Exponential approximation schemas for some graph problems

N. Boria, N. Bourgeois, B. Escoffier, V. Th. Paschos

January 27, 2011

1 Introduction

Among network design problems, TRAVELING SALESMAN PROBLEM and MIN STEINER TREE have been extensively studied in combinatorial optimization, due to both their numerous practical applications and their theoretical interest. Among these works, many results deal with complexity and polytime approximation algorithms of these NP-hard problems. In particular, even in restricted versions, these problems are known to be APX-hard. Moreover, many results have been obtained dealing with the exact and/or parameterized resolution of these problems. The goal of this article is to explore the approximability of these problems in superpolynomial or moderately exponential time. Roughly speaking, if a given problem is solvable in time say $O^*(\gamma^n)$ but is NP-hard to approximate within some ratio r, we seek for r-approximation algorithms with complexity - significantly - lower than $O^*(\gamma^n)$. This issue has already been tackled for several other problems such as Minimum Set Cover [10, 6], Minimum Coloring [5], Maximum Independent Set and Minimum Vertex Cover [5], Minimum Bandwidth [14],... Similar issues arise in the field of FPT algorithms, where approximation notions have been introduced for instance in [11, 7]. Among the several natural questions occurring in this setting, two kept most of our attention in this work:

- The first one deals with ratios close to one: for an APX-hard problem solvable in time $O^*(\gamma^n)$, can we find for *any* $\epsilon > 0$ a $1 + \epsilon$ (or $1 - \epsilon$) approximation algorithm working in time $O^*(\gamma_\epsilon^n)$, where $\gamma_\epsilon < \gamma$.
- The second one deals with larger ratios: given a polytime r-approximation algorithm, can we reach ratios slightly better than r at "low" exponential cost, in particular ratios $r - \epsilon$ (or $r + \epsilon$) in time $O^*(\gamma_\epsilon^n)$ where $\gamma_\epsilon \to 1$ when $\epsilon \to 0$?

Both in TRAVELING SALESMAN PROBLEM and in MIN STEINER TREE, a complete undirected graph $G = (V, E)$ is given, together with a distance or cost $c(e) \geq 0$ for each edge $e \in E$. The instance is said to be metric if c satisfies the triangle inequality, ie., $c(u, v) + c(v, w) \geq c(u, w)$ for any three vertices u, v, w.

For MIN STEINER TREE, a subset $S \subseteq V$ of *terminal vertices* (or simply *terminals*) is given. A Steiner tree $T = (V', E')$ is a subgraph of G ($V' \subseteq V$ and $E' \subseteq E$) which is a tree and which spans all the terminals, ie., $S \subseteq V'$. The cost of the tree T is $c(T) = \sum_{e \in E'} c(e)$. The goal of the problem is to find a Steiner tree of minimum cost. Note that the instances are usually assumed to be metric since, if an instance is not metric, one can easily transform it into an equivalent metric one by replacing the cost of each edge $e = (u, v)$ by the shortest path between u and v. The problem is well known to be NP-hard and even APX-hard [1]. The feasible solution consisting of computing a minimum spanning tree on the subset of terminals is a trivial 2-approximate solution, and several works managed to improve this ratio, up to the polytime approximation algorithm of [22] guaranteeing a ratio $1 + \ln(3)/2 < 1.55$. Dealing with exact resolution, the problem is known to be solvable in $O^*(1.36^n)$ (and exponential space) and in $O^*(1.62^n)$ in polynomial space [12]. Moreover, the problem is fixed parameter tractable when the parameter is the number of terminal $k = |S|$: it is solvable in time $O^*((2 + \epsilon)^k)$, for any $\epsilon > 0$, and exponential space [19]. In the case of bounded edge costs, these results can be improved: the problem is solvable in time $O^*(1.36^n)$ and polynomial space, and in time $O^*(2^k)$ and polynomial space, see [21].

A solution of TRAVELING SALESMAN PROBLEM is a tour (Hamiltonian cycle) γ on the vertices. The cost of γ is the sum of the cost of its edges. A lot of different versions of TRAVELING SALESMAN PROBLEM have been studied, including maximization and minimization versions of instances with general or metric costs. These four versions are NP-hard and even APX-hard, but have different behaviors in terms of polytime approximation. In term of exact resolution, it is well known that by a dynamic programming algorithm, all the mentioned versions of TRAVELING SALESMAN PROBLEM are solvable in time and space $O^*(2^n)$ [16]. Note that though this result has not been improved so far, very recently a major breakthrough has been obtained in [2] where the Hamiltonian cycle problem is solved in $O^*(1.66^n)$. In polynomial space, the best running time known so far for TRAVELING SALESMAN PROBLEM is $O^*(4^n n^{\log n})$ reached by the algorithm in [15] (see [3]).

In this article, we study in Section 2 the possibility to get ratios arbitrary close to 1 with a running time better than the one of exact computation. We answer positively by proposing algorithms for MIN STEINER TREEand some versions of TRAVELING SALESMAN PROBLEM. In both cases, the basic idea is to find a small part of the instance verifying some suitable properties, then to solve the instance on the remaining part and to build finally a global solution. In Section 3, we show that one can take advantage of the possible existence of some polytime r-approximation algorithm in order to reach interesting running time for ratios slightly better than r.

In all the article, n is the number of vertices and m the number of edges of a given graph $G = (V, E)$.

2 Obtaining ratios arbitrary close to 1

2.1 MIN STEINER TREE

In what follows, (G, S, c) is an instance of MIN STEINER TREE where G is the input graph, $S \subseteq V$ is the set of terminals and c is the edge cost function. We denote $k = |S|$ the number of terminals. We will propose approximation algorithms that works in time smaller than exact ones. We will present the results according to the best algorithms known so far and given in [21]: when costs are bounded we call `ExactST-PARAM`(G, S) the algorithm that computes an optimum Steiner tree in time $O^*(2^k)$ and `ExactST`(G, S) the one with time complexity $O^*(1.36^n)$. Since we use as subroutines these algorithms, we also make the assumption that costs are bounded. However, it is easy to see that similar results would be obtained by considering other exact algorithms (valid on any edge costs and/or with other complexity).

We first show Lemma 1. Finding subtrees (of a given tree) satisfying some properties has already been used to achieve interesting running times for MIN STEINER TREE in [12]. Here, we adapt this idea to our approximation setting by showing that there exists a subtree of an optimal solution of "small" cost but containing a "large" number of terminals. If T is a tree rooted at r and v is a vertex in T, then $T(v)$ is the subtree of T rooted at v (consisting of v and its descendants).

Lemma 1 *Let T a Steiner tree (with k terminals). For any $p \leq k$, there exists a subtree $\tau \subset T$ containing at least p and at most $3p$ terminals, and whose cost is at most $3c(T)p/k$.*

Proof. Root the tree at some vertex r, and consider the following procedure.

- Set $v \leftarrow r$.
- While there exists a son u of v such that $T(u)$ contains at least $p + 1$ terminals, set $v \leftarrow u$.
- Now, there exists a subset of sons $u_1, \ldots, u_i$ such that the subtree rooted at v but restricted to the sons $u_1, \cdots, u_i$ contains at least $p + 1$ and at most $2p + 1$ terminals (or $2p$ if v is not terminal). Remove this subtree from the graph (but keep v if v has other sons) and go back to step 1 if the remaining graph tree more than $3p$ terminals.

This way, we build a set of subtrees $\tau_1, \cdots, \tau_z$ whose edge sets $E_1, \ldots, E_z$ are a partition of the edges of T, and such that each τ_i contains at most $2p+1$ and at least $p+1$ terminals - except the last one that contain at least p and at most $3p$ terminals. Moreover, a subtree τ_i has at most one vertex (its root) which might be common with some τ_j with $j > i$, hence to each subtree can be associated at least p vertices with no overlap. Thus:

- $c(T) \geq \sum c(\tau_j)$

- $k/(3p) \leq z \leq k/p$

and finally:

$$\exists j, c(\tau_j) \leq c(T) \times \frac{3p}{k}$$

□

Remark 1 *Note that the following facts, that we will need later, hold:*

1. *There is also a subtree $\tau \subset T$ containing at least p and at most $3p$ terminals, and whose cost is at least $c(T)p/k$.*

2. *Dealing with the total number of nodes n_T in the Steiner tree, for any $p \leq n_T$, there exists a subtree $\tau \subset T$ containing p vertices and whose cost is at most $3c(T)p/n_T$.*

The second fact is obtained using a similar proof as Lemma 1. It gives the existence of such a subtree τ with at least p and at most $3p$ vertices, but we can remove vertices in τ' until it contains p vertices.

Using Lemma 1, we are able to propose an approximation algorithm, called `SchemeST-PARAM`, which computes a $(1+\epsilon)$ approximate solution in time smaller than $O^*(2^k)$. This algorithm first identifies thanks Lemma 1 a part of the instance with a partial solution of small cost, contract the graph, apply an exact algorithm on the contracted graph, and patch the two partial solution. Formally, it works as follows:

1. For any subset $W \subset S$ of size at least $p = \epsilon k/3$ and at most ϵk compute `ExactST-PARAM`(G, W). Let W_0 the subset whose steiner tree $T_0 =$ `ExactST-PARAM`(G, W_0) has minimum cost.

2. Build the contraction graph G' : all the nodes from T_0 are replaced by a single node t_0. For any $u \in V \setminus T_0$, $c(t_0, u) = \mathrm{min_{v \in T_0} c(v, u)}$.

3. Compute $T_1 =$ `ExactST-PARAM`$(G', S \cup t_0 \setminus W_0)$.

4. Build $T_0 \cup T_1$, that means the subgraph of G that contains all the vertices and edges of T_0 and $T_1 \setminus t_0$ and for any edge (t_0, u) from T_1 contains the edge of minimum cost between u and T_0.

5. Return a spanning tree of $T_0 \cup T_1$.

Proposition 1 *When costs are bounded, for any $\epsilon \leq 1/5$ `SchemeST-PARAM` outputs a $(1 + \epsilon)$ approximation for* MIN STEINER TREE *with running time $O^*(2^{(1-\epsilon/3)k})$ and polynomial space.*

Proof. First, since t_0 is a terminal of T_1, $T_0 \cup T_1$ is connected, and it is clear that the computed tree contains all the terminals. Thus, the algorithm produces a steiner tree of (G, S).

The complexity of the first step of the algorithm is $O^*\left(\binom{k}{\epsilon k}2^{\epsilon k}\right)$, while in the second step we solve the problem on an instance with at most $k(1-\epsilon/3)$ terminals, hence in time $O^*\left(2^{k-k\epsilon/3}\right)$. But using Stirling's formula, we know that $\binom{k}{\epsilon k} = O^*\left(\frac{1}{(\epsilon^\epsilon(1-\epsilon)^{1-\epsilon})^k}\right)$. Then, we have $\binom{k}{\epsilon k}2^{\epsilon k} = O^*(2^{k-\epsilon k/3})$ for $\epsilon \leq 1/5$.

Let us now consider the approximation ratio. According to Lemma 1, and since we check among other subtrees the heaviest subtree of the optimal, the cost of T_0 is at most $3c(\mathrm{opt(G,S))p/k} = \mathrm{c(opt(G,S))}\epsilon$. On the other hand, the optimum solution on the contraction graph cannot be heavier than the optimal solution in the initial instance, hence $c(T_0) \leq c(\mathrm{opt(G,S)}$. Finally,

$$c(T_0 \cup T_1) \leq c(T_0) + c(T_1) \leq c(\mathrm{opt(G,S)})(1+\epsilon)$$

□

We now show that this parameterized result allows the achievement of a similar result when the parameter is the number of vertices. We will use the second item of Remark 1. We consider the algorithm `Scheme-ST` which works as follows:

1. If $k \leq 3n/8$, then run `ExactST-PARAM`, and outputs the computed solution.

2. Otherwise:

   (a) for any subset $W \subset V$ of size $p = \epsilon n/8$ and at most $\epsilon n/3$ compute a minimum cost spanning tree on $G[W]$. Let W_0 be the subset whose spanning tree T_0 on W_0 has minimum cost.

   (b) Build the contraction graph G' : all the nodes from W_0 are replaced by a single node t_0. For any $u \in V \setminus W_0$, $c(t_0, u) = \min_{\mathrm{v} \in \mathrm{W}_0} \mathrm{c(v,u)}$.

   (c) Set $S' = S \cup t_0 \setminus W_0$ if W_0 contains a terminal vertex, otherwise $S' = S \setminus W_0$. Compute $T_1 =$ `ExactST`(G', S').

   (d) Build $T_0 \cup T_1$, that means the subgraph of G that contains all the vertices and edges of T_0 and $T_1 \setminus t_0$ and for any edge (t_0, u) from T_1 contains the edge of minimum cost between u and T_0.

   (e) Return a tree of $T_0 \cup T_1$ which spans all the terminals.

We denote by $\gamma \in [1.35, 1.36]$ a constant such `ExactST` works in time $O^*(\gamma^n)$.

Proposition 2 *When costs are bounded, for any $\epsilon \leq 4/5$ `SchemeST` outputs a $(1+\epsilon)$ approximation for* MIN STEINER TREE *with running time $O^*(\gamma^{(1-\epsilon/8)n})$ and polynomial space.*

Proof. In the case $k \leq 3n/8$, the output solution is optimal and obtained in time $O^*(2^k) = O^*(2^{3n/8}) = O^*(\gamma^{(1-\epsilon/8)n})$ for $\epsilon \leq 1/3$. Note that a slightly better result could be obtained using `SchemeST-PARAM` instead of the exact algorithm.

In the second case, the algorithm obviously outputs a feasible solution. The running time of the first step is $O^*(\binom{n}{\epsilon/3n})$, while in the second part we solve the problem in an instance with $n(1-\epsilon/8)$ nodes, hence in time $O^*\left(\gamma^{n(1-\epsilon/8)}\right)$. But using Stirling formula, we have $\binom{n}{\epsilon n/3} = O^*(2^{n(1-\epsilon/8})$ for $\epsilon \leq 4/5$.

Dealing with approximation ratio, note that we have $k \geq 3n/8$, hence an optimum Steiner tree τ contains n' vertices where $3n/8 \leq n' \leq n$. Using Remark 1, there exists a subtree of τ with cost at most $\epsilon c(\tau)$ on $\epsilon n'/3$ vertices. Since $\epsilon n/8 \leq \epsilon n'/3 \leq \epsilon n/3$, T_0 has cost at most $\epsilon c(\tau)$. As previously, the optimum solution on the contraction graph cannot be heavier than the optimal solution in the initial instance, hence $c(T_0) \leq c(\tau) = c(\mathrm{opt(G,S)})$. Finally,

$$c(T_0 \cup T_1) \leq c(T_0) + c(T_1) \leq c(\mathrm{opt(G,S)})(1+\epsilon)$$

□

2.2 TSP problems

As mentioned in introduction, it is well known that by a dynamic programming algorithm, optimum TSP (where optimum is minimum or maximum) is solvable in time and space $O^*(2^n)$ [16]. In polynomial space, the best running time known so far is $O^*(4^n n^{\log n})$ reached by the algorithm in [15] (see [3]). We present in this section two approximation algorithms `SchemeTSP-ES` and `SchemeTSP-PS` that provide for some versions of TSP a $(1+\epsilon)$-approximate solution (or $(1-\epsilon)$ if we deal with a maximization version). `SchemeTSP-ES` works in time $O^*(2^{(1-\Theta(\epsilon))n})$ and exponential space, and `SchemeTSP-PS` works in time $O^*(4^{(1-\Theta(\epsilon))n} n^{\log n})$ and polynomial space.

First, it is easy to see that the exact algorithms mentioned above for optimum TSP can be adapted to work within the same running time and space for the optimum cost Hamitonian Path problem when both extremities are fixed. Let us denote `ExactTSP-PS`(G,s,t) and `ExactTSP-ES`(G,s,t) two algorithms that compute an optimum cost Hamiltonian path between s and t respectively in time $O^*(4^n n^{\log n})$ and polynomial space, and in time $O^*(2^n)$ and exponential space.

Let us first describe `SchemeTSP-PS`. It depends on a parameter p (the value of which depends on the version of TSP) and uses `ExactTSP-PS`(G,s,t) as a subroutine. `SchemeTSP-PS` can be described as follows:

1. For any subset $U \subset S$ of size p compute `ExacstTSP-PS`$(G[U],u,v)$ for any two vertices $u,v \in U$. Let U_0, u^*, v^* be the subset and the vertices whose optimum cost Hamiltonian path $\gamma_0 =$ `ExactTSP-PS`$(G[U_0],u^*,v^*)$ has optimum cost.

2. Fix $G' = G[(V \setminus U_0) \cup \{u^*, v^*\}]$

3. Compute $\gamma_1 =$ `ExactTSP-PS`(G',u^*,v^*).

4. Return $\gamma_0 \cup \gamma_1$.

Algorithm `SchemeTSP-ES` is similar up to the following modifications:

- In Step 1, use dynamic programming to compute for any subset $U \subset S$ of size p and for any two vertices $u, v \in U$ an optimal Hamiltonian path between u and v in $G[U]$.
- In Step 3, use `ExactTSP-ES`(G, s, t) instead of `ExactTSP-PS`(G, s, t).

Lemma 2 *If $p \leq n/5$, `SchemeTSP-ES` produces an Hamiltonian cycle in time $O^*(2^{n-p})$ (and exponential space).*

If $p \leq n/3$, `SchemeTSP-PS` produces an Hamiltonian cycle in time $O^*(4^{n-p}n^{\log n})$ and polynomial space.

Proof. Since γ_0 and γ_1 are Hamiltonian paths between u^* and v^* on two subgraphs that intersect only in the extremities u^* and v^* of γ_0 and γ_1, $\gamma_0 \cup \gamma_1$ is an Hamiltonian cycle on G.

For `SchemeTSP-ES`, Step 1 works in time $O^*(\binom{n}{p})$ (for $p \leq n/2$) and Step 3 in time $O^*(2^{n-p})$. Using Stirling's formula, when $p \leq n/5$, this leads to the fact that the global running time is then $O^*(2^{n-p})$.

For `SchemeTSP-PS`, the total running time is $O^*\left(\binom{n}{p}2^p + 4^{n-p}n^{\log n}\right)$. Using again Stirling formula, we get that $\binom{n}{p}2^p = O(4^{n-p})$ for $p \leq n/3$ (actually even for p a bit greater than $n/3$), hence the running time is $O^*(4^{n-p}n^{\log n})$ for $p \leq n/3$. □

Let γ^* be an optimum solution for the problem dealt with.

Lemma 3 *If we deal with a minimization problem (resp. a maximization problem) then $c(\gamma_0) \leq pc(\gamma^*)/n$ (resp. $c(\gamma_0) \geq pc(\gamma^*)/n$).*

Proof. Since we try any possible subsets of vertices of size p, we try in particular all the p-subsequences of consecutive vertices from γ^*; the lightest (resp. heaviest) of these subsequences has cost at most (resp. at least) $pc(\gamma^*)/n$. □

Now, we use this algorithm to several versions of TSP. The first one is the famous MIN METRIC TSP, where the costs satisfy the triangle inequality: $c(u, v) \leq c(u, x) + c(x, v)$ for any vertices u, v, x.

Proposition 3 *It is possible to compute a $(1 + \epsilon)$-approximation for* MIN METRIC TSP*:*

- *in time $O^*(2^{(1-\epsilon/2)n})$, for any $\epsilon \leq 2/5$.*
- *in time $O^*(4^{(1-\epsilon/2)n}n^{\log n})$ and polynomial space, for any $\epsilon \leq 2/3$.*

Proof. Let $\epsilon \leq 1$, and run `SchemeTSP-ES` with $p = n\epsilon/2$. Then $p \leq n/5$ and thanks to Lemma 2 the running time is $O^*(2^{n-p}) = O^*(2^{(1-\epsilon/2)n})$.

Consider now an optimal solution γ^*. By the triangle inequality, if γ' is an optimal solution for TSP on G', then $c(\gamma') \leq c(\gamma^*)$.

Let u', v' be the predecessors of u^* and v^* in γ' (oriented arbitrarily). Then removing from γ' the edges (u', u^*), (v^*, v') and adding the edge (v', u') creates an Hamiltonian path in G' between u^* and v^* of cost $c(\gamma')+c(v', u')-c(u', u^*)-c(v^*, v')$, hence:

$$c(\gamma_1) \leq c(\gamma') + c(v', u') - c(u', u^*) - c(v^*, v') \leq c(\gamma') + c(u^*, v^*) \leq c(\gamma') + c(\gamma_0)$$

where the last inequalities follow from the triangle inequality. Then using Lemma 3 we get:

$$c(\gamma_0 \cup \gamma_1) \leq c(\gamma') + 2c(\gamma_0) \leq c(\gamma^*)(1 + 2p/n) = c(\gamma^*)(1 + \epsilon)$$

For the result in polynomial space, the proof of the ratio is the same. The running time follows from Lemma 2 since for $\epsilon \leq 2/3$, $p \leq n/3$.

□

Now, we show that a similar result holds for the TSP problem when costs are restricted to be integers between 1 and a fixed integer $k \geq 2$, both in the minimization version MIN TSP-k and the maximization version MAX TSP-k.

Proposition 4 *It is possible to compute a $(1+\epsilon)$-approximation for* MIN TSP-*k:*

- *in time $O^*(2^{(1-\epsilon/(k-1))n})$, for any $\epsilon \leq (k-1)/5$.*
- *in time $O^*(4^{(1-\epsilon/(k-1))n} n^{\log n})$ and polynomial space, for any $\epsilon \leq (k-1)/3$.*

Proof. Let $\epsilon \leq (k-1)/5$, and run `SchemeTSP-ES` with $p = (\epsilon n - k)/(k-1)$. Then $p \leq \epsilon n/(k-1) \leq n/5$ and thanks to Lemma 2 the running time is $O^*(2^{n-p})$. Since $p \geq n\epsilon/(k-1) - 2$, we get the claimed running time thanks to Lemma 2.

Consider now an optimal solution γ^*. This solution contains $2x \leq 2p$ edges with one endpoint in U_0 and one in $V \setminus U_0$. If we remove from γ^* all the vertices in U_0 (and their adjacent edges) we get x pathes (possibly consisting of one unique vertex) in $V \setminus U_0$. We build an Hamiltonian path on vertices in $V \setminus U_0$ by adding $x-1$ edges between these pathes, and then by adding two more edges we get an hamiltionian path γ' between u^* and v^* in G'. To build γ' from γ^*, at least $2x$ have been removed, and $x+1$ edges have been added. Since costs are between 1 and k, and using the fact that $c(\gamma^*) \geq n$, we get:

$$c(\gamma_1) \leq c(\gamma') \leq c(\gamma^*) + (x+1)k - 2x \leq c(\gamma^*)(1 + (pk - 2p + k)/n).$$

Using Lemma 3, we get

$$c(\gamma_0 \cup \gamma_1) \leq c(\gamma^*)(1 + (pk - p + k)/n) = c(\gamma^*)(1 + \epsilon)$$

For the result in polynomial space, the proof of the ratio is the same. The running time follows from Lemma 2 since for $\epsilon \leq (k-1)/3$, $p \leq n/3$. □

The case of MAX TSP-k is similar to the previous one.

Proposition 5 *It is possible to compute a $(1-\epsilon)$-approximation for* MAX TSP-k*:*

- *in time $O^*(2^{(1-\epsilon/(2(k-1)))n})$, for any $\epsilon \leq 2(k-1)/5$.*
- *in time $O^*(4^{(1-\epsilon/(2(k-1)))n} n^{\log n})$ and polynomial space, for any $\epsilon \leq 2(k-1)/3$.*

Proof. Let $\epsilon \leq 2(k-1)/5$, and run `SchemeTSP-ES` with $p = \epsilon n/(2(k-1))$. Then $p \leq n/5$ and thanks to Lemma 2 the running time is $O^*(2^{n-p})$ which is the claimed running time.

Consider an optimal solution γ^*. This solution contains $2x \leq 2p$ edges with one endpoint in U_0 and one in $V \setminus U_0$ and contains y edges with both endpoints in U_0, with $x+y=p$. If we remove from γ^* all the vertices in U_0 we get x pathes (possibly consisting of one unique vertex) in $V \setminus U_0$. We build an Hamiltonian path γ' between u^* and v^* in G' by linking these pathes, u^* and v^* by adding $x+1$ edges. $2x+y$ edges of cost at most k have been removed while $x+1$ have been added, for a total cost difference of at most $(2x+y)k - x - 1 \leq 2pk - p$. Hence,

$$c(\gamma_1) \geq c(\gamma') \geq c(\gamma^*) - p(2k-1) \geq c(\gamma^*)(1 - p(2k-1)/n),$$

$$c(\gamma_0 \cup \gamma_1) \geq c(\gamma^*)(1 - p(2k-1)/n) + c(\gamma^*)p/n = c(\gamma^*)(1-\epsilon).$$

For the result in polynomial space, the proof of the ratio is the same. The running time follows from Lemma 2 since for $\epsilon \leq 2(k-1)/3$, $p \leq n/3$. □

3 Improving ratios obtained with polytime algorithms

In this section, we would like to take advantage of the existence for MIN STEINER TREE and for some versions of TSP of polytime approximation algorithms. If there is a polytime r-approximation algorithm, we would like in particular to get ratios $r-\epsilon$ in time $O^*(\gamma_\epsilon^n)$ (or k for MIN STEINER TREE) where $\gamma_\epsilon \to 1$ when $\epsilon \to 0$. A first idea is to modify the algorithms in the previous sections in order to use an approximation algorithm instead of an exact one when computing a solution on the big part of the instance. We will see that this simple idea works for MIN STEINER TREE. However, this generally does not lead to interesting results for TSP problems because in many cases better ratios are known for TSP than for minimum cost Hamiltonian path when both extremities are fixed.

3.1 MIN STEINER TREE

MIN STEINER TREE is approximable in polynomial time within ratio $r_{ST} = 1 + \ln(3)/2$ [22], let us call ApxST the algorithm reaching this ratio. Let us consider the following algorithm, called ExpApxST.

1. For any subset $W \subset S$ of size at least $p = \epsilon k/3$ and at most ϵk:
   (a) Compute T_0 =ExactST-PARAM(G, W).
   (b) Build the contraction graph G' : all the nodes from T_0 are replaced by a single node t_0. For any $u \in V \setminus T_0$, $c(t_0, u) = \min_{v \in T_0} c(v, u)$.
   (c) Compute $T_1 =$ ApxST$(G', S \cup t_0 \setminus W_0)$.
   (d) Build $T_0 \cup T_1$, that means the subgraph of G that contains all the vertices and edges of T and $T_1 \setminus t_0$ and for any edge (t_0, u) from T_1 contains the edge of minimum cost between u and T_0.
   (e) Consider a spanning tree T_W of $T_0 \cup T_1$.
2. Output the best among the solutions S_w computed.

Proposition 6 *ExpApxST outputs a $(r_{ST} - (r_{ST} - 1)\epsilon/3)$-approximate solution in time $O^*(\binom{k}{\epsilon k})$ and polynomial space.*

Proof. The running time of the algorithm is $O^*(\binom{k}{\epsilon k}) = O^*\left((1/\epsilon^\epsilon(1-\epsilon)^{1-\epsilon})^k\right)$. Consider an optimum solution τ. Thanks to Remark 1 after Lemma 1, there is a subtree $\tau' \subset \tau$ containing at least p and at most $3p$ terminals, and whose cost is *at least* $c(\tau)p/k$. When W is the set of terminals of τ', we have $c(T_0) \leq c(\tau')$. But contracting τ' in a single vertex in τ gives a feasible solution of the contracted graph G', whose cost is $c(\tau) - c(\tau')$. It follows that ApxST will output a solution T_1 such that $c(T_1) \leq r_{st}(c(\tau) - c(\tau'))$. Finally, we get:

$$\begin{aligned} c(T_0 \cup T_1) &\leq c(T_0) + c(T_1) \leq c(\tau') + r_{ST}(c(\tau) - c(\tau') \\ &\leq r_{ST}c(\tau) - (r_{ST} - 1)c(\tau)p/k = (r_{ST} - (r_{ST} - 1)\epsilon/3)c(\tau) \end{aligned}$$

□

By setting $\epsilon' = (r_{ST} - 1)\epsilon/3$, this is a $(1 + \epsilon)$ approximation in time $O^*(\gamma_{\epsilon'}^k)$, where $\gamma_{\epsilon'}$ goes to 0 when ϵ' goes to 0. For instance, this gives a 1.547-approximation ($r_{ST} \sim 1.5493...$) in time $O^*(1.07^k)$.

3.2 TSP problems

As mentioned, the same simple idea does not lead to interesting results for TSP problems since to patch partial solutions we need to solve minimum cost Hamiltonian path problems, and in many cases better ratios are known for TSP than for minimum cost Hamiltonian path when both extremities are fixed. Indeed, the famous MIN METRIC TSP is approximable within ratio 3/2 [9], but for the MIN METRIC HP (when both extremities are fixed) the best known ratio achievable in polynomial time is 5/3 [17]. For maximization versions, MAX TSP and MAX METRIC TSP are approximable with asymptotic ratios of 61/81 and 17/20 [17], while only a 1/2-approximation algorithm is known for the corresponding versions of maximum cost Hamiltonian path when both extremities are fixed [20].

So, the idea is tu use an approximation algorithm for TSP to get an Hamiltonian cycle γ_1 on G', and then to somehow combine γ_0 and γ_1 to get an Hamiltonian cycle on G.

Suppose that `PolyAPP` is a polytime r-approximate algorithm for a TSP problem. We consider the following algorithm `MExpAPP-PS` that works for maximization versions.

- For any subset $U \subset S$ of size $p = \epsilon n$ and for any two vertices $u, v \in U$:
  - compute γ_0 =`ExactTSP-PS`$(G[U], u, v)$
  - Fix $G' = G[V \setminus U_0]$.
  - Compute γ_1 = `PolyAPP`(G').
  - Let γ_1' be the path obtained from γ_1 by removing the lightest edge (z, t) of γ_1 and adding edges (u, z) and (t, v).
  - Consider the solution $\gamma_0 \cup \gamma_1'$.
- Return the best solution computed.

We also consider the exponential space version `MExpAPP-ES` where we first compute by dynamic programming for any subset $U \subset S$ of size $p = \epsilon n$ and for any two vertices $u, v \in U$ an optimum cost Hamiltonian path between s and t in $G[U]$.

Proposition 7 *If `PolyAPP` is a r-approximate algorithm, then `MExpAPP-PS` and `MExpAPP-ES` are $(r + \epsilon(1-r) - O(1/n))$-approximation algorithms. `MExpAPP-PS` runs in time $O^*(\binom{n}{\epsilon n} 4^{\epsilon n} n^{\log n})$ and polynomial space. `MExpAPP-ES` runs in time $O^*(\binom{n}{\epsilon n})$ (and exponential space).*

Proof. It is easy to see that the running time of `MExpAPP-PS` is $O^*(\binom{n}{p} 4^p p^{\log p})$, while dynamic programming leads to a running time of $O^*(\binom{n}{p})$ for `MExpAPP-ES`.

Let γ^* be an optimum solution for the problem dealt. Let us consider all the n sequences $(u', u, \cdots, v, v')$ of $p + 2$ consecutive vertices in γ^* (oriented arbitrarily), and let us denote by $\gamma^*_{u,v}$ the path from u to v in γ^*. Then, by an average based argument, for any nonnegative μ, ν, there exist u, v such that:

$$\mu c(\gamma^*_{u,v}) - \nu(c(u', u) + c(v', v)) \geq \mu \frac{p}{n} c(\gamma^*) - \nu \frac{2}{n} c(\gamma^*) = (\epsilon\mu - 2\nu/n)c(\gamma^*)$$

We consider in the sequel the sequence for the following values:

$$\begin{aligned}\mu &= 1 - r\left(1 - \frac{1}{(1-\epsilon)n}\right)\\ \nu &= r\left(1 - \frac{1}{(1-\epsilon)n}\right)\end{aligned}$$

Obviously $c(\gamma^*) = c(\gamma^*_{u,v}) + c(\gamma^*_{v,u})$. If U_0 denotes the vertices in $\gamma^*_{u,v}$, we consider the solution built by the `MExpAPP-PS` (or `MExpAPP-ES`) for U_0, u, v. Of course, $c(\gamma_0) = c(\gamma^*_{u,v})$. Moreover, since we have removed the lightest edge in γ_1, we have:

$$c(\gamma_1') \geq \left(1 - \frac{1}{(1-\epsilon)n}\right) c(\gamma_1) \geq \left(1 - \frac{1}{(1-\epsilon)n}\right) rc(\gamma_1^*) = \nu c(\gamma_1^*)$$

where $c(\gamma_1^*)$ is an optimum cost Hamiltonian cycle in G'. Finally the output solution S verifies:

$$c(S) \geq c(\gamma_0) + c(\gamma_1') \geq c(\gamma^*_{u,v}) + \nu c(\gamma_1^*)$$

Now, removing edges (u, u') and (v, v') and adding (u', v') in $\gamma^*_{v,u}$ results in an Hamiltonian cycle in G', hence:

$$c(\gamma_1^*) \geq c(\gamma^*_{v,u}) - c(u, u') - c(v, v') = c(\gamma^*) - c(\gamma^*_{u,v}) - c(u, u') - c(v, v')$$

Finally

$$\begin{aligned}c(S) &\geq \nu c(\gamma^*) + \mu c(\gamma^*_{u,v}) - \nu(c(u', u) + c(v', v))\\ &\geq \nu c(\gamma^*) + (\epsilon\mu - 2\nu/n)c(\gamma^*) = (r + \epsilon(1-r) - O(1/n))c(\gamma^*)\end{aligned}$$

□

Let us apply Proposition 7 using the results of [8]: MaxTSP and MaxMetricTSP are approximable in polynomial time with (asymptotic) ratios 61/81 and 17/20 respectively.

ratio	0.753	0.76	0.77	0.8	0.85
running time (exp space)	P	1.14^n	1.29^n	1.63^n	1.96^n
running time (poly space)	P	1.42^n	1.65^n	2.12^n	3.37^n

Table 1: Approximation of MaxTSP

We conclude this section by tackling min asymmetric TSP, the directed version of the (symmetric) min metric TSP. A well known $(1 + \log(n))$-approximation algorithm has been proposed in [13] (recently improved to $0.842 \log(n)$ in [18]). We show that the algorithm of [13] can be very easily adapted to get interesting tradeoffs between running time and approximation.

ratio	0.85	0.86	0.87	0.9
running time (exp space)	P	1.28^n	1.49^n	1.89^n
running time (poly space)	P	1.42^n	1.79^n	3.00^n

Table 2: Approximation of MaxMetricTSP

Proposition 8 *For any integer $k \geq 1$, it is possible to compute a $(1+k)$-approximation of* MIN ASYMMETRIC TSP*:*

- *in time $O^*(2^{n/k})$*
- *in time $O^*(4^{n/k}n^{\log n})$ and polynomial space*

Proof.

The algorithm in [13] works as follows. Starting from $G_0 = G$, it computes a minimum cost 2-factor C_0 of the graph, ie., a minimum cost collection of cycles such that each vertex is in exactly one cycle. Of course, an optimal cost Hamiltonian cycle γ_0^* in G_0 is a particular 2-factor, hence $c(\gamma_0^*) \geq c(C_0)$. If C_0 contains only one cycle, we are done, otherwise we choose one arbitrary vertex in each cycle of C_0, and build the subgraph G_1 of G_0 induced by these vertices. We iterate the same process, and get a 2-factor C_1 of cost $c(C_1) \leq c(\gamma_1^*)$, where γ_1^* is an optimum cost Hamiltonian cycle on G_1. The process ends at some step t, when C_t has only one cycle.

The union of all the cycles in all the 2-factors $C_0, \cdots, C_t$ is a strongly connected graph where all vertices have even in-degree and even out-degree, so by taking shortcuts, using the triangle inequality we get an Hamiltonian cycle γ such that:

$$c(\gamma) \leq \sum_{i=0}^{t} c(C_i) \leq \sum_{i=0}^{t} c(\gamma_i^*)$$

By triangle inequality, $c(\gamma_i^*) \leq c(\gamma_0^*)$ hence

$$c(\gamma) \leq (1+t)c(\gamma_0^*)$$

Now, since each cycle has at least 2 vertices, the number n_i of vertices in G_i verifies $n_i \leq \lfloor n_{i-1}/2 \rfloor$, and this leads to $n_i \leq n_0/2^i$.

Here, instead of repeating this process until C_i has only one cycle, we repeat it until C_k (of course unless the algorithm has stopped before), and then we compute an exact solution on G_k in time $O^*(2^{n_k})$. Since $n_k \leq n_0/2^k = n/2^k$, the claimed running time follows. Moreover, we have:

$$c(\gamma) \leq \sum_{i=0}^{k-1} c(C_i) + c(\gamma_k^*) \leq kc(\gamma_0^*) + c(\gamma_k^*) = (1+k)c(\gamma_0^*)$$

The time bound follows from the fact that the asymmetric cost Hamiltonian path is solvable in time $O^*(2^n)$ (and exponential space), or in time $O^*(4^n n^{\log n})$ and polynomial space.

□

References

[1] Marshall W. Bern and Paul E. Plassmann. The steiner problem with edge lengths 1 and 2. *Inf. Process. Lett.*, 32(4):171–176, 1989.

[2] Andreas Björklund. Determinant sums for undirected hamiltonicity. In *FOCS*, pages 173–182. IEEE Computer Society, 2010.

[3] Andreas Björklund and Thore Husfeldt. Exact algorithms for exact satisfiability and number of perfect matchings. *Algorithmica*, 52(2):226–249, 2008.

[4] Hans L. Bodlaender and Michael A. Langston, editors. *Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, Zürich, Switzerland, September 13-15, 2006, Proceedings*, volume 4169 of *Lecture Notes in Computer Science*. Springer, 2006.

[5] Nicolas Bourgeois, Bruno Escoffier, and Vangelis Th. Paschos. Efficient approximation of combinatorial problems by moderately exponential algorithms. In Frank K. H. A. Dehne, Marina L. Gavrilova, Jörg-Rüdiger Sack, and Csaba D. Tóth, editors, *WADS*, volume 5664 of *Lecture Notes in Computer Science*, pages 507–518. Springer, 2009.

[6] Nicolas Bourgeois, Bruno Escoffier, and Vangelis Th. Paschos. Efficient approximation of min set cover by moderately exponential algorithms. *Theor. Comput. Sci.*, 410(21-23):2184–2195, 2009.

[7] Liming Cai and Xiuzhen Huang. Fixed-parameter approximation: Conceptual framework and approximability results. In Bodlaender and Langston [4], pages 96–108.

[8] Zhi-Zhong Chen, Yuusuke Okamoto, and Lusheng Wang. Improved deterministic approximation algorithms for Max TSP. *Inf. Process. Lett.*, 95(2):333–342, 2005.

[9] Nicos Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical report 338, Grad. School of Industrial Administration, CMU, 1976.

[10] Marek Cygan, Lukasz Kowalik, and Mateusz Wykurz. Exponential-time approximation of weighted set cover. *Inf. Process. Lett.*, 109(16):957–961, 2009.

[11] Rodney G. Downey, Michael R. Fellows, and Catherine McCartin. Parameterized approximation problems. In Bodlaender and Langston [4], pages 121–129.

[12] Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. Faster steiner tree computation in polynomial-space. In Dan Halperin and Kurt Mehlhorn, editors, *ESA*, volume 5193 of *Lecture Notes in Computer Science*, pages 430–441. Springer, 2008.

[13] A. Frieze, G. Galbiati, and F. Maffioli. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 12:23–39, 1982.

[14] Martin Fürer, Serge Gaspers, and Shiva Prasad Kasiviswanathan. An exponential time 2-approximation algorithm for bandwidth. In Jianer Chen and Fedor V. Fomin, editors, *IWPEC*, volume 5917 of *Lecture Notes in Computer Science*, pages 173–184. Springer, 2009.

[15] Yuri Gurevich and Saharon Shelah. Expected computation time for hamiltonian path problem. *SIAM J. Comput.*, 16(3):486–502, 1987.

[16] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.

[17] J.A. Hoogeveen. Analysis of christofides' heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, 10:291–295, 1991.

[18] Haim Kaplan, Moshe Lewenstein, Nira Shafrir, and Maxim Sviridenko. Approximation algorithms for asymmetric tsp by decomposing directed regular multigraphs. *J. ACM*, 52(4):602–626, 2005.

[19] Daniel Mölle, Stefan Richter, and Peter Rossmanith. A faster algorithm for the steiner tree problem. In Bruno Durand and Wolfgang Thomas, editors, *STACS*, volume 3884 of *Lecture Notes in Computer Science*, pages 561–570. Springer, 2006.

[20] Jérôme Monnot. Approximation algorithms for the maximum hamiltonian path problem with specified endpoint(s). *European Journal of Operational Research*, 161(3):721–735, 2005.

[21] Jesper Nederlof. Fast polynomial-space algorithms using möbius inversion: Improving on steiner tree and related problems. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *ICALP (1)*, volume 5555 of *Lecture Notes in Computer Science*, pages 713–725. Springer, 2009.

[22] Gabriel Robins and Alexander Zelikovsky. Improved steiner tree approximation in graphs. In *SODA*, pages 770–779, 2000.