UNIVERSITE PARIS DAUPHINE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision

CNRS UMR 7024

CAHIER DU LAMSADE

233

March 2005

Reoptimization of minimum and maximum travelling salesman's tours

G. Ausiello, B. Escoffier, J. Monnot V. Th. Paschos

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Reoptimization of minimum and maximum traveling salesman's tours

Giorgio Ausiello[1], Bruno Escoffier[2], Jérôme Monnot[2], and Vangelis Th. Paschos[2]

[1] Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Roma, Italy. Email: ausiello@dis.uniroma1.it.
[2] Lamsade, CNRS and Université Paris Dauphine, France. Email: {escoffier,monnot,paschos}@lamsade.dauphine.fr.

Abstract. In this paper, reoptimization versions of the traveling salesman problem (TSP) are addressed. Assume that an optimum solution of an instance is given and the goal is to determine if one can maintain a good solution when the instance is subject to minor modifications. We study the case where nodes are inserted in, or deleted from, the graph. When inserting a node, we show that the reoptimization problem for MinTSP is approximable within ratio 4/3 if the distance matrix is metric. We show that, dealing with metric MaxTSP, a simple heuristic is asymptotically optimum when a constant number of nodes are inserted. In the general case, we propose a 4/5-approximation algorithm for the reoptimization version of MaxTSP.

1 Introduction

The traveling salesman problem (TSP) is one of the most interesting and paradigmatic optimization problems. In both minimization and maximization versions, TSP has been widely studied and a large bibliography is available (see, for example, the books [8, 12, 13]). As it is well known, both versions of TSP are NP-hard but although in the case of MaxTSP the problem is approximable within constant ratio for all kinds of graphs [4, 10], in the case of MinTSP approximation algorithms are known only for the metric case [5], i.e., when the graph distances satisfy the triangle inequality.

In this paper, we deal with *reoptimization* issue. We consider the case where instances of a given optimization problem are subject to minor modifications. The problem we are interested in consists, given an optimum solution on the initial instance, of trying to maintain efficiently a good solution when the instance is slightly modified. This issue has already been studied for other optimization problems such as scheduling problems (see [18, 2], or [3] for practical applications) and classical polynomial problems where the goal is to recompute the optimum solution as fast as possible ([6, 11]). It has been recently considered for MinTSP in [1]. The modifications for TSP consists in adding a new node to the initial graph (we have a new city to visit), or removing one node from this graph (a city is dropped from the tour).

More precisely, we suppose that an n node graph G is given and an optimum solution of MinTSP for G has already been computed. In the problem-version we deal with, denoted MinTSP+ in the sequel, G is transformed into a graph G' by adding a new node v_{n+1} together with all edges connecting v_{n+1} to any node of G. How can we reuse the known optimum solution of MinTSP for G in order to compute a good approximate solution for G'? An analogous problem denoted MinTSP- consists of reoptimizing MinTSP when a node v in G is deleted together with all edges incident to it. In [1], Archetti, Bertazzi and Speranza show that both MinTSP+ and MinTSP- are NP-hard. Moreover they prove that if the simple *best insertion* rule is used for updating the previously known optimum tour, a (tight) 3/2 approximate tour for MinTSP+ in metric case can be obtained whereas in the general case, they propose some instances leading to the claim that best insertion rule is not a constant approximation; the same (tight) 3/2 approximation ratio is obtained for MinTSP- in the metric case. In their paper, the authors of [1] were mainly motivated by the situation where a short amount of time is available for the reoptimization. However, another interesting question is to know if the knowledge of an optimum solution for a part of the input graph leads to strictly better approximation ratios for the whole of the graph than those achieved in the classical approximation framework.

In this paper we provide new insights for the reoptimization of MinTSP (for metric graphs), both in the case of a single update and in the case where k new nodes are inserted (denoted MinTSP+k). For MinTSP+ in metric case we show that by combining the best insertion heuristics with Christofides' algorithm the result of [1] can be outperformed, by achieving approximation ratio 4/3. Moreover, it is possible to show that, for any k, MinTSP+k can be approximated asymptotically better than 3/2, although, for large values of k, the approximation ratio converges to Christofides' bound. On the other hand, dealing with the general case, we prove that MinTSP+ is not constant approximable. We also study reoptimization of MaxTSP, by considering the problems MaxTSP+ and MaxTSP+k for the first time, both in the metric and in the general case (note that these problems are obviously **NP**-hard). In particular we show that, in the metric case, for any k, the best insertion rule is asymptotically optimum; in fact, for any k, MaxTSP+k can be approximated with ratio $\left(1 - \frac{O(k)}{\sqrt{n}}\right)$. In the general case we can exhibit a 4/5-approximation algorithm, an improvement over the approximation ratio 61/81, achieved in [4] (under the classical approximation paradigm). The paper is organized as follows. In the next section, we provide basic definitions and notation. In Section 3, we address the reoptimization of MinTSP under single and multiple node insertions. Next, in Section 4, we consider the reoptimization of MaxTSP, first under single node insertion (both in the metric and in the general case) and subsequently under multiple insertions (in the metric case). Finally, in Section 5, some results concerning MinTSP- and MaxTSP- are provided. Concluding remarks are contained in Section 6.

All the missing and sketched proofs are given in the appendix.

2 Preliminaries

In this section we provide the formal definitions of the problems addressed in the paper, namely Min and MaxTSP+k, Min and MaxTSP-k. Then, we introduce three heuristics, Best Insertion, Longest Insertion, and Nearest Insertion, classically studied in the literature (see for instance [16, 8]) because they give rise to fast algorithms to solve TSP, and particularly suitable when dealing with reoptimization.

Definition 1 (MinTSP+k, MaxTSP+k). *We are given an instance* (I_{n+k}, T_n^*) *where* $I_{n+k} = (K_{n+k}, d)$*,* K_{n+k} *is a complete graph on* $n+k$ *nodes* $\{v_1, \cdots, v_{n+k}\}$*, with nonnegative weights* d *on the edges, and* T_n^* *is an optimum solution of MinTSP (resp. MaxTSP) on* $I_n = (K_n, d)$*, sub-instance of* I_{n+k} *induced by the nodes* $\{v_1, \cdots, v_n\}$*.*

Question : find a shortest (resp. longest) tour for the whole instance I_{n+k}*.*

Definition 2 (MinTSP-k, MaxTSP-k). *We are given an instance* (I_{n+k}, T_{n+k}^*) *where* $I_{n+k} = (K_{n+k}, d)$*,* K_{n+k} *is a complete graph on* $n+k$ *nodes* $\{v_1, \cdots, v_{n+k}\}$*, with nonnegative weights* d *on the edges, and* T_{n+k}^* *is an optimum solution of MinTSP (resp. MaxTSP) on* I_{n+k}*.*

Question : find an shortest (resp. longest) tour on $I_n = (K_n, d)$*, sub-instance of* I_{n+k} *induced by the nodes* $\{v_1, \cdots, v_n\}$*.*

For the case $k = 1$, we simply denote the problems MinTSP+, MaxTSP+, Min TSP- and MaxTSP-.

For TSP, a particular rapid way to get a tour is to iteratively insert nodes according to given rules, as the following classical ones.

Definition 3 (Nearest, Longest and Best Insertion rule). *Given a tour* T *on a graph* $G = [V, E]$*, and a node* $v \notin V$*, we insert* v *in the sequence of nodes of* T *as follows:*

- *Nearest Insertion: we find a node* v^* *minimizing* $d(u, v)$ *for* $u \in V$*, and insert* v *before or after* v^* *(choosing the best solution) in the tour;*
- *Longest Insertion: we find a node* v^* *maximizing* $d(u, v)$ *for* $u \in V$*, and insert* v *before or after* v^* *(choosing the best solution) in the tour;*
- *Best Insertion: we find an edge* $(u^*, v^*) \in T$ *optimizing* $(d(v, u) + d(v, w) - d(u, w))$ *for* $(u, v) \in T$*, and insert* v *between* u^* *and* v^**.*

Concerning polynomial approximation of MinTSP in the metric case, it is shown in [16] that the behavior of Nearest and Best Insertions are quite different since the algorithms based on these two rules are a 2 and a $O(\log n)$ -approximation respectively.

Finally, when nodes are deleted, the most natural way to get a solution from a tour on the initial instance consists in taking the shortcut.

Definition 4 (Deletion). *Given a tour T on a graph $G = [V, E]$, and a node $v \in V$, deletion consists in building a tour by deleting v in T (removing (u, v) and (v, w) from T and adding (u, w)).*

3 Reoptimizing minimum TSP under node insertions

In this section, we study the reoptimization problems where one node is inserted (MinTSP+) and several nodes are inserted (MinTSP+k). We show that we can improve the result of [1] proving that, in the metric case, MinTSP+ is approximable within ratio 4/3.

On the contrary, if the distance is not assumed to be metric, then the knowledge of an optimum solution in the initial instance is not useful at all in order to find an approximate solution of the final instance since MinTSP+ (and consequently MinTSP+k) is not constant approximable (unless **P**$\neq$**NP**).

Finally, we generalize the result in the metric case by showing that when k nodes are inserted we get a $(3/2 - 1/(4k + 2))$-approximation algorithm.

3.1 One node insertion

When dealing with metric instances of MinTSP+, it is proved in [1] that Best Insertion gives a 3/2-approximate solution. Actually, we can show that Nearest Insertion also provides this bound. Of course, running Christofides' algorithm on the final instance gives directly also a 3/2-approximate solution. Here we show that a simple combination of Nearest (or Best) Insertion and Christofides' algorithm leads to a better approximation ratio.

Theorem 1. *In the metric case, MinTSP+ is approximable within ratio* $4/3$.

Proof. Consider an optimum solution T^*_{n+1} on the whole instance I_{n+1}, and the solution T^*_n given to us on the sub-instance I_n.

Let v_i and v_j be the 2 neighbors of v_{n+1} in T^*_{n+1}, and let T_1 be the tour obtained from T^*_n with the Nearest Insertion rule.

Using the triangle inequality, we easily get $d(T_1) \leq d(T^*_{n+1}) + 2d(v^*_{n+1}, v_{n+1})$ where we recall that $d(v^*_{n+1}, v_{n+1}) = \min\{d(v_i, v_{n+1}) : i = 1, \cdots, n\}$. Thus

$$d(T_1) \leq d(T^*_{n+1}) + 2\max\{d(v_i, v_{n+1}), d(v_j, v_{n+1})\} \tag{1}$$

Now, consider the algorithm of Christofides ([5]) applied on I_{n+1}. This gives a tour T_2 of length at most $1/2d(T^*_{n+1}) + MST(I_{n+1})$, where $MST(I_{n+1})$ is the value of a minimum spanning tree on I_{n+1}. Note that $MST(I_{n+1}) \leq d(T^*_{n+1}) - \max(d(v_i, v_{n+1}), d(v_j, v_{n+1}))$. Hence :

$$d(T_2) \leq \frac{3}{2}d(T^*_{n+1}) - \max(d(v_i, v_{n+1}), d(v_j, v_{n+1})) \tag{2}$$

We take the best solution between T_1 and T_2. A combination of equations (1) and (2) with coefficients 1 and 2 gives the expected result.

Obviously, if we apply Best Insertion instead of Nearest Insertion, the same result holds. Note that the running time of this algorithm is dominated by the one of Christofides' algorithm. □

In [1], it is shown that if the distance is not assumed to be metric, then Best Insertion is not constant approximate for MinTSP+. We strengthen this result by proving that this holds for any polynomial algorithm.

To do this, we need an intermediate result. Given a graph $G = [V, E]$ where $a, b, s, t \in V$, and an hamiltonian path of G from a to b, we consider the problem of determining if there exists an hamiltonian path from s to t. Using a slight modification of the result of [14], we can show that this problem, denoted by $\text{SHP}_{a,b,s,t}$ in the sequel, is **NP**-complete (see appendix).

Lemma 1. *$SHP_{a,b,s,t}$ is **NP**-complete (even in bipartite graphs of maximum degree 5).*

This lemma leads to the following inapproximability result.

Theorem 2. *In the general case, MinTSP+ is not $2^{p(n)}$-approximable, if **P**≠**NP**, for any polynomial p.*

Proof. We apply the general method described in [17]. Let $\rho > 1$. We start from an instance of $\text{SHP}_{a,b,s,t}$, *i.e.* a graph $G_n = [V, E]$ with n nodes, four nodes a, b, s, t, and an hamiltonian path P from a to b. We construct an instance (I_{n+1}, T^*_n) in the following way:

- If $(v_i, v_j) \in E$, then $d(v_i, v_j) = 1$.
- $d(a, b) = 1$ and $d(v_{n+1}, s) = d(v_{n+1}, t) = 1$.
- All the other edges have a weight $\rho(n + 1) + 1$.

It is clear that $T^*_n = P \cup \{(a, b)\}$ is an optimum solution of $I_n = (K_n, d)$ with cost $d(T^*_n) = n$. Thus, (I_{n+1}, T^*_n) is an instance of MinTSP+. Let T^*_{n+1} be an optimum solution of (K_{n+1}, d). Remark that any ρ-approximate solution allows us to decide if $d(T^*_{n+1}) = n + 1$. However $d(T^*_{n+1}) = n + 1$ iff there is a hamiltonian path from s to t in G_n. Setting $\rho = 2^{p(n)}$, we obtain the claimed result. □

3.2 *k* node insertions

When k nodes are inserted, we can generalize the result of Theorem 1 in the following way.

Theorem 3. *In the metric case, MinTSP+k is approximable within ratio $3/2 - 1/(4k + 2)$*

Proof. Consider the given optimum solution T^*_n. We apply Nearest Insertion with a priority rule. In a first step, we sort the vertices to be inserted (and relabel them) in such a way that for all $p > n$, there exists v_j, $j < p$ such that

$d(v_p, v_j) = \min\{d(v_i, v_l) : i \geq p, l < p\}$. Note that $d(v_p, v_j) \leq d_{max}(T^*_{n+k})$, where $d_{max}(T^*_{n+k})$ is a maximal weighted edge in T^*_{n+k}.

Then we insert the k vertices using Nearest Insertion.

For the analysis, note that when inserting vertex v_p, we increase the distance by $\Delta_p \leq 2d(v_p, v_j) \leq 2d_{max}(T^*_{n+k})$. We finally get an approximate solution T_1 such that

$$d(T_1) \leq d(T^*_n) + 2kd_{max}(T^*_{n+k}) \leq d(T^*_{n+k}) + 2kd_{max}(T^*_{n+k}) \tag{3}$$

Christofides' algorithm gives a solution T_2 such that

$$d(T_2) \leq \frac{3}{2}d(T^*_{n+k}) - d_{max}(T^*_{n+k}) \tag{4}$$

We take the best solution between T_1 and T_2. A combination of equations (3) and (4) with coefficients 1 and $2k$ gives $d(T) \leq \left(\frac{3}{2} - \frac{1}{4k+2}\right) d(T^*_{n+k})$.

Note that the computation time of T_1 is $O(k(n + k))$, hence the global complexity is dominated by running Christofides' algorithm. □

4 Reoptimizing maximum TSP under node insertions

In this section, we consider the reoptimization of the maximization version of TSP. In the metric case, Best Insertion is a very good strategy since it is asymptotically optimum. Note that the usual MaxTSP problem in the metric case does not admit a PTAS (using [15]) and that the best algorithms for it are asymptotically 17/20 (deterministic, [4]) and 7/8 (randomized, [10]).

If the distance is not assumed to be metric, the situation is a bit more complicated. Longest and Best Insertion are only a 1/2-approximation. This situation is quite disappointing since we can easily prove that iterating Longest Insertion (from the empty graph) with a priority rule is already a 1/2-approximation for MaxTSP; however, we can get a polynomial algorithm achieving a ratio of 4/5. This shows that the knowledge of an optimum solution on the initial instance is useful since the best algorithm for the usual MaxTSP achieves an approximation ratio of 61/81 ([4]).

Finally, in section 4.2, we generalize the result in the metric case showing that if we insert a constant number on nodes, then iterating Best Insertion is also an asymptotically optimum strategy.

Note that the **NP**-hardness of all these problems is obvious since otherwise, starting from the empty graph, we could solve polynomially MaxTSP.

4.1 One node insertion

The central result of this section is the asymptotical optimality of Best Insertion. It is interesting to note that the behavior of Best and Longest Insertion are quite different for MaxTSP+ since Longest Insertion is only a 2/3-approximation, even asymptotically (see appendix).

Proposition 1. *For MaxTSP+, in the metric case, Longest Insertion gives a 2/3-approximation, and this bound is tight (even if the graph has an arbitrary large number of nodes).*

Theorem 4. *In the metric case, Best Insertion is asymptotically optimum. More precisely, if the graph has n nodes, then Best Insertion is $(1 - O(1/\sqrt{n}))$-approximate.*

Proof (Sketch).(see appendix for a complete proof) Let T_n^* be an optimum solution on the initial instance I_n, T_{n+1}^* an optimum solution on the final instance I_{n+1}, and T the solution obtained by applying Best Insertion on T_n^*. Let $K = \sqrt{n}$ and $1 \leq k \leq K$.

Consider the following subsequence of nodes $(a_k, \cdots, a_1, v_{n+1}, b_1, \cdots, b_k)$ in T_{n+1}^*. Let J_k be the sub-instance of I_{n+1} induced by all the nodes but v_{n+1}, $a_1, a_2, \cdots, a_{k-1}$ and $b_1, b_2, \cdots, b_{k-1}$ (in particular J_1 is (K_n, d), the initial graph). We have :

$$d(T_{n+1}^*) \leq d(v_{n+1}, a_1) + d(v_{n+1}, b_1) + \sum_{i=1}^{k-1} d(a_i, a_{i+1}) + \sum_{i=1}^{k-1} d(b_i, b_{i+1}) + opt(J_k)$$

where $opt(J_k)$ is the value of an optimum solution on J_k. Indeed, there is an hamiltonian path in T_{n+1}^* between a_k and b_k, the value of which is at most $opt(J_k)$.

Let $d_m^k(v)$ be the medium distance between a node v and the nodes in J_k, i.e., $d_m^k(v) = \frac{1}{|J_k|} \sum_{v_i \in J_k} d(v, v_i)$. Using the triangle inequality, we get that for any pair (u, v) of nodes (and for any k), $d(u, v) \leq d_m^k(u) + d_m^k(v)$. Hence we get an upper bound on $d(T_{n+1}^*)$:

$$d(T_{n+1}^*) \leq 2 \left(d_m^k(n+1) + \sum_{i=1}^{k-1} d_m^k(a_i) + \sum_{i=1}^{k-1} d_m^k(b_i) \right) + d_m^k(a_k) + d_m^k(b_k) + opt(J_k) \tag{5}$$

Now, our goal is to lower bound first $d(T_n^*)$ and then $d(T)$ in order to get the following inequality :

$$d(T) \geq \left(1 - \frac{O(k)}{n}\right) (d(T_{n+1}^*) - d_m^k(a_k) - d_m^k(b_k)) \tag{6}$$

To achieve this, first consider an optimum solution $T^*(J_k)$ (of value $opt(J_k)$) of J_k. Considering a particular subsequence $(v_1, \cdots, v_{2k-1})$ of $T^*(J_k)$, we insert the $2(k-1)$ nodes $a_1, a_2, \cdots, a_{k-1}$ and $b_1, b_2, \cdots, b_{k-1}$ in $T^*(J_k)$ in order to get the sequence $(v_1, a_1, \cdots, a_{k-1}, v_k, b_1, \cdots, b_{k-1}, v_{2k-1})$. Considering each node of J_k as v_1, we get with these insertions $n - 2(k-1)$ tours on I_n. After a careful counting of the edges appearing in these tours, one can show that:

$$d(T_n^*) \geq 2 \left(\sum_{i=1}^{k-1} d_m^k(a_i) + \sum_{i=1}^{k-1} d_m^k(b_i) \right) + \left(1 - \frac{O(k)}{n}\right) opt(J_k) \tag{7}$$

Now, we relate $d(T)$ and $d(T_n^*)$. Consider each of the n possible insertions of v_{n+1} in T_n^*. Since each edge of T_n^* is removed exactly once, we get that $nd(T) \geq (n-1)d(T_n^*) + 2\sum_{i=1}^n d(v_{n+1}, v_i)$. Using $\sum_{i=1}^n d(v_{n+1}, v_i) \geq \sum_{v \in J_k} d(v_{n+1}, v) = (n - 2(k-1))d_m^k(n+1)$, we get:

$$d(T) \geq \left(1 - \frac{1}{n}\right) d(T_n^*) + 2\left(1 - \frac{2(k-1)}{n}\right) d_m^k(n+1) \tag{8}$$

From inequalities (5), (7) and (8), we can derive (6).

Inequality (6) is valid for any k. Let us write it for $k = 1, \cdots, K$, and consider the two following cases :

1. If, for some k, $d_m^k(a_k) + d_m^k(b_k) \leq \frac{1}{K} d(T_{n+1}^*)$, then we get

$$d(T) \geq \left(1 - \frac{O(k)}{n}\right)\left(1 - \frac{1}{K}\right) d(T_{n+1}^*)$$

   Since $k \leq K = \sqrt{n}$, we get $d(T) \geq \left(1 - O\left(\frac{1}{\sqrt{n}}\right)\right) d(T_{n+1}^*)$.
2. In the other case, for any k, $d_m^k(a_k) + d_m^k(b_k) \geq \frac{1}{K} d(T_{n+1}^*)$. However, this is impossible. Indeed, by making the sum, we get $\sum_{k=1}^K d_m^k(a_k) + d_m^k(b_k) \geq d(T_{n+1}^*)$. But (details are omitted here), one can show that this would lead to $d(T_n^*) \geq 2\left(1 - \frac{O(K)}{n}\right) d(T_{n+1}^*)$, which is impossible for n large enough.

□

From Theorem 4, we get the following corollary.

Corollary 1. *MaxTSP+ admits a PTAS in the metric case.*

Proof. Let $\varepsilon > 0$. To get a $(1-\varepsilon)$-approximation algorithm, we just have to apply Best Insertion on graphs with roughly $n \geq O(1/\varepsilon^2)$ nodes, and to solve optimally the other instances. □

Unfortunately, if the triangle inequality is not assumed, Best Insertion has a much worse behavior (see appendix).

Proposition 2. *For MaxTSP+, in the general case, Best Insertion and Longest Insertion give a 1/2-approximation, and this bound is tight (even if the graph has an arbitrary large number of nodes).*

However, we can use a more sophisticated algorithm to get a better approximation ratio.

Theorem 5. *MaxTSP+ is asymptotically approximable within ratio 4/5.*

Proof (Sketch). Assume n even; thus T_n^* is the sum of two perfect matchings M_1 and M_2 (if n is odd we can add the remaining edge to each matching. Details are omitted). Suppose $d(M_1) \geq d(M_2)$. We get:

$$d(M_1) \geq \frac{1}{2} d(T_n^*) \tag{9}$$

Let v_i and v_j be the neighbors of v_{n+1} in T_{n+1}^*. Consider $M^* = M_1 \cup \{(v_i, v_{n+1}), (v_{n+1}, v_j)\}$. Obviously, M^* can be found in polynomial time by guessing nodes v_i and v_j. Wlog., we can assume that M^* does not contain any cycle (otherwise, $(v_i, v_j) \in T_n^*$ and thus Best Insertion gives an optimum tour).

Now, consider $\mathcal{C} = \{C_1, \cdots, C_p\}$, a 2-matching (i.e., a partition of $\{v_1, \cdots, v_{n+1}\}$ into node disjoint cycles) of maximum weight among the 2-matchings satisfying (i) $\{(v_i, v_{n+1}), (v_{n+1}, v_j)\} \subset C_1$ and (ii) $|C_1| \geq 6$. Such a 2-matching can be found in polynomial time by testing all the possible subsequences of nodes $(v_{i''}, v_{i'}, v_i, v_{n+1}, v_j, v_{j'})$ (and thanks to the polynomiality of finding a maximum weight 2-matching, [9]). Obviously, we deduce:

$$d(\mathcal{C}) \geq d(T_{n+1}^*) \tag{10}$$

Applying the method of Serdyukov [19], we can iteratively for $i = 1, \cdots, p$, delete an edge $e_i \in C_i$, and add this edge to M^* in such a way that M^* does not contain any cycle. Note that in this method we can chose in C_1 a deleted edge not in M^* that does not create a cycle in P_1 (thanks to the length of C_1).

At the end, $P_1 = \cup_{i=1}^{p} (C_i \setminus \{e_i\})$ and $P_2 = M^* \cup_{i=1}^{p} \{e_i\}$ are two collection of node disjoint paths. Finally, we build two tours T_1 and T_2 by adding some edges to P_1 and P_2 respectively. Taking the best tour, and using inequalities (9) and (10), we get a tour T_3 with:

$$d(T_3) \geq \frac{3}{4} d(T_{n+1}^*) + \frac{1}{4}(d(v_i, v_{n+1}) + d(v_{n+1}, v_j)) \tag{11}$$

On the other hand, the Best Insertion gives a tour T_4 verifying:

$$d(T_4) \geq \frac{n-1}{n} d(T_n^*) \geq \frac{n-1}{n} d(T_{n+1}^*) - \frac{n-1}{n}(d(v_i, v_{n+1}) + d(v_{n+1}, v_j)) \tag{12}$$

Adding inequality (11) with coefficient $(n-1)/n$ and inequality (12) with coefficient $1/4$ we obtain a tour satisfying $d(T) \geq \frac{4n-4}{5n-4} d(T_{n+1}^*)$. □

4.2 *k* node insertions

When several nodes are inserted, we can iteratively use the Best Insertion rule to obtain an asymptotically optimum solution. This result is based on the following lemma.

Lemma 2. *If T_n is a ρ-approximation on the initial instance on n nodes G_n, then Best Insertion applied on T_n gives a $\rho\left(1 - \frac{O(1)}{\sqrt{n}}\right)$-approximate solution (in the metric case) on the instance G_{n+1} on $n+1$ nodes.*

Proof (Sketch). This is an easy generalization of the proof of theorem 4. Note that equation (5) and (7) still hold. Then, by taking into account that T_n is a ρ-approximation, we get, instead of equation (6):

$$d(T) \geq \rho \left(1 - \frac{O(k)}{n}\right)(d(T^*_{n+1}) - d^k_m(a_k) - d^k_m(b_k)) \tag{13}$$

The end of the proof is analogous, up to the factor ρ. □

Theorem 6. *Iterated Best Insertion is a* $\left(1 - \frac{O(k)}{\sqrt{n}}\right)$*-approximation algorithm for MaxTSP+k in the metric case.*

Proof. Using proposition 2, we get, after k steps, a solution T_k such that:

$$d(T_k) \geq \left(1 - \frac{O(1)}{\sqrt{n}}\right)^k d(T^*_{n+k}) \geq \left(1 - \frac{O(k)}{\sqrt{n}}\right) d(T^*_{n+k})$$

□

Using a similar proof as in corollary 1, we easily get the following result.

Corollary 2. *For any constant k (and even for any $k = o(\sqrt{n})$), MaxTSP+k admits a PTAS in the metric case.*

5 Node deletions

Now, we give a few results concerning the reoptimization problems when nodes are deleted from the initial graph. Recall that in [1] it is shown that MinTSP- is **NP**-hard, even if distances are only 1 and 2, and that deletion is a tight 3/2-approximation in the metric case. Here, we show that MinTSP- is very hard to approximate if the triangle inequality doesn't not hold.

Dealing with MaxTSP-, we show that the problem is **NP**-hard, and that deletion is a tight 1/2-approximation algorithm (general and metric cases).

Proposition 3. *In the general case, MinTSP- is not $2^{p(n)}$-approximable, if* ***P≠NP****, for any polynomial p.*

Proof. The proof is a direct adaptation of the one of [1] showing that this problem is **NP**-hard. We consider the following problem, shown to be NP-complete in [14]: given a graph $G = [V, E]$ and an hamiltonian path P between two nodes a and b in G, determine if there's an hamiltonian cycle in G.

Given such an instance, we construct an instance on MinTSP-. The node set of the graph K_{n+1} is $V \cup \{v_{n+1}\}$, and the distances are:

- $d(v_i, v_j) = 1$ if $(v_i, v_j) \in E$;
- $d(v_{n+1}, a) = d(v_{n+1}, b) = 1$;
- Other distances are $\rho n + 1$.

The tour $T^*_{n+1} = P \cup \{(v_{n+1}, a), (v_{n+1}, b)\}$ is an optimum solution on $I_{n+1} = (K_{n+1}, d)$. Let T^*_n be an optimum solution on the instance I_n. Then $d(T^*_n) = n$ iff G has an hamiltonian cycle, and a ρ approximate solution allows to decide if $d(T^*_n) = n$. We get the lower bound setting $\rho = 2^{p(n)}$. □

Proposition 4. *MaxTSP- is NP-hard, even if distances are only 1 and 2.*

Proof. In [1], it is shown that MinTSP- is NP-hard, even if distances are only 1 and 2. We have a trivial reduction from MinTSP- to MaxTSP- if distances are only 1 and 2: we just have to flip the distances between 1 and 2. Solving MinTSP- is equivalent to solve MaxTSP- with the new distances. □

As a final result, let us remark that the deletion strategy has the same behavior in the metric case and in the general one (see appendix for the proof).

Proposition 5. *For MaxTSP-, deletion gives a 1/2-approximation, and this bound is tight (even if the graph has an arbitrary large number of nodes). These results hold in the general case as well as in the metric case.*

These results might be strengthened, but they seem to indicate that the knowledge of an optimum solution in the initial instance may not be really helpful to get good approximation ratios when nodes are deleted.

6 Conclusion

In this article we have proposed some complexity and approximability results for reoptimization versions of TSP. We have exhibited an interesting asymmetry between the maximization and the minimization versions: while we get an almost optimum tour by simply inserting the new node in the right position for MaxTSP+ (in the metric case), this is not true when dealing with the minimization version. One can even show that in order to get an almost optimum solution for MinTSP+, we need, on some instances, to change $n - o(n)$ edges from the initial optimum solution. This leads us to conjecture that MinTSP+ does not admit a PTAS.

Following our approach, an interesting generalization would be to consider TSP in a fully dynamic situation. Starting from a given solution (optimum or approximate) on an initial graph, the graph evolves (nodes are added and deleted), and the goal is to maintain efficiently, along this process, an approximate solution as good as possible. Some of our results can be easily generalized when starting from an approximate (instead of optimum) solution, and can be useful in such approach.

References

1. C. Archetti, L. Bertazzi, and M. G. Speranza. Reoptimizing the traveling salesman problem. *Networks*, 42(3):154–159, 2003.

2. M. Bartusch, R. H. Mohring, and F. J. Radermacher. Scheduling project networks with resource constraints and time windows. *Ann. Oper. Res.*, 16:201–240, 1988.
3. M. Bartusch, R. H. Mohring, and F. J. Radermacher. A conceptional outline of a dss for scheduling problems in the building industry. *Decision Support Systems*, 5:321–344, 1989.
4. Z.-Z. Chen, Y. Okamoto, and L. Wang. Improved deterministic approximation algorithms for Max TSP. *Information Processessing Letters*, 95:333–342, 2005.
5. N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 338, Grad School of Industrial Administration, Canergi-Mellon University, Pittsburgh, 1976.
6. D. Eppstein, Z. Galil, G. F. Italiano, and A. Nissenzweig. Sparsification - a technique for speeding up dynamic graph algorithms. *Journal of the ACM*, 44(5):669–696, 1997.
7. M. R. Garey and D. S. Johnson. *Computer and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.
8. G. Gutin and A. P. Punnen. *The Traveling Salesman Problem and Its Variations.* Combinatorial Optimization. Kluwer Academic Publishers, 2002.
9. D. Hartvigsen. *Extensions of Matching Theory.* PhD thesis, Carnegie-Mellon University, 1984.
10. R. Hassin and S. Rubinstein. A 7/8-approximation algorithm for metric Max TSP. *Information Processessing Letters*, 81(5):247–251, 2002.
11. M. R. Henzinger and V. King. Maintaining minimum spanning trees in dynamic graphs. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *ICALP*, volume 1256 of *Lecture Notes in Computer Science*, pages 594–604. Springer, 1997.
12. D. S. Johnson and L. A. McGeoch. The traveling salesman problem : a case study. In E. Aarts and J. K. Lenstra, editors, *Local search in combinatorial optimization.* Wiley, 1997.
13. E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. *The Traveling Salesman Problem: a guided tour of Combinatorial Optimization.* Discrete Mathematics and Optimization. Wiley, 1985.
14. C. H. Papadimitriou and K. Steiglitz. Some complexity results for the traveling salesman problem. In *STOC'76*, pages 1–9, 1976.
15. C. H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18:1–11, 1993.
16. D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis II. An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.*, 6(3):563–581, 1977.
17. S. Sahni and T. F. Gonzalez. P-complete approximation problems. *J. ACM*, 23(3):555–565, 1976.
18. M. W. Schäffter. Scheduling with forbidden sets. *Discrete Applied Mathematics*, 72(1-2):155–166, 1997.
19. A. I. Serdyukov. An algorithm with an estimate for the traveling salesman problem of the maximum. *Upravlyaemye Sistemy (in Russian)*, 25:80–86, 1984.

Appendix

Proof of Lemma 1:

Proof. We reduce the usual s, t-hamiltonian path problem, known to be **NP**-complete in general graphs of maximum degree 3 ([7]), to $\mathrm{SHP}_{a,b,s,t}$.

Let $G = [V, E]$ be a graph of maximum degree 3 with node set $V = \{v_1, \ldots, v_n\}$. We construct the bipartite graph $H = [V', E']$ containing n copies $H_1, \ldots, H_n$ of the graph depicted in Figure 1.

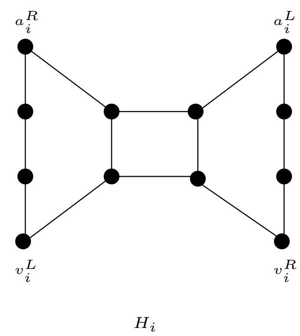

Fig. 1. The graph H_i.

Finally, we connect the copies together in the following way:

- We add the edge set $E_2 = \{(a_i^L, a_{i+1}^R) : i = 1, \ldots, n-1\}$.
- If $(v_i, v_j) \in E$, then we add edges (v_i^R, v_j^L) and (v_i^L, v_j^R).

From this construction we can observe that any hamiltonian path of H must traversed each copy H_i in one of the two following ways: either $P_{1,i}$ or $P_{2,i}$ (see Figure 2).

The resulting graph is bipartite (each copy H_i is bipartite and a left node v_i^L is only connected to a right node v_j^R) of maximum degree 5 and the edge set $P^* = E_2 \cup_{i=1}^n P_{2,i}$ induces an hamiltonian path from a_1^R to a_n^L in H.

We claim that G has an hamiltonian path from v_1 to v_n iff H has an hamiltonian path from v_1^L to v_n^R.

Let $P = (v_{i_1}, \ldots, v_{i_n})$ with $i_1 = 1$ and $i_n = n$ be an hamiltonian path of G. We build the hamiltonian path P' from v_1^L to v_n^R in H using the set of paths $\cup_{i=1}^n P_{1,i}$ and the edge set $\{(v_{i_j}^R, v_{i_{j+1}}^L) : j = 1, \ldots, n-1\}$.

Conversely, let P' be an hamiltonian path from v_1^L to v_n^L in H. Using the previous property of H_i, we deduce that P' must contain all the paths $P_{1,i}$ for $i = 1, \ldots, n$. Thus, the edge set $P = \{(v_i, v_j) : (v_i^R, v_j^L) \in P'\}$ is an hamiltonian path from v_1 to v_n. □

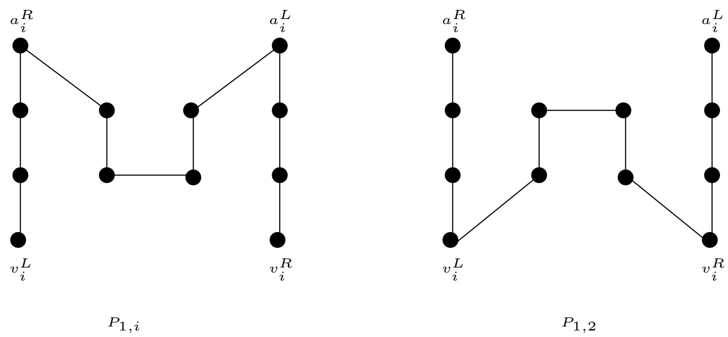

Fig. 2. The two graph hamiltonian path $P_{1,i}$ and $P_{2,i}$ of H_i.

Proof of Proposition 1:

Let v_{i_1} and v_{j_1} be the neighbors of v^* in T_n^* and let v_i and v_j be the neighbors of v_{n+1} in T_{n+1}^*. By construction of Longest Insertion rule, the produced solution T_1 verifies:

$$2d(T_1) \geq 2d(T_n^*)+2d(v^*, v_{n+1})+d(v_{i_1}, v_{n+1})+d(v_{j_1}, v_{n+1})-d(v_{i_1}, v^*)-d(v_{j_1}, v^*) \tag{14}$$

Let us prove that we also have:

$$d(T_1) \geq \frac{1}{2}d(T_{n+1}^*) + \frac{1}{2}(d(v_i, v_{n+1}) + d(v_j, v_{n+1})) \tag{15}$$

Consider the two cases:

- $v^* \notin \{v_i, v_j\}$; when we walk around T_n^* from v_{i_1}, v^*, v_{j_1}, assume that we met v_j and v_i in this order. Thus, using the triangle inequality we deduce:

$$d(T_n^*) \geq d(v_{i_1}, v^*) + d(v^*, v_{j_1}) + d(v_{j_1}, v_j) + d(v_i, v_{i_1}) \tag{16}$$

  The inequality (15) holds using inequalities (14), (16), and the triangle inequality.

- $v^* \in \{v_i, v_j\}$; assume $v^* = v_i$. In this case, we deduce $d(T_n^*) \geq d(v_{i_1}, v_i) + d(v_i, v_{j_1}) + d(v_{j_1}, v_j) + d(v_j, v_{i_1})$ and the inequality (15) also holds.

On the other hand, using the triangle inequality we also get:

$$d(T_1) \geq d(T_n^*) \geq d(T_{n+1}^*) - d(v_i, v_{n+1}) - d(v_j, v_{n+1}) \tag{17}$$

Adding inequality (15) with coefficient 2 and inequality (17), the expected result follows.

In order to show the tightness, consider (I_{n+1}, T_n^*) where $d(v_{n+1}, v_n) = n+1$, $d(v_{n+1}, v_i) = n$, $\forall i = 1, \cdots, n-1$, $d(v_n, v_i) = 2n$, $\forall i = 1, \cdots, n-1$, and

$d(v_i, v_j) = 0$, $\forall i, j$ $1 \leq i < j \leq n-1$. The tour T_n^* is given by the sequence $(v_1, v_2, \cdots, v_n, v_1)$. Longest Insertion on T_n^* gives $d(T_1) = 4n+1$ whereas $d(T_{n+1}^*) = 6n$. □

Complete proof of Theorem 4:

Proof. Let T_n^* be an optimum solution on the initial instance I_n, T_{n+1}^* an optimum solution on the final instance I_{n+1}, and T the solution obtained by applying Best Insertion on T_n^*.

Let $K = \sqrt{n}$ and $1 \leq k \leq K$.

Consider the following subsequence of nodes $(a_k, \cdots, a_1, v_{n+1}, b_1, \cdots, b_k)$ in T_{n+1}^*.

Let J_k be the sub-instance of I_{n+1} induced by all the nodes but v_{n+1}, $a_1, a_2, \cdots, a_{k-1}$ and $b_1, b_2, \cdots, b_{k-1}$ (in particular J_1 is (K_n, d), the initial instance).

We have :

$$d(T_{n+1}^*) \leq d(v_{n+1}, a_1) + d(v_{n+1}, b_1) + \sum_{i=1}^{k-1} d(a_i, a_{i+1}) + \sum_{i=1}^{k-1} d(b_i, b_{i+1}) + opt(J_k)$$

where $opt(J_k)$ is the value of an optimum solution on J_k. Indeed, there is an hamiltonian path in T_{n+1}^* between a_k and b_k, the value of which is at most $opt(J_k)$.

Let $d_m^k(v)$ be the medium distance between a node v and the nodes in J_k, i.e., $d_m^k(v) = \frac{1}{|J_k|} \sum_{v_i \in J_k} d(v, v_i)$. Using the triangle inequality, we get that for any pair (u, v) of nodes (and for any k), $d(u, v) \leq d_m^k(u) + d_m^k(v)$. Hence we get an upper bound on $d(T_{n+1}^*)$:

$$d(T_{n+1}^*) \leq 2 \left(d_m^k(n+1) + \sum_{i=1}^{k-1} d_m^k(a_i) + \sum_{i=1}^{k-1} d_m^k(b_i) \right) + d_m^k(a_k) + d_m^k(b_k) + opt(J_k) \quad (18)$$

Now, our goal is to lower bound first $d(T_n^*)$ and then $d(T)$ in order to get the following inequality :

$$d(T) \geq \left(1 - \frac{O(k)}{n}\right) \left(d(T_{n+1}^*) - d_m^k(a_k) - d_m^k(b_k)\right) \quad (19)$$

To achieve this, first consider an optimum solution $T^*(J_k)$ (of value $opt(J_k)$) of J_k. Consider a particular node v_1 in J_k, and let us call $v_2, \cdots, v_{2k-1}$ the nodes following v_1 in $T^*(J_k)$ (given an arbitrary order). We insert the $2(k-1)$ nodes $a_1, a_2, \cdots, a_{k-1}$ and $b_1, b_2, \cdots, b_{k-1}$ in $T^*(J_k)$ in the following way:

$$v_1, a_1, v_2, a_2, ..., v_{k-1}, a_{k-1}, v_k, b_1, v_{k+1}, ..., b_{k-1}, v_{2k-1}$$

Hence we get a tour on I_n. We apply this construction $n - 2(k-1)$ times, considering for v_1 all the nodes in J_k. Then we get $n - 2(k-1)$ tours on I_n.

However, for each node v of J_k and each a_i, the edge (a_i, v) appears twice (the same holds for (b_i, v)). Moreover, each edge of $T^*(J_k)$ is removed $2(k-1)$ times, hence appears $n-2(k-1)-2(k-1)$ times. This leads to the following inequality:

$$(n-2(k-1))d(T_n^*) \geq 2(n-2(k-1))\left(\sum_{i=1}^{k-1} d_m^k(a_i) + \sum_{i=1}^{k-1} d_m^k(b_i)\right)$$
$$+(n-2(k-1)-2(k-1))opt(J_k)$$

$$d(T_n^*) \geq 2\left(\sum_{i=1}^{k-1} d_m^k(a_i) + \sum_{i=1}^{k-1} d_m^k(b_i)\right) + \left(1 - \frac{2(k-1)}{n-2(k-1)}\right) opt(J_k) \qquad (20)$$

Considering that $k \leq \sqrt{n}$:

$$d(T_n^*) \geq 2\left(\sum_{i=1}^{k-1} d_m^k(a_i) + \sum_{i=1}^{k-1} d_m^k(b_i)\right) + \left(1 - \frac{O(k)}{n}\right) opt(J_k) \qquad (21)$$

Now, we relate $d(T)$ and $d(T_n^*)$. Consider each of the n possible insertions of v_{n+1} in T_n^*. Since each edge of (T_n^*) is removed exactly once, we get:

$$nd(T) \geq nd(T_n^*) + 2\sum_{i=1}^{n} d(v_{n+1}, v_i) - d(T_n^*)$$

However $\sum_{i=1}^{n} d(v_{n+1}, v_i) \geq \sum_{v \in J_k} d(v_{n+1}, v) = (n-2(k-1))d_m^k(n+1)$.

$$d(T) \geq \left(1 - \frac{1}{n}\right) d(T_n^*) + 2\left(1 - \frac{2(k-1)}{n}\right) d_m^k(n+1)$$

Using (21), we derive:

$$d(T) \geq \left(1 - \frac{O(k)}{n}\right)\left(2\left(d_m^k(n+1) + \sum_{i=1}^{k-1} d_m^k(a_i) + \sum_{i=1}^{k-1} d_m^k(b_i)\right) + \left(1 - \frac{O(k)}{n}\right) opt(J_k)\right)$$

Using (18), the previous inequality gives equation (19).

This equation is valid for any k. Now, let us write this inequality for $k = 1, \cdots, K$. Let us consider the two following cases :

1. If, for some k, $d_m^k(a_k) + d_m^k(b_k) \leq \frac{1}{K} d(T_{n+1}^*)$, then we get $d(T) \geq \left(1 - \frac{O(k)}{n}\right)\left(1 - \frac{1}{K}\right) d(T_{n+1}^*)$. Hence:
$$d(T) \geq \left(1 - \frac{O(K)}{n}\right)\left(1 - \frac{1}{K}\right) d(T_{n+1}^*)$$
Since $K = \sqrt{n}$, we get $d(T) \geq \left(1 - O\left(\frac{1}{\sqrt{n}}\right)\right) d(T_{n+1}^*)$.

2. In the other case, for any k, $d_m^k(a_k) + d_m^k(b_k) \geq \frac{1}{K} d(T_{n+1}^*)$. We now show that this is impossible. By making the sum, we get :

$$\sum_{k=1}^{K} d_m^k(a_k) + d_m^k(b_k) \geq d(T_{n+1}^*) \tag{22}$$

The only remaining thing is to upper bound this sum. Let us use (21) with $k = 2$:

$$d(T_n^*) = opt(J_1) \geq 2(d_m^2(a_1) + d_m^2(b_1)) + \left(1 - \frac{2}{n}\right) opt(J_2)$$

If we apply the same inequality for $opt(J_2)$, we get

$$d(T_{n+1}^*) \geq 2(d_m^2(a_1) + d_m^2(b_1)) + 2\left(1 - \frac{2}{n}\right)(d_m^3(a_2) + d_m^3(b_2)) \\ + \left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-2}\right) opt(J_3)$$

Then, by recurrence:

$$d(T_{n+1}^*) \geq 2\left(1 - \frac{2}{n-2K}\right)^K \left(\left(\sum_{k=1}^{K} d_m^{k+1}(a_k) + d_m^{k+1}(b_k)\right) + opt(J_{K+1})\right)$$

Now let us remark that :

$$(n-2k) d_m^{k+1}(a_k) = (n - 2(k-1)) d_m^k(a_k) - d(a_k, b_k)$$

So, $d_m^{k+1}(a_k) \geq d_m^k(a_k) - \frac{d(a_k,b_k)}{n-2k}$, and hence:

$$\sum_{k=1}^{K} d_m^{k+1}(a_k) + d_m^{k+1}(b_k) \geq \left(\sum_{k=1}^{K} d_m^k(a_k) + d_m^k(b_k)\right) - \frac{2\sum_{k=1}^{K} d(a_k, b_k)}{n-2K}$$

But $\sum_{k=1}^{K} d(a_k, b_k)$ is the value of a matching, hence $\sum_{k=1}^{K} d(a_k, b_k) \leq d(T_{n+1}^*)$. Using 22, we get:

$$d(T_n^*) \geq 2\left(1 - \frac{O(K)}{n}\right) d(T_{n+1}^*)$$

This is impossible for n large enough. □

Proof of Proposition 2

Proof. We only present the proof for Best Insertion. Consider an optimum solution T^*_{n+1} and let v_i and v_j be the neighbors of v_{n+1} in T^*_{n+1}. Then

$$d(T^*_{n+1}) \leq d(T^*_n) + d(v_i, v_{n+1}) + d(v_j, v_{n+1})$$

Now, consider the insertion of v_{n+1} between v_i and a neighbor $v_{i'}$ in T^*_n.

$$d(T) \geq d(T^*_n) + d(v_i, v_{n+1}) + d(v_{i'}, v_{n+1}) - d(v_i, v_{i'})$$

Doing the same with a neighbor $v_{j'}$ of v_j, and making the sum leads to :

$$d(T) \geq d(T^*_n) + \frac{1}{2}(d(v_i, v_{n+1}) + d(v_j, v_{n+1}) - d(v_i, v_{i'}) - d(v_j, v_{j'}))$$

To conclude, we just have to notice that $d(v_i, v_{i'}) + d(v_j, v_{j'}) \leq d(T^*_n)$.

For the upper bound, consider the instance where all the distances in (K_n, d) are 0, and consider an optimum solution $(v_1, v_2, \cdots, v_n)$. We assume that n is even. Then we set $d(v_i, v_{n+1}) = 0$ if i is even and 1 if i is odd. Then obviously Best Insertion gives a solution of value 1, while the optimum solution has value 2. □

Proof of Proposition 5

Proof. To get the approximation result, let v_i and v_j the nodes adjacent to v_{n+1} in T^*_{n+1}. Consider now an optimum solution T^*_n, and let $v_{i'}$ and $v_{i''}$ the two neighbors of v_i and $v_{j'}$ and $v_{j''}$ the two neighbors of v_j in T^*_n. T^*_n is $(v_i, v_{i'}, \cdots, v_{j'}, v_j, v_{j''}, \cdots, v_{i''})$ (with possibly $i' = j'$ or $i'' = j''$). Note that if $v_{i'}$ or $v_{i''}$ is v_j, or if $v_{j'}$ or $v_{j''}$ is v_i, then deletion is optimum. Let P' be the sequence of nodes between $v_{i'}$ and $v_{j'}$ (eventually empty) and P'' the one between $v_{i''}$ and $v_{j''}$.

Consider the following solutions on G: $T_1 = (v_{n+1}, v_i, v_{i'}, P', v_{j'}, v_{i''}, P'', v_{j''}, v_j)$, and the symmetrical $T_2 = (v_{n+1}, v_i, v_{i''}, P'', v_{j''}, v_{i'}, P', v_{j'}, v_j)$.

We get :

$$2d(T^*_{n+1}) \geq 2d(v_{n+1,v_i}) + 2d(v_{n+1,v_j}) + 2d(T^*_n) - d(v_i, v_{i'}) - d(v_i, v_{i''}) - d(v_j, v_{j'}) - d(v_j, v_{j''})$$

With deletion we get a solution T such that $d(T) \geq d(T^*_{n+1}) - d(v_{n+1,v_i}) - d(v_{n+1,v_j})$. Taking into account that $d(v_i, v_{i'}) + d(v_i, v_{i''}) + d(v_j, v_{j'}) + d(v_j, v_{j''}) \leq d(T^*_n)$, we get $2d(T) \geq d(T^*_n)$.

For the tightness of the bound, consider a graph on $n+1$ nodes where $d(v_i, v_j) = M$ is $i \leq 2$ and $j \geq 3$ (or vice versa) and $d(v_i, v_j) = 1$ otherwise. Then an optimum solution is $(v_1, v_{n+1}, v_2, v_n, v_{n-1}, \cdots, v_3, v_1)$, with value $4M + n - 3$. When we delete node v_{n+1}, we get a solution of value $2M + n - 2$. However, the value of an optimum solution on the final instance is $4M + n - 4$. The ratio can be arbitrary close to 1/2. □