Discussiones Mathematicae
Graph Theory 20 (2000) 231–242

PERSISTENCY IN THE TRAVELING SALESMAN PROBLEM ON HALIN GRAPHS

Vladimír Lacko

Department of Geometry and Algebra
P.J. Šafárik University
Jesenná 5, 041 54 Košice, Slovakia

e-mail: lackov@kosice.upjs.sk

Abstract

For the Traveling Salesman Problem (TSP) on Halin graphs with three types of cost functions: sum, bottleneck and balanced and with arbitrary real edge costs we compute in polynomial time the persistency partition $E_{All}, E_{Some}, E_{None}$ of the edge set E, where:
$E_{All} = \{e \in E, e$ belongs to all optimum solutions$\}$,
$E_{None} = \{e \in E, e$ does not belong to any optimum solution$\}$ and
$E_{Some} = \{e \in E, e$ belongs to some but not to all optimum solutions$\}$.

Keywords: persistency, traveling salesman problem, Halin graph, polynomial algorithm.

2000 Mathematics Subject Classification: 05C45, 68Q25.

1 Introduction

It is a common case that the minimum solution of an optimization problem is not unique. If this occurs there is often a need for some additional constraints that further reduce the number of optimum solutions of the given problem. But before applying those new constraints and solving the new (and possibly harder) problem, we try to describe in some fashion the current set of optimum solutions. Since listing all of them may not be efficient (e.g., in case of exponential number of optimum solutions), the following approach can be utilized: characterize all the decision variables according to their behaviour with respect to optimum solutions. Hence for problems where feasible sets are sets of edges of a graph we define the folowing persistency partition of the set of edges E:

$E_{All} = \{e \in E, e$ belongs to all optimum solutions$\}$ (1-persistent edges),
$E_{None} = \{e \in E, e$ does not belong to any optimum solution$\}$ (0-persistent edges) and
$E_{Some} = \{e \in E, e$ belongs to some but not to all optimum solutions$\}$ (w-persistent edges).

In this paper we study the persistency for Traveling Salesman Problem (TSP). The Traveling Salesman Problem is NP-complete in general case. Thus, we direct our attention to a special type of graphs — so called Halin graphs, where TSP can be solved in polynomial time.

A *Halin graph* $H = T \cup C$ is obtained by embedding a tree T without nodes of degree 2 in the plane and adding a cycle C (outer cycle) joining the leaves of T in such a way, that the resulting graph is planar. Given a Halin graph $H = T \cup C$ and arbitrary real edge costs the (sum) TSP calls for finding a Hamilton circuit O having minimum sum of edge weights.

A polynomial algorithm for finding one optimum solution of TSP on Halin graphs was given in [4].

The notion of persistency was introduced in [5] for the maximum cardinality problem in bipartite graphs. Further results concerning persistency in the assignment and transportation problem, bases of matroids, matroid intersection problem, flows in networks, matchings in general and bipartite graphs and spanning forests in graphs can be found in [1], [2], [3] and [6]. The persistency in matroid product problem was treated in [7].

2 TSP on Halin Graphs

Let $H = T \cup C$ be a Halin graph and $c(e)$ for $e \in E(H)$ be the real costs of its edges. We denote the cost of a set $E_0 \subseteq E(H)$ by $c(E_0) = \sum_{e \in E_0} c(e)$. If T is a star then H is called a *wheel*. Otherwise T has at least two nonleaves. Let w be a nonleaf which is adjacent to only one other nonleaf of T. We denote the set of leaves of T adjacent to w by $T(w)$ and call the subgraph of H induced by $\{w\} \cup T(w)$ a *fan* with *centre* w. (See Figure 1.)

The following three lemmas can be found in [4]:

Lemma 21. *A Halin graph which is not a wheel has at least two fans.*

Let $G = (V, E)$ be a graph and let $S \subseteq V$. We denote by $G \times S$ the graph obtained from G by shrinking S to form a new pseudonode $\tilde{S}$. That is, the nodes of $G \times S$ are $V - S$ plus node $\tilde{S}$; the edges are all edges of G which

do not have both ends in S and those incident with one node in S have now that end taken to be the pseudonode $\tilde{S}$. If G' is a subgraph of G, we abbreviate $G \times V(G')$ by $G \times G'$, otherwise if G' is not a subgraph of G we write $G \times G'$ instead of $G \times (V(G') \cap V(G))$.

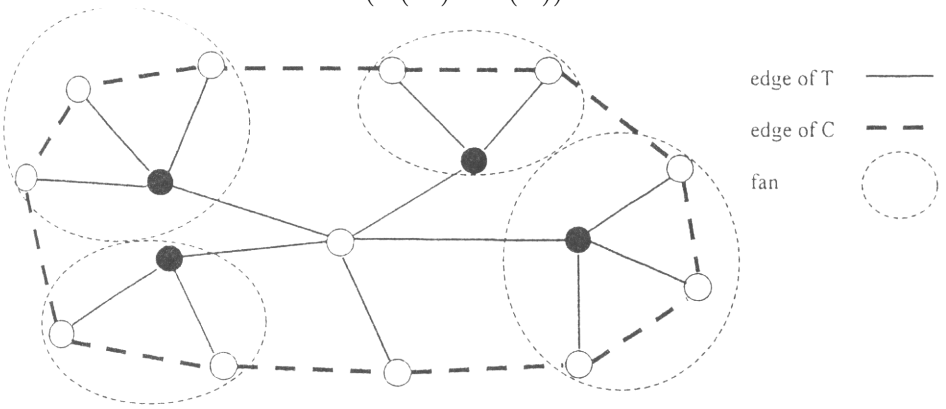

Figure 1

Lemma 22. *If F is a fan in a Halin graph H, then $H \times F$ is a Halin graph.*

A simple but useful property is the following:

Lemma 23. *Let $G = (V, E)$ be a connected graph. Every Hamilton cycle contains exactly two edges of every 3-edge cutset of G.*

Now we describe the polynomial algorithm for the TSP on a Halin graph $H = T \cup C$ which was proposed in [4]. The algorithm creates a sequence of fans $\{F_i\}_{i=1}^{q-1}$ and a sequence of Halin graphs $\{H_i\}_{i=1}^{q}$ obtained by shrinking corresponding fans F_i, i.e. $H_i = H_{i-1} \times F_i$. After shrinking, costs are updated to get sequence of cost functions $\{c^i\}_{i=1}^{q}$. Initially $H_1 = H$ and $c^1 = c$.

Suppose, that the algorithm has already created a sequence $\{H_1, H_2, \ldots, H_i\}$ of Halin graphs. For Halin graph H_i there are two possibilities:

Case 1. H_i is a wheel. The number of different Hamilton cycles in H_i is equal to the degree of the star T. So it is easy to obtain an optimum solution.

Case 2. H_i is not a wheel. By Lemma 21, H_i has at least two fans; let F_i be any fan of H_i. Let v be the centre of F_i and let $u_1, u_2 \ldots, u_r$ for $r \geq 2$ be the nodes of F_i that belong to C (in that order). There are exactly

three edges j, k and l of H_i joining nodes of F_i to nodes of $V - V(F_i)$. (See Figure 2) In what follows we will call the edges j and l (incident with u_1 and u_r respectively) the side edges and the edge k (incident with v) the middle edge.

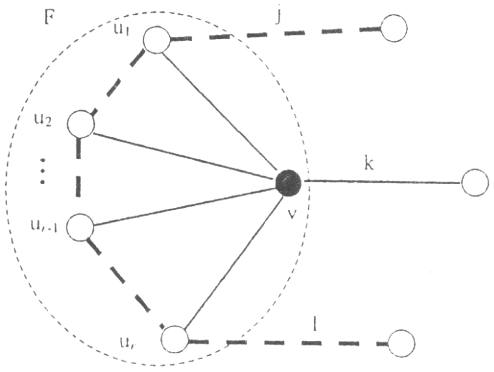

Figure 2

By Lemma 23 any optimum Hamilton cycle O of H_i uses exactly two edges of $\{j, k, l\}$. If O uses exactly one side edge, then there is only one possibility to traverse the nodes of F_i. If O uses both side edges, then it must traverse the nodes of F_i in the order $u_1, u_2, \ldots, u_i, v, u_{i+1}, \ldots, u_r$ for some $i \in I \subseteq \{1, 2, \ldots, r-1\}$. Let $\Delta_i = c(v, u_i) + c(v, u_{i+1}) - c(u_i, u_{i+1})$. Then index set I consists of those indices, for which Δ_i attains its maximum in F_i.

Let K be the sum of the costs of the edges in F_i belonging to the outer cycle C of H. Then

- if Hamilton cycle O uses j and k, the edges in F_i contribute $\tilde{C}_{jk} = K + c(v, u_r)$ to its cost
- if Hamilton cycle O uses k and l, the edges in F_i contribute $\tilde{C}_{kl} = K + c(v, u_1)$ to its cost
- if Hamilton cycle O uses j and l, the edges in F_i contribute $\tilde{C}_{jl} = K + c(u_i, v) + c(v, u_{i+1}) - c(u_i, u_{i+1})$ for $i \in I$ to its cost

The following lemma summarizes the results stated in [4]:

Lemma 24. *Let H_i be a Halin graph with real costs of edges denoted by c^i and F_i its fan. Let $H_{i+1} = H_i \times F_i$. If O is an optimum Hamilton cycle in (H_i, c^i) then $O' = O \times F_i$ is an optimum Hamilton cycle in Halin graph*

H_{i+1} with cost function c^{i+1} defined by

$$(1)\quad c^{i+1}(e) = \begin{cases} c^i(e) & \text{for } e \in E(H_{i+1}) - \{j,k,l\}, \\ c^i(j) + \frac{1}{2}\left(\tilde{C}_{jl} + \tilde{C}_{jk} - \tilde{C}_{kl}\right) & \text{if } e = j, \\ c^i(k) + \frac{1}{2}\left(\tilde{C}_{kl} + \tilde{C}_{jk} - \tilde{C}_{jl}\right) & \text{if } e = k, \\ c^i(l) + \frac{1}{2}\left(\tilde{C}_{jl} + \tilde{C}_{kl} - \tilde{C}_{jk}\right) & \text{if } e = l. \end{cases}$$

If O' is an optimum Hamilton cycle in Halin graph (H_{i+1}, c^{i+1}), then there exists an optimum Hamilton cycle O in (H_i, c^i) such that $O' = O \times F_i$.

The algorithm consists of recursively applying Case 2 until the graph is reduced to a wheel H_q, when Case 1 is applied. Then the shrunk fans are expanded in the reverse order and the Hamilton cycle is extended through them. The total time needed for this algorithm is $O(|V|)$.

3 Persistency Partition

We show in this section how to compute the persistency partition for the sum TSP on a given Halin graph H.

As the first step we recursively apply Case 2 of the previously described algorithm of Cornuéjols et al. until the initial Halin graph H is reduced to a wheel H_q.

The idea is now first to compute the persistency partition of the wheel H_q. Since H_q is very simple, this is an easy task. In the next step the pseudonode of H_q created by restriction $H_{q-1} \times F_{q-1}$ is replaced with fan F_{q-1} and the persistency partition for all the edges of the restored fan F_{q-1} is computed. In this way we obtain the persistency partition for Halin graph H_{q-1}. We repeat this step until we get persistency partition of the original Halin graph H.

Suppose that in the p-th step of computing persistency partition we have a Halin graph H_i $(i = q - p + 1)$ which is not equal to the original Halin graph H (otherwise the proofs is complete) with cost function c^i. Then there exists a Halin graph H_{i-1} with fan F_{i-1} such that $H_i = H_{i-1} \times F_{i-1}$. Lemma 24 describes the connection between optimum TSP solutions in H_i and H_{i-1}. Here again we can replace the corresponding pseudonode in H_i with fan F_{i-1} and we need to compute the persistency partition only for edges of fan F_{i-1}, since from Lemma 24 persistency of edges not in fan F_{i-1} is the same in H_{i-1} as in H_i.

It is left to describe how to compute the persistency partition of edges of the restored fan F_{i-1} in Halin graph H_{i-1} with cost function c^{i-1}. The pseudonodes have always degree 3 (all nodes lying on the outer cycle of a Halin graph have this degree). Suppose, that every edge $e \in E(H_{i-1})$ is labelled with one of $0, 1$ or w meaning that it is 0–, 1– or w-persistent in TSP on H_i. For the given pseudonode v there are three possibilities (see Figure 3) how edges i, j and k can be labelled:

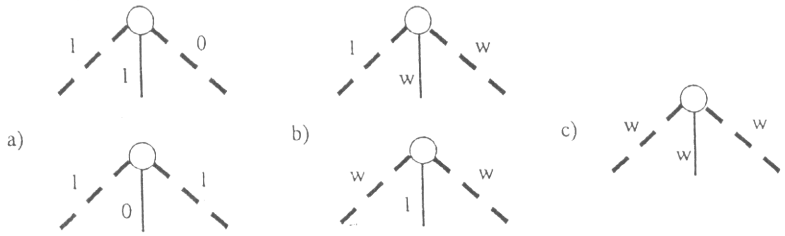

Figure 3

1. $(1,1,0),(1,0,1),(0,1,1)$ configurations = there is only one possibility for any optimum Hamilton cycle to traverse node v.
2. $(1,w,w),(w,1,w),(w,w,1)$ configurations = there are two possibilities, one edge is used by all optimum Hamilton cycles.
3. (w,w,w) configuration = for each of the three possible pairs of edges $\{j,k,l\}$ there is an optimum Hamilton cycle using that pair.

The remaining configurations of types $(1,1,1)$, $(0,0,0)$, $(w,1,1)$, $(1,0,0)$, $(w,0,0)$, $(0,w,w)$ and $(0,1,w)$ are clearly impossible.

We will describe in detail how to deal with only one type of configuration (namely 2). The other two types can be treated similarly.

In type 2 configuration we distinguish two cases (see Figure 3b):

Case a. The middle edge k is 1-persistent (configuration $(w,1,w)$).
In this case, we have a unique $j-k$ path through fan F_{i-1} (using nodes $u_1,u_2,\ldots,u_r,v$) and also unique $k-l$ path (using nodes $v,u_1,u_2,\ldots,u_r$) which are parts of some optimum Hamilton cycle in H_{i-1}. Therefore we may assign persistency to all edges of fan F_{i-1} (except for $\{j,k,l\}$ for which we have already computed persistency) as follows:

edges (u_i,u_{i+1}) for $i=1,2,\ldots,r-1$:	1-persistent
edges $(v,u_1),(v,u_r)$:	w-persistent
edges $(v,u_i), 1<i<r$:	0-persistent

Case b. One of the side edges j or l is 1-persistent (configuration $(1, w, w)$ or $(w, w, 1)$).
Suppose that edge j is 1-persistent. This case is a little more complicated since here we must deal with $j - l$ paths through fan F_{i-1}, which may not be unique. Suppose, that the optimum $j - l$ paths through F_{i-1} use nodes $\{(u_1, \ldots, u_p, v, u_{p+1}, \ldots, u_r);\ p \in I \subseteq \{1, \ldots, r-1\}\}$ for some index set I. The set of optimum $j-l$ paths can be computed along with fan shrinking at the first step of the algorithm. Here for each $p \in I$ the relation between costs in H_{i-1} and H_i is $c^{i-1}(\{u_1, \ldots, u_p, v, u_{p+1}, \ldots, u_r\}) = c^i(j) + c^i(l)$. Now we may assign the persistency in the following way:

$$\text{edges } (u_p, u_{p+1}): \begin{cases} w\text{-persistent}, & \text{if } p \in I, \\ 1\text{-persistent}, & \text{otherwise.} \end{cases}$$

$$\text{edges } (v, u_p), p < r: \begin{cases} w\text{-persistent}, & \text{if } p \in I, \\ 0\text{-persistent}, & \text{otherwise.} \end{cases}$$

$$\text{edges } (v, u_r): \begin{cases} 1\text{-persistent}, & \text{if } I = \{r-1\}, \\ w\text{-persistent}, & \text{otherwise.} \end{cases}$$

For the first step of computing persistency partition we need $O(|V|)$ time (see Section 2). Note that each time a fan is restored in a Halin graph, the number of nonleaf nodes of the tree is increased by one. Therefore the total number of times a fan will be restored is $O(|V(T)|)$.

If a fan F_i contains t_i nodes, then the time for computing persistency of its edges is clearly at most $O(t_i)$. Restoring a fan F_i increases the number of nodes of the graph by t_i and since $\sum t_i = |V|$, the total complexity of persistency computation is $O(|V|)$ which is also the total time bound for this algorithm.

As an easy observation from the previous results we have that in the unweighted version (all edge weights are equal) of the TSP on Halin graphs all edges are w-persistent: it is a known fact that Halin graphs are 1-hamiltonian (i.e., after removal of any edge they remain hamiltonian) and by looking closer to the fan reduction and restoration process we can see that every edge can belong to some Hamilton cycle (compare [4]).

4 Bottleneck Version of TSP

In the previous section we explored the TSP on Halin graphs with the cost function $c(O) = \sum_{e \in O} c(e)$. The bottleneck version of this problem uses the

cost function in the form:

$$c_a(O) = \max_{e \in O} c(e), \text{ where } O \text{ is a Hamilton cycle,}$$

and finds a Hamilton cycle which has minimum value of c_a.

A trivial aproach to the bottleneck case consists of two steps.

Step 1. Find the optimum value c_a^{Opt} of cost function $c_a(O)$.

For the given cost $c(\tilde{e})$ of edge $\tilde{e} \in E$ we can decide whether there exists a Hamilton cycle O such that $c_a(O) \leq c(\tilde{e})$ easily: we assign new costs c' to edges of Halin graph H:

$$c'(e) := \begin{cases} 0, & \text{if } c(e) \leq c(\tilde{e}), \\ 1, & \text{otherwise.} \end{cases}$$

Then we use the algorithm for finding sum-optimum solution of TSP on Halin graph (see [4]) which in $O(|V|)$ time gives us an optimum solution O^* with the following property:

$c'(O^*) = 0$ if and only if H contains a Hamilton cycle O with $c_a(O) \leq c(\tilde{e})$.

Now the optimum value c_a^{Opt} can be found using binary search for costs $c(\tilde{e})$.

Step 2. For each edge $e \in E$ with weight $c(e) \leq c_a^{Opt}$ decide, whether there exist optimum Hamilton cycles O_1 and O_2 having $e \in E(O_1)$ and $e \notin E(O_2)$.

Again we assign new costs to edges:

$$c''(e) := \begin{cases} 0, & e = \tilde{e}, \\ 1, & e \neq \tilde{e} \text{ and } c(e) \leq c^{Opt}, \\ 2, & \text{otherwise,} \end{cases}$$

and

$$c'''(e) := \begin{cases} 0, & e \neq \tilde{e} \text{ and } c(e) \leq c^{Opt}, \\ 1, & \text{otherwise.} \end{cases}$$

Then we solve two TSPs with cost functions c'' and c''' and find sum-optimum solutions O'' and O''' respectively. Now $c''(O'') = |V| - 1$ is equivalent to the existence of O_1 and $c'''(O''') = 0$ is equivalent to the existence of O_2. The persistency partition is straightforward from that point.

In Step 1 $\log |V|$ iterations of binary search are enough for finding the optimum value of the cost function. It sums up to the complexity

$O(|V|\log|V|)$. In Step 2 we have to test at most $O(|V|)$ edges with $O(|V|)$ time for one test. It gives the total complexity $O(|V|^2)$ for the trivial approach.

However, we show here how it is possible to decide about the persistency in bottleneck case in $O(|V|)$ time using a similar algorithm as for the sum case. The key step of determining persistency in the sum case was fan shrinking. For each shrunk fan F a new costs were assigned to the side edges j, l and to the middle edge k to assure that sums $c(j)+c(k), c(j)+c(l), c(k)+c(l)$ will be the same as the costs of minimum $j-k, j-l, k-l$ paths respectively. In the bottleneck case we are not always able to make a similar assignment of new costs. (See e.g., Figure 4: newly assigned costs of edges j, k and l have to satisfy $\max\{c(j), c(k)\} = \max\{c(j), c(l)\} = 2$ and $\max\{c(k), c(l)\} = 3$, which is impossible.)

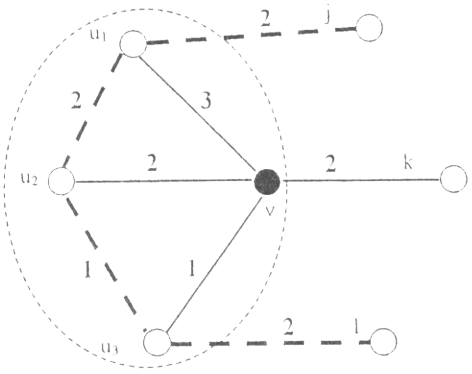

Figure 4

Therefore we must utilize a different approach: we assign to each pseudonode u_i a list of numbers c^i_{jk}, c^i_{kl} and c^i_{jl} denoting costs of minimum $j-k, k-l$ and $j-l$ paths through the corresponding shrunk fan. With the costs assigned that way we can continue with fan shrinking in a similar fashion. See for example the situation depicted in Figure 5. Here are two pseudonodes u_1, u_3 with assigned lists of costs $c^1_{jk}, c^1_{kl}, c^1_{jl}$ and $c^3_{jk}, c^3_{kl}, c^3_{jl}$. There are three possible paths between side edges j and l with costs $\max\{0, c^1_{jk}, 0, b, 0, c^3_{jl}, 0, d\}, \max\{0, c^1_{jl}, 0, b, 0, c^3_{kl}, 0, d\}$ and $\max\{0, c^1_{jl}, 0, 0, c^3_{jk}, 0, c, d\}$.

In general, the cost of a path passing through vertices $u_1, \ldots, u_i, v, u_{i+1}, \ldots, u_r$ is equal to the maximum number of the following union of sets:

$$\{c(u_q, u_{q+1}); q \neq i\} \cup \{c^q_{jl}; u_q \text{ is pseudonode}, q \neq i, q \neq i+1\} \cup$$
$$\{c^i_{jk}; u_i \text{ is pseudonode}\} \cup \{c^{i+1}_{kl}; u_{i+1} \text{ is pseudonode}\} \cup$$
$$\{c(u_i, v), c(v, u_{i+1})\}.$$

The case of paths between the middle edge k and one of the side edges is even easier because of their uniqueness.

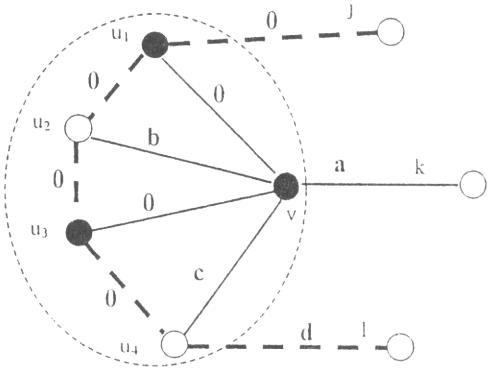

Figure 5

By such a modification of persistency algorithm described in the previous chapter we are able to solve the persistency partition of the bottleneck TSP in $O(|V|)$ time, which is clearly an optimum algorithm.

5 Balanced Version of TSP

The balanced version of this problem deals with the cost function in the form:

$$c_b(O) = \max_{e \in O} c(e) - \min_{e \in O} c(e), \text{ where } O \text{ is a Hamilton cycle,}$$

and finds a Hamilton cycle which has minimum value of c_b.

In the balanced case a trivial approach would consist of testing all possible cost intervals $\langle c_i, c_j \rangle$ whether there is a feasible solution of a TSP consisting only of edges with costs from the tested interval. The complexity of feasibility test is $O(|V|)$ as it was shown in bottleneck case which gives an overall complexity $O(|V|^3)$.

However, using the results of the bottleneck case it is possible to decide about persistency in the balanced case in $O(|V|^2)$ time. We employ the following approach:

Step 1. For each different cost $c(h)$ look for the optimum solution O_h of TSP with the bottleneck cost function c_a and with lower bound $c(h)$, i.e., where only edges with weight greater or equal to $c(h)$ are allowed. Denote the set of such optimum solutions Sol_{Opt}. Now it is easy to see that the optimum value of balanced cost function is $c^{opt} = \min_{O_h \in Sol_{Opt}} c_b(O_h)$.

Step 2. Let $Cost_{Opt} = \{c(h); O_h \in Sol_{Opt}$ and $c_b(O_h) = c^{opt}\}$ be the set of those lower bounds $c(h)$, for which the corresponding Hamilton cycle O_h is optimal in balanced case. There can be more than one optimum corresponding to given lower bound $c(h)$, but optimum sets for different lower bounds are clearly disjoint.

Step 3. Compute persistency partitions $E^h_{All}, E^h_{Some}, E^h_{None}$ for each lower bound $c(h) \in Cost_{Opt}$ and then merge them to compute the final persistency partition according to the rule:

$$E_{All} = \bigcap_{c(h) \in Cost_{Opt}} E^h_{All};\ E_{None} = \bigcap_{c(h) \in Cost_{Opt}} E^h_{None};\ E_{Some} = E - E_{All} - E_{None}$$

It is left to show how to find an optimum solution of TSP with cost function c_a and with lower bound $c(h)$. We assign a large constant K to all edges $e \in E$ with weights less than $c(h)$ (K should be greater than $\max_{e \in E} c(e)$). We then find the optimum solution O^* of TSP with cost function c_a using new costs. If $c_a(O^*) = K$, then there is no optimum of TSP with lower bound $c(h)$ and we continue with testing next lower bound, otherwise O^* is the optimum we are looking for.

The complexity of the above approach is $O(|V|^2)$: there are at most $O(|V|)$ lower bounds $c(h)$ and for each one we need to employ one search for optimum solution, which takes $O(|V|)$ time. Then we need to compute persistency partition for each $c(h) \in Cost_{Opt}$ which takes at most $|Cost_{Opt}|.O(|V|) = O(|V|^2)$ time. The final merging of persistency partitions takes only $O(|V|)$ time.

References

[1] K. Cechlárová, *Persistency in the assignment and transportation problems*, Math. Methods of Operations Research **47** (1998) 234–254.

[2] K. Cechlárová and V. Lacko, *Persistency in some combinatorial optimization problems*, in: Proc. Mathematical Methods in Economy 99 (Jindřichúv Hradec, 1999) 53–60.

[3] K. Cechlárová and V. Lacko, *Persistency in combinatorial optimization problems on matroids*, to appear in Discrete Applied Math.

[4] G. Cornuéjols, D. Naddef and W.R. Pulleyblank, *Halin graphs and the Traveling salesman problem*, Mathematical Programming **26** (1983) 287–294.

[5] M.C. Costa, *Persistency in maximum cardinality bipartite matchings*, Operations Research Letters **15** (1994) 143–149.

[6] V. Lacko, *Persistency in optimization problems on graphs and matroids*, Master Thesis, UPJŠ Košice, 1998.

[7] V. Lacko, *Persistency in the matroid product problem*, in: Proc. CEEPUS Modern Applied Math. Workshop (AGH Kraków, 1999), 47–51.

Received 11 January 2000
Revised 29 March 2000