.* J. Assoc. Comp. Mach. **21** (1974), 500–504.
3. J. A. Bondy and U. S. R. Murty: *Graph theory with applications.* Amer. Elsevier Publ. Co., Inc., New York, 1976.
4. D. Cvetković: *M traveling salesmen in a complete digraph with an asymmetric weight matrix* (Serbian), unpublished report. Faculty of Electrical Engineering, Belgrade, 1986, 1–30.
5. D. Cvetković: *M traveling salesmen in a complete digraph with an asymmetric band weight matrix* (Serbian), unpublished report. Faculty of Electrical Engineering, Belgrade, 1986.
6. D. Cvetković: *Connection between global and local solutions of the traveling salesmen problem* (Serbian), unpublished report. Faculty of Electrical Engineering, Belgrade, 1987, 1–11.
7. D. Cvetković, V. Dimitrijević, M. Milosavljević: *A survey of some non-standard traveling salesman problems.* YUJOR **2**, No 2 (1992), 163–185.
8. D. Cvetković, S. Simić: *Best suboptimal solutions in combinatorial optimization problems.* Proc. of the XVIII Conf. SYM-OP-IS '91, Herceg Novi 9–11 X.1991, ed. R. Stanojević, J. Vuleta, Herceg Novi 1991, 103–105.
9. G. B. Dantzig, J. H. Ramser: *The truck dispatching problem.* Management Science **6** (1960), 80–91.
10. V. Dimitrijević, M. Milosavljević: *M-traveling salesmen problem with constraint on the number of visited cities* (Serbian). Proc. of the XIII Conf. SYM-OP-IS '86, Herceg Novi 1986, Institut za ekonomiku industrije, Beograd 1986, 721–726.
11. F. Harary: *Graph theory.* Reading, Ma., 1969.
12. G. Laporte: *The traveling salesman problem: An overview of exact and approximate algorithms.* European Journal of Operational Research **59** (1992), 231–247.
13. E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, D. B. Shmoys, eds.: *The traveling salesman problem.* John Wiley and Sons, Chichester – New York – Brisbane – Toronto –Singapore, 1985.

14. M. Milosavljević, M. Obradović: *An algorithm for finding the shortest paths on a class of large dimensional graphs* (Serbian). Naučno-tehnički pregled, **35**, No 6 (1985), 3–6.
15. M. Milosavljević, M. Obradović: *An algorithm for finding the k shortest paths on a class of large dimensional graphs* (Serbian). Proc. of the X Conf., Sarajevo-Jahorina '86, 1986, 204–1 – 204–7.

(Received October 3, 1994)

Faculty of Electrical Engineering,
University of Belgrade,
P.O.Box 816, 11001 Belgrade,
Yugoslavia

Military Technical Academy,
Neznanog Junaka 38, 11000 Belgrade,
Yugoslavia

Institute for Applied Mathematics and Electronics,
Kneza Miloša 37, 11000 Belgrade,
Yugoslavia