STUDIA UNIV. "BABEŞ–BOLYAI", MATHEMATICA, Volume **LIII**, Number 1, March 2008

A CUTTING PLANE APPROACH TO SOLVE THE RAILWAY TRAVELING SALESMAN PROBLEM

PETRICĂ C. POP, CHRISTOS D. ZAROLIAGIS, AND GEORGIA HADJICHARALAMBOUS

Abstract. We consider the Railway Traveling Salesman Problem. We show that this problem can be reduced to a variant of the generalized traveling salesman problem, defined on an undirected graph $G = (V, E)$ with the nodes partitioned into clusters, which consists in finding a minimum cost cycle spanning a subset of nodes with the property that exactly two nodes are chosen from each cluster. We describe an exact exponential time algorithm for the problem, as well we present two mixed integer programming models of the problem. Based on one of this models proposed, we present an efficient solution procedure based on a cutting plane algorithm. Extensive computational results for instances taken from the railroad company of the Netherlands Nederlandse Spoorwegen and involving graphs with up to 2182 nodes and 38650 edges are reported.

1. **Introduction**

Assume that a salesman traveling with railways wishes to visit a certain number of cities. The salesman has a limited budget and the goal is to establish a schedule that allows him to visit all the cities and returning to the starting city at the total minimum cost, taking into consideration that when arrived at a station he/she has to spend some time for his affairs and then to continue his journey to another city with the first available train. We shall refer to this problem as the *Railway Traveling Salesman problem*, denoted (RTSP).

Received by the editors: 06.01.2007.

2000 *Mathematics Subject Classification.* 90C11, 90C27, 05C05.

Key words and phrases. generalized traveling salesman problem, integer programming, cutting planes.

The salesman is aware of the time schedule and is therefore able to construct the corresponding time-expanded graph $G = (V, E)$, see [1]. In that graph, every event (arrival or departure of a train) at a station corresponds to a node and edges between nodes represent either elementary connections between two events (i.e. served by a train that does not stop in-between), or waiting within a station. Nodes representing time events belonging to the same station (city) will be referred to as nodes within the same cluster, and the total number of clusters equals the total number of stations p that the salesman has to visit. There are two types of edges: inter-cluster edges (corresponding to elementary connections between the stations) and intra-cluster edges (corresponding to waiting in a station for some later connection). With this graph at hand the salesman can associate costs to its edges according to the cost measure he/she wants to minimize. Consequently, the RTSP reduces in finding a Hamiltonian tour H of the minimum cost in the subgraph of G induced by S, where $S \subseteq V$ such that S contains exactly two nodes from every cluster. This leads to a variant of the so-called *generalized traveling salesman problem* (GTSP).

The generalized traveling salesman problem, introduced by Laporte and Nobert [5] and by Noon and Bean [6] is defined on a complete undirected graph G whose nodes are partitioned into a number of subsets (clusters) and whose edges have a nonnegative cost. The GTSP asks for finding a minimum-cost Hamiltonian tour H in the subgraph of G induced by S, where $S \subseteq V$ such that S contains *at least* one node from each cluster.

A different version of the problem called E-GTSP arises when imposing the additional constraint that *exactly* one node from each cluster must be visited.

Both problems GTSP and E-GTSP are *NP*-hard, as they reduce to traveling salesman problem when each cluster consists of exactly one node.

The GTSP has several applications to location and telecommunication problems. More information on these problems and their applications can be found in Fischetti, Salazar and Toth [1, 2], Laporte, Asef-Vaziri and Sriskandarajah [3], Laporte, Mercure and Nobert [4]. It is worth to mention that Fischetti, Salazar and

Toth [2] solved the GMST problem to optimality for graphs with up to 442 nodes using a branch-and-cut algorithm.

In this paper, we introduce the (above mentioned) variant of the GTSP, called the 2-GTSP, which, given a graph G with non-negative edge costs, asks for finding a minimum cost Hamiltonian tour H of G spanning a subset of nodes that includes exactly two nodes from each cluster and exactly one edge from each cluster. Clearly, a solution to 2-GTSP is a solution to the railway traveling salesman problem.

The aim of this paper is to provide an exact algorithm for the 2-GTSP as well as two integer programming formulations of the problem and an efficient cutting plane algorithm.

2. **Definition and Complexity of the 2-GTSP**

Let $G = (V, E)$ be an n-node undirected graph whose edges are associated with non-negative costs. We will assume w.l.o.g. that G is a complete graph (if there is no edge between two nodes, we can add it with an infinite cost). Let $V_1, ..., V_p$ be a partition of V into p subsets called *clusters* (i.e. $V = V_1 \cup V_2 \cup ... \cup V_p$ and $V_l \cap V_k = \emptyset$ for all $l, k \in \{1, ..., p\}$). We denote the cost of an edge $e = \{i, j\} \in E$ by c_{ij} or by $c(i, j)$. Let $e = \{i, j\}$ be an edge with $i \in V_l$ and $j \in V_k$. If $l \neq k$ the e is called an *inter-cluster* edge; otherwise e is called an *intra-cluster* edge.

The *2-generalized traveling salesman problem* (2-GTSP) asks for finding a minimum-cost tour H spanning a subset of nodes such that H contains exactly two nodes from each cluster V_i, $i \in \{1, ..., p\}$. The problem involved two related decisions:

- choosing a node subset $S \subseteq V$, such that $|S \cap V_k| = 2$, for all $k = 1, ..., p$.
- finding a minimum cost Hamiltonian cycle in the subgraph of G induced by S.

We will call such a cycle a *2-Hamiltonian tour*. An example of a 2-Hamiltonian tour for a graph with the nodes partitioned into 6 clusters is presented in the next figure.

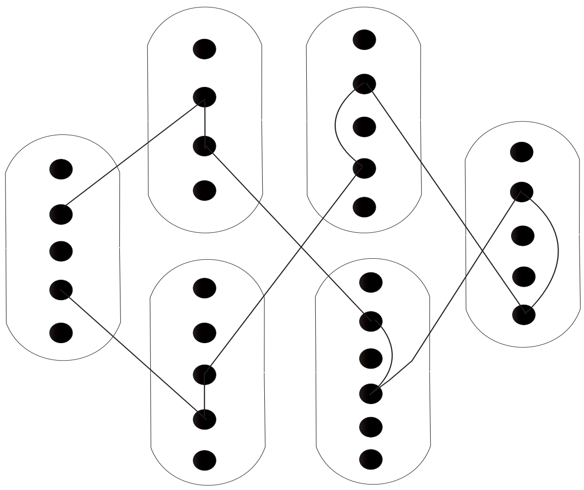

FIGURE 1. Example of a 2-Hamiltonian tour

As we already mentioned, both problems GTSP and E-GTSP are *NP*-hard, as they reduce to traveling salesman problem when each cluster consists of exactly one node. Consequently, the 2-GSTP is also an NP-hard problem.

3. **An Exact Algorithm for the 2-GTSP**

In this section, we present an algorithm that finds an exact solution to the 2-GTSP.

Given a sequence $(V_{k_1}, ..., V_{k_p})$ in which the clusters are visited, we want to find the best feasible 2-Hamiltonian tour H^* (w.r.t cost minimization), visiting the clusters according to the given sequence. This can be done in polynomial time, by solving $|V_{k_1}|$ shortest path problems as we will describe below.

We construct a layered network, denoted by LN, having $p+1$ layers corresponding to the clusters $V_{k_1}, ..., V_{k_p}$ and in addition we duplicate the cluster V_{k_1}. The layered network contains all the nodes of G plus some extra nodes v' for each $v \in V_{k_1}$. There is an arc (i, j) for each $i \in V_{k_l}$ and $j \in V_{k_{l+1}}$ $(l = 1, ..., p-1)$, having the cost c_{ij} and an arc (i, h), $i, h \in V_{k_l}$, $(l = 2, ..., p)$ having cost c_{ih}. Moreover, there is an arc (i, j') for each $i \in V_{k_p}$ and $j' \in V_{k_1}$ having cost $c_{ij'}$.

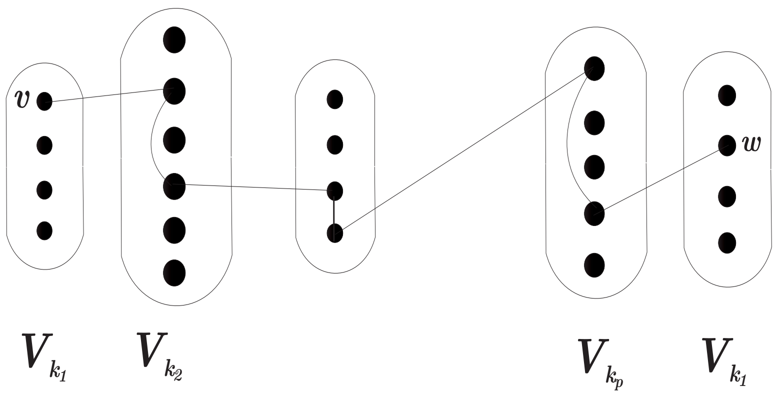

FIGURE 2. Example showing a 2-Hamiltonian tour in the constructed layered network LN

For any given $v \in V_{k_1}$, we consider paths from v to w', $w' \in V_{k_1}$, that visits exactly two nodes from each cluster $V_{k_2}, ..., V_{k_p}$, hence it gives a feasible 2-Hamiltonian tour.

Conversely, every 2-Hamiltonian tour visiting the clusters according to the sequence $(V_{k_1}, ..., V_{k_p})$ corresponds to a path in the layered network from a certain node $v \in V_{k_1}$ to $w' \in V_{k_1}$.

Therefore, it follows that the best (w.r.t cost minimization) 2-Hamiltonian tour H^* visiting the clusters in a given sequence can be found by determining all the shortest paths from each $v \in V_{k_1}$ to each $w' \in V_{k_1}$ with the property that visits exactly two nodes and one edge each from clusters $(V_{k_2}, ..., V_{k_p})$.

The overall time complexity is then $|V_{k_1}|O(m + n \log n)$, i.e. $O(nm + nlogn)$ in the worst case. We can reduce the time by choosing $|V_{k_1}|$ as the cluster with minimum cardinality.

Notice that the above procedure leads to an $O((p-1)!(nm + nlogn))$ time exact algorithm for the 2-GTSP, obtained by trying all the $(p-1)!$ possible cluster sequences. So, we have established the following result:

Theorem 1. *The above procedure provides an exact solution to the 2-GSTP in* $O((p-1)!(nm+nlogn))$ *time, where* n *is the number of nodes,* m *is the number of edges and* p *is the number of clusters in the input graph.*

Clearly, the algorithm presented, is an exponential time algorithm unless the number of clusters p is fixed.

4. Integer Programming Formulations of the 2-GTSP

In this section, we present two different integer programming formulations of the 2-GTSP.

In order to formulate the 2-GTSP as an integer program, we introduce the binary variables:

$$x_e = x_{ij} = \begin{cases} 1 & \text{if the edge } e = \{i,j\} \in E \\ & \text{is included in the selected subgraph} \\ 0 & \text{otherwise,} \end{cases}$$

$$z_i = \begin{cases} 1 & \text{if the node } i \text{ is included in the selected subgraph} \\ 0 & \text{otherwise.} \end{cases}$$

A feasible solution to the 2-GTSP can be seen as a cycle free subgraph with $2p-1$ edges connecting all the clusters such that exactly two nodes are selected from each cluster.

For $F \subseteq E$ and $S \subseteq V$, let $E(S) = \{e = \{i,j\} \in E \mid i,j \in S\}$, $x(F) = \sum_{e \in F} x_e$ and $z(S) = \sum_{i \in S} z_i$. Also, let $x(V_k, V_k) = \sum_{i,j \in V_k, i<j} x_{ij}$, where $k \in \{1, ..., p\}$.

The 2-GTSP can be formulated as the following 0-1 integer programming problem:

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
s.t. \quad & z(V_k) = 2, && \forall\, k \in \{1, ..., p\} && (1)\\
& x(\delta(i)) = 2z_i, && \forall\, i \in V \setminus V_1 && (2)\\
& x(E) = 2p - 1 && && (3)\\
& x(V_k, V_k) = 1, && \forall\, k \in \{2, ..., p\} && (4)\\
& x(E(S)) \leq 2r - 1, && \forall\, S = \cup_{i=1}^{r} V_{l_i}, 2 \leq r \leq p-1 && (5)\\
& x_e \in \{0, 1\}, && \forall\, e \in E && (6)\\
& z_i \in \{0, 1\}, && \forall\, i \in V. && (7)
\end{aligned}$$

where for $i \in V \setminus V_1$, the set, denoted by $\delta(i)$, is defined as

$$\delta(i) = \{e = \{i, j\} \in E \mid j \in V\}.$$

In the above formulation, constraint (1) guarantee that from every cluster we select exactly two nodes, constraints (2) require that the number of edges incident with a node i to be either 2 (if node i is visited) or 0 otherwise, constraint (3) guarantees that the selected subgraph has $2p-1$ edges, constraints (4) guarantee that from every cluster we select (except the starting cluster) we select one edge and finally constraints (5) eliminate all the cycles connecting at most $p-1$ clusters.

Replacing the subtour elimination constraints (5) by connectivity constraints, we result in the so-called *generalized cut-set formulation*:

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
s.t. \quad & (1), (3) - (5) \text{ and} \\
& x(\delta(S)) \geq 2(z_i + z_j - 1), \quad \forall\, S \subset V, \text{with } 2 \leq |S| \leq p-1 \\
& \qquad\qquad \text{and } \forall\, i \in S, j \in V \setminus S. && (8)
\end{aligned}$$

where for $S \subseteq V$, the *cut-set*, denoted by $\delta(S)$, is defined as

$$\delta(S) = \{e = \{i, j\} \in E \mid i \in S, j \notin S\}.$$

In the above formulation, constraints (8) are the connectivity constraints saying that each cut separating two visited nodes (i and j) must be crossed at least twice.

In the addition to the constraints that appear in the previous formulations, we consider also the following constraints specific to the railway traveling salesman problem:

$$t_d z_d - t_a z_a \geq t_k, \;\; \forall\, a, d \in V_k, \; 2 \leq k \leq p. \tag{9}$$

The above constraints are saying that the difference between the departure and arrival times has to be at least a specified time period t_h (depending on the city), this means that the traveling salesman has to stay in each city for some time to finish his business. If the difference is too small, the salesman may fail to solve his business, on the other hand, if the difference is too large, the waiting time at the station will be inconvenient.

The disadvantage of the described integer programming formulations is their exponential number of constraints (we have to choose subsets of V, constraints (5) and (8)). These constraints can be omitted and then can be generated as needed by a *separation algorithm*: one can start without constraints (5), solve the corresponding relaxation, then generate subtour inequalities that are violated by the current solution. The separation algorithm for subtour constraints is based on network flow techniques, for further details see [2].

5. **Solution procedure and computational results**

We used the following *cutting plane algorithm* in order to solve the 2-GTSP:

1. Let the integer programming (IP) formulation consists of the constraints (1)-(4),(6),(7) and (9).

2. Solve the IP and assume that the optimal solution consists of r subtours: $S_1, ..., S_r$.
3. If $r = 1$, then STOP; the solution is optimal to the 2-GTSP. Otherwise, add to the IP formulation the corresponding constraints that eliminate the cycles $S_1, ..., S_r$ and go to Step 2.

The algorithm was written in C and for each instance we have created the corresponding integer program, which we solved it with CPLEX 6.5.

Test data for our algorithm are real networks from the Dutch railroad company *Nederlandse Spoorwegen.*

The first three data sets contains the Intercity train connections among the larger cities in the Netherlands, stopping only at the main train stations, and thus are considered faster than the normal trains. These trains operate at least every half an hour. The second real-world data set, contains the schedules of the Intercity trains and regional trains. The regional trains connect the cities in only one region, including some main stations, while trains stop at each intermediate station between two main ones.

Some characteristics of the graphs that were used for the real-world data sets and the computational results obtained using the cutting plane algorithm are displayed in the next table:

Table: Computational results for solving the RTSP

Pb. name	*No. stations*	*No. nodes*	*No. edges*	*LB/OPT*	*Sol. time*
NS1 (IC)	5	394	4240	100	14.08 s
NS2 (IC)	7	674	9754	100	64.57 s
NS3 (IC)	9	926	16271	100	206.52 s
NS4 (IC+IR)	12	1470	23850	100	39.30 min
NS5 (IC+IR)	12	1586	27383	100	72.28 min
NS6 (IC+IR)	15	1722	28200	100	1.05 h
NS7 (IC+IR)	15	1946	34450	100	5.45 h
NS8 (IC+IR)	18	2182	38650	100	4.52 h

The first four columns in the table give the name of the problem and the size of the problem: the number of stations, the number of nodes and the number of edges. The next two columns describe the cutting plane procedure and contain: the lower bounds obtained as a percentage of the optimal value of the RTSP (LB/OPT) and the computational times (CPU) for solving the RTSP to optimality.

6. **Conclusions**

We considered the Railway Traveling Salesman Problem (RTSP), which consists in finding a minimum cost tour for a salesman traveling with railways and wishing to visit a certain number of cities. We showed that the RTSP can be reduced to a variant of the Generalized Traveling Salesman problem.

Based on one of the integer programming formulations that we proposed, we present an efficient solution procedure based on a cutting plane algorithm. Computational results for real networks from the Dutch railroad company *Nederlandse Spoorwegen* and involving graphs with up to 2182 nodes and 38650 edges are reported.

References

[1] Fischetti, M., Salazar, J.J., Toth, P., *The symmetric generalized traveling salesman polytope*, Networks, **26**(1995), 113-123.

[2] Fischetti, M., Salazar, J.J., Toth, P., *A branch-and-cut algorithm for the symmetric generalized traveling salesman problem*, Operations Research, **45**(1997), 378-394.

[3] Laporte, G., Asef-Vaziri, A., Sriskandarajah, C., *Some applications of the generalized traveling salesman problem*, J. Oper. Res. Soc., **47**(1996), 1461-1467.

[4] Laporte, G., Mercure, H., Nobert, Y., *Generalized Traveling Salesman Problem through n sets of nodes: the asymmetrical cases*, Discrete Applied Mathematics, **18**(1987), 185-197.

[5] Laporte, G., Nobert, Y., *Generalized Traveling Salesman through n sets of nodes: an integer programming approach*, INFOR, **21**(1983), 61-75.

[6] Noon, C.E., Bean, J.C., *A Lagrangian based approach for the asymmetric generalized traveling salesman problem*, Operations Research, **39**(1991), 623-632.

[7] Schultz, F., Wagner, D., Weihe, K., *Dijkstra's Algorithm On-line: An Empirical Case Study from the Public Railroad Transport*, ACM Journal of Experimental Algorithmics, **5**(12)(2000).

Department of Mathematics and Computer Science
North University of Baia Mare
Baia Mare, Romania
E-mail address: pop_petrica@yahoo.com

Department of Computer Engineering and Informatics
University of Patras
Patras, Greece
E-mail address: zaro@ceid.upatras.gr

Department of Computer Engineering and Informatics
University of Patras
Patras, Greece
E-mail address: hadjicha@ceid.upatras.gr