Saving an Epsilon: A 2-approximation for the k-MST Problem in Graphs

Naveen Garg
Computer Science and Engineering
Indian Institute of Technology
Hauz Khas, New Delhi
naveen@cse.iitd.ac.in

ABSTRACT

We present a polynomial time 2-approximation algorithm for the problem of finding the minimum tree that spans at least k vertices. Our result also leads to a 2-approximation algorithm for finding the minimum tour that visits k vertices and to a 3-approximation algorithm for the problem of finding the maximum number of vertices that can be spanned by a tree of length at most a given bound.

Categories and Subject Descriptors

F.2.2 [**Nonnumerical algorithms and Problems**]: Analysis of Algorithms and problem complexity—*Computations on Discrete Structures*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph Algorithms, Trees*

General Terms

Algorithms, Theory

Keywords

Approximation Algorithms, Spanning Trees

1. INTRODUCTION

Given an undirected graph $G = (V, E)$ with non-negative edge lengths $l : E \rightarrow \Re^+$ we consider the problem of finding a minimum length tree spanning any k vertices; we call such a tree a k-MST. This problem was first considered by [17] who proved it to be **NP**-hard; the authors also gave a $3\sqrt{k}$ approximation algorithm for this problem. Although the problem is easily stated and algorithms for finding small spanning trees are very well understood, it was a long sequence of improvements [4, 16, 8, 11, 3] that led to the current bound of $(2 + \epsilon)$ [2]; this algorithm has a running time that is exponential in ϵ^{-1}. In this paper we present a polynomial time 2-approximation algorithm for this problem.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
STOC'05, May 22-24, 2005, Baltimore, Maryland, USA.
Copyright 2005 ACM 1-58113-960-8/05/0005 ...$5.00.

A special case of the problem arises when we are given n points in the plane with the Euclidean metric and we wish to find the minimum length tree spanning any k points. Once again a long sequence of results improved the approximation guarantee from an initial $O(k^{1/4})$ [17] to $O(\log k)$ [12] to $O(\log k/\log\log n)$ [10] to $O(1)$ [6] to $2\sqrt{2}$ [15] and finally to $(1+\epsilon)$ [1, 14] with the last algorithm having a running time that is exponential in ϵ^{-1}.

A problem similar to the minimum k-MST problem is that of finding *a tour that visits at least k vertices and has minimum length*; we call this the minimum k-TSP. A bounded approximation guarantee for this problem can be obtained only under the assumption that the edge-lengths satisfy triangle inequality. Our approximation algorithm for the minimum k-MST also yields a 2-approximation algorithm for this problem.

In the *budget* version of the k-MST problem we wish to find a tree of length at most a given budget B which spans the maximum number of vertices. Johnson et.al. [13] gave a $5 + \epsilon$ approximation for this problem and observed that a 2-approximation for k-MST would yield a 3-approximation for this budget version.

The paper introduces some new ideas and techniques which we believe will find use in solving similar such problems. We modify the Goemans-Williamson procedure for the prize collecting traveling salesman problem and define threshold potential as one at which some tree spans at least k vertices. We give an effective procedure for computing the threshold potential. The lower bound we use for our approximation algorithm is also not straightforward since we really have to solve many different subproblems and take the best solution from amongst them. We extend the Goemans-Williamson analysis to argue that in each iteration the edges of the tree we pick see a total dual that is at most twice the increase in dual values for some relevant sets. Here, we crucially use the fact that in the Goemans-Williamson argument we do not need the contribution from one of the active leaf components to get a bound of two. Further, we also have a case when the picked set of edges might from a cycle on the active components. However, in such a case we can have only one cycle and so the small slack in the Goemans-Williamson analysis saves the argument.

The above discussion also highlights the key reason why we cannot argue that the cost of the tree picked is at most the cost of the best k-stroll. As was shown in [9] such an argument can be made for certain values of k by exploiting the same slack in the Goemans-Williamson argument.

In this paper we present a 2-approximation algorithm for the unrooted version of the k-MST problem. In the rooted version a specified vertex r has to be included in the k vertices we pick. However, these two versions of the problem are equivalent. This is because, an α-approximation for the unrooted version can be used to obtain an α-approximation for the rooted version. This is done by attaching n vertices at the root with 0 cost edges and searching for a tree with $n+k$ vertices. Clearly any such tree will have to include the root vertex and the n vertices attached to it. Conversely, an α-approximation for the rooted version can be used to obtain an α-approximation to the unrooted version by running the algorithm for all n choices of the root and picking the best tree found.

Note that this equivalence of the rooted and unrooted versions does not extend to the budget version of the k-MST problem since in showing that the rooted version of the problem can be solved using an algorithm for the unrooted version we modified the optimum value of the budget version. Thus, while a 3-approximation is now known for the unrooted version we do not know of any bound for the rooted version.

The first result showing a constant approximation for a budgeted, rooted version was obtained by [7] for the orienteering problem: given a budget B and a starting vertex s we wish to find a path of length at most B which starts at s and includes the maximum number of vertices. The 4-approximation in [7] was improved to a 3-approximation by [5]. We believe that the ideas developed in this paper should lead to a 2-approximation algorithm for this problem.

2. THE 2-APPROXIMATION ALGORITHM

We begin by discussing a variation of the primal-dual algorithm by Goemans and Williamson for the *Prize Collecting Traveling Salesman problem*. This procedure has two phases — *a growth phase* and a *pruning phase*. We begin the growth phase by assigning every vertex an initial potential p. We will assign values y_S to $S \subset V$ in such a manner that $\forall e \in E, \sum_{S:e\in\bigtriangledown(S)} y_S \leq l_e$; here $\bigtriangledown(S)$ denotes the edges in the cut $(S, \overline{S})$. An edge for which $\sum_{S:e\in\bigtriangledown(S)} y_S = l_e$ is called *tight*. Let F be the set of tight edges at any point in this procedure and let $\mathcal{C}$ be the connected components in the subgraph (V, F). With every component in $\mathcal{C}$ we associate a non-negative potential; a component with zero potential is *passive* and a component with non-zero potential is *active*. Initially F is empty and hence $\mathcal{C}$ are the singleton vertices in V each of which has potential $p > 0$.

We raise y_S for every active component S uniformly, while simultaneously reducing the potentials of these active components by an equal amount. This is done till either an edge goes tight or some component S, becomes passive. In the latter case we do not increase y_S any further and continue with the remaining active components. When an edge $e = (u, v)$ goes tight we combine the two components containing the endpoints of the edge e, say S_u, S_v, into one component. The potential of the new component $S_u \cup S_v$ is the sum of the potentials of S_u and S_v. The edge e is included in F. Note that the edges in F form a forest at any point in this procedure.

The growth phase continues until all components become passive. Let F_p be the final forest obtained. Let $\mathcal{S}_p$ denote the collection of vertex-subsets which have a non-zero dual, that is

$$\mathcal{S}_p = \{S \subseteq V | y(S) > 0\}$$

Note that these are the sets that were active components at some stage of the growth phase. Hence $\mathcal{S}_p$ is a laminar family. Let $\pi : \mathcal{S}_p \rightarrow \Re^+$ be such that $\pi(S)$, $S \in \mathcal{S}_p$, is the potential of S when it was created; we call this the initial potential of set S. If $V \notin \mathcal{S}_p$ then we add V to $\mathcal{S}_p$ and assign $\pi(V) = 0$.

Suppose S merges with some other set at time t. We color S white if S was active at time t, S is colored *grey* if it was active before t but becomes passive at time t and S is colored *black* if it was passive before t. The function $\chi_p : \mathcal{S}_p \rightarrow \{\text{black, white, grey}\}$, captures this coloring. Thus $\chi_p(S), S \in \mathcal{S}_p$ is the color assigned to S. If S does not merge with any other set then it would be colored black.

In the pruning phase we delete some edges of F_p. Let F be the current set of edges; initially $F = F_p$. If there exists a black set S such that there is only one edge in F, say e, that crosses S, *ie.* $e \in \bigtriangledown(S)$, then we remove e from F. The black sets can be considered in any order. However, if we were to consider the black sets in the reverse order in which they were formed the we would not have to consider any set more than once. The pruning phase stops only when no black set has degree one in F. We use a hat on top of an edge set to denote the edges that remain after pruning. Thus, $\widehat{F_p}$ are the edges of F_p which remain after pruning. For the rest of this paper we use `modified-GW`(p) to denote a run of the modified Goemans Williamson procedure with an initial potential p on each vertex.

Let $\alpha(p)$ be the maximum number of vertices spanned by a tree in $\widehat{F_p}$. For now we assume that we have a procedure $\mathcal{Q}$ to determine an initial potential q such that $\alpha(q+) \geq k$ and $\alpha(q-) < k$ where $q-$ (resp $q+$) is infinitesimally smaller (resp. larger) than q. The procedure $\mathcal{Q}$ takes as input the graph $G = (V, E)$ with edge lengths $l : E \rightarrow \Re^+$ and returns the threshold potential q.

Let O^* be the vertices in an optimum solution and let O be the inclusion-wise minimal set in $\mathcal{S}_{q+}$ that includes all vertices of O^*; by including V in $\mathcal{S}_{q+}$ we have ensured that O exists.

LEMMA 1. *The minimum tree spanning O^* has length at least $q * k - \pi(O)$.*

Let T be the tree in $\widehat{F_{q+}}$ that spans at least k vertices and Q be a set in $\mathcal{S}_{q+}$ that includes all vertices of T.

LEMMA 2. *There exists a tree T^*, spanning exactly k vertices and having length at most $2(q * k - \pi(Q))$.*

Our proof of this lemma is constructive. We exhibit a procedure $\mathcal{T}$ to find a tree T^*.

We use these two lemmas as follows. Let $Q \in \mathcal{S}_{q+}$ be the set containing T with maximum initial potential. If $\pi(O) \leq \pi(Q)$ then Lemma 2 finds a tree T^* spanning k vertices which is a 2-approximation to the optimum. But if $\pi(O) > \pi(Q)$ then we consider all sets of $\mathcal{S}_{q+}$ with initial potential strictly larger than $\pi(Q)$. Let $\mathcal{M}$ denote the inclusion-wise maximal sets in this collection. Note that the sets in $\mathcal{M}$ are disjoint and no set contains all vertices of T.

For every set $S \in \mathcal{M}$ we find a 2-approximation to the k-MST in the graph induced by the vertices of S. Then among the $|\mathcal{M}| + 1$ trees so found the tree with minimum

length would be a 2-approximation to the optimum k-MST in G. This implies that the number of different subgraphs in which we would have to run procedures $\mathcal{Q}$,$\mathcal{T}$ is at most n.

3. THE PROCEDURE $\mathcal{Q}$

In the growth phase of `modified-GW`(p) it may happen that at some stage edges e_1, e_2 between components S_1, S_2 go tight simultaneously. Note that in the forest of tight edges only one of e_1, e_2 would be picked. Further, it might be the case that for an initial potential $p-$ the edge e_1 gets tight before e_2 while for an initial potential $p+$ the edge e_2 gets tight before e_1. To resolve such ambiguities we always consider initial potentials in the form $p+, p-$. It may still happen that at an initial potential $p-$ (or $p+$) two edges go tight simultaneously. In such a case we would pick that edge which occurs first in some fixed lexicographic ordering of the edges. Note that for initial potentials of the form $p-, p+$ no set is colored grey.

To determine the threshold potential we start with values of potential min, max such that $\alpha(min) < k$ and $\alpha(max) > k$. We then try to narrow this range of potentials. Suppose at some point (l, r) is the open interval such that $\alpha(l+) < k$ and $\alpha(r-) > k$. Consider the order in which edges get tight when we run `modified-GW`$(l+)$ and `modified-GW`$(r-)$ and let the first $i-1$ edges in this order be identical and let them be $e_1, e_2, \ldots, e_{i-1}$.

We now consider runs of `modified-GW`() only till the point that the i^{th} edge tightens. Let $\mathcal{S}_p^i$ denote the restriction of $\mathcal{S}_p$ to the sets that were created before the i^{th} edge got tight in a run of `modified-GW`(p).

It is easy to see that the collections $\mathcal{S}_{l+}^i$ and $\mathcal{S}_{r-}^i$ are identical; this follows from the fact that the same edges are getting tightened in the same order in the two runs of `modified-GW`(). This implies that for all $p \in (l, r)$, the collections $\mathcal{S}_p^i$ are identical. All non-maximal sets in ${\mathcal{S}_p}^i$ have merged with other sets and so they can be assigned colors. Let χ_p^i be this coloring. We assume that χ_{l+}^i, χ_{r-}^i are identical; this assumption will be removed later. This in turn implies that for all $p \in (l, r)$, χ_p^i is the same.

Let e_i (resp. f_i) be the i^{th} edge that gets tight when we run `modified-GW`$(l+)$ (resp. `modified-GW`$(r-)$). We first consider the case when e_i, f_i are different and determine how, the time at which these edges get tight, changes with change in the initial potential.

Consider the edge e_i. The fact that $\mathcal{S}_p^i$ and χ_p^i are identical for $p \in (l, r)$ implies that the time at which e gets tight, decreases linearly with the initial potential. This is true except for a small "kink" which arises if one of the two inclusion-wise maximal sets in $\mathcal{S}_{l-}^i$ which contain the end-points of e_i changes from black to white. Suppose this happens at an initial potential p_1. In other words, at an initial potential p_1, one of the sets which merged when e_i got tight becomes inactive at the instant e_i goes tight. As shown in Figure 3 we can draw a plot of the time that e_i gets tight versus the initial potential. Note that in making this plot we ignore all edges other than $e_1, e_2, \ldots, e_{i-1}$. A similar such plot can be drawn for the edge f_i; the only difference here is in the fact that f_i might not get tight when the initial potential is less than some threshold, say p_2. Let p_3 be the potential at which there is a "kink" in the plot of f_i. Further, if these two plots intersect then p_4 be that initial potential.

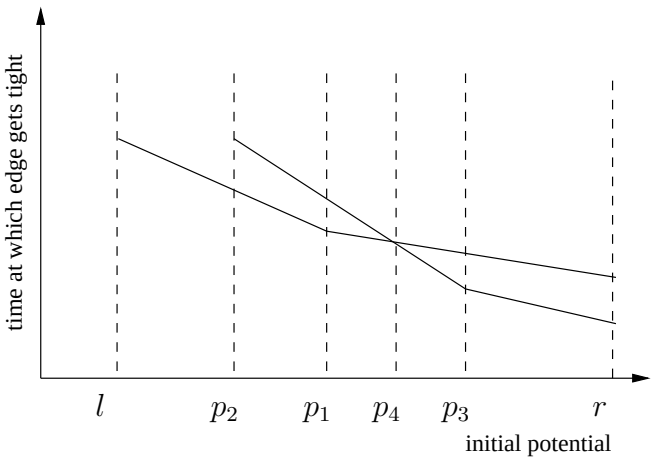

Figure 1: Determining the i^{th} edge to go tight

The potentials p_1, p_2, p_3, p_4 divide the interval (l, r) into at most 5 sub-intervals. By considering $\alpha(p-), \alpha(p+)$ for $p \in \{p_1, p_2, p_3, p_4\}$, we either identify a threshold potential or reduce the range (l, r) to one of these sub-intervals. In the latter case we would have made progress since we would have eliminated one of e_i, f_i as candidates for the i^{th} tight edge.

In this manner we keep narrowing the range until we reach a stage when the i^{th} edge to get tight when we run `modified-GW`$(l+)$ and `modified-GW`$(r-)$ are identical. This also implies that for all potentials, $p \in (l, r)$, the collection of sets $\mathcal{S}_p^{i+1}$ are identical.

This now brings us to the next case when edges e_i, f_i are the same. Let S_1, S_2 be the two components containing the endpoints of e_i. Note that S_1, S_2 would not be maximal in $\mathcal{S}_p^{i+1}$ and so we need to ensure that they are assigned the same color in χ_{l+} and χ_{r-}. Note that if $\chi_{l+}(S_1)$ is white then so is $\chi_{r-}(S_1)$. The same holds true for set S_2. Also note that at least one of $\chi_{l+}(S_1), \chi_{l+}(S_2)$ is white. Hence the only interesting case is when $\chi_{l+}(S_1)$ is white and $\chi_{l+}(S_2)$ is black while both $\chi_{r-}(S_1), \chi_{r-}(S_2)$ are white. Note that by increasing the initial potential we advance the time at which e_i gets tight and this in turn would lead to S_2 becoming white. We determine the minimum potential, say q, for which this happens. Note that $\chi_q(S_2)$ is grey while $\chi_{q-}(S_2)$ is black and $\chi_{q+}(S_2)$ is white. Once again by considering $\alpha(q+), \alpha(q-)$ we either find a threshold potential or are able to update the range of potentials so that for all potentials, p, in this new range χ_p^{i+1} are identical.

If we continue in this manner and are not able to find any threshold potential then we would have a range (l, r) such that $F_{l+} \subseteq F_{r-}$. We first consider the possibility that $F_{l+} \subset F_{r-}$ and determine the minimum initial potential, say q, in the range (l, r) for which some edge not in F_{l+} gets tight. Once again we consider $\alpha(q+), \alpha(q-)$ and either determine that q is a threshold potential or update the range of potentials appropriately.

Eventually, our range (l, r) will satisfy the properties that F_{l+}, F_{r-} are identical, $\mathcal{S}_{l+}, \mathcal{S}_{r-}$ are identical and so are χ_{l+}, χ_{r-}. This implies that $\widehat{F_{l+}}$, $\widehat{F_{r-}}$ are also identical. But we know that $\alpha(l+) < k$ and $\alpha(r-) > k$. This yields a contradiction and hence at some stage we would have found a threshold potential, q.

4. THE PROCEDURE $\mathcal{T}$

Let q be the threshold potential returned by procedure $\mathcal{Q}$. We consider a sequence of steps for transforming the tuple $(\mathcal{S}_{q-}, \chi_{q-}, F_{q-})$ to the tuple $(\mathcal{S}_{q+}, \chi_{q+}, F_{q+})$. Let $(\mathcal{S}, \chi, F)$ be the current tuple in this transformation.

Note that the collections $\mathcal{S}_{q+}, \mathcal{S}_{q-}$ differ only in sets which have a zero initial potential in `modified-GW`(q). Such sets would be part of $\mathcal{S}_{q+}$ but not of $\mathcal{S}_{q-}$. We first consider these sets in the order in which they were formed in `modified-GW`$(q+)$. Let S be the set under consideration and suppose it was formed by the merging of sets S_1, S_2 along the edge e; note that $e \in F_{q+} \backslash F_{q-}$. The edge e has an infinitesimally small residual length in `modified-GW`$(q-)$ and so we can include e in F and also include S in $\mathcal{S}$. The function χ is also modified so that $\chi(S_1) = \chi_{q+}(S_1), \chi(S_2) = \chi_{q+}(S_2)$ and $\chi(S)$ is black. After we have considered all sets in $\mathcal{S}_{q+} \backslash \mathcal{S}_{q-}$, we have that $\mathcal{S} = \mathcal{S}_{q+}$ and for each tree in F there is a corresponding tree in F_{q+} spanning the same set of vertices.

We now consider a sequence of steps that modify the coloring function. Note that χ, χ_{q+} need not be the same. But these functions differ only for sets S for which $\chi_q(S)$ is grey. For such S, $\chi(S) = \chi_{q-}(S)$ is black while $\chi_{q+}(S)$ is white. We consider such sets, one by one, in the order in which they were created in the procedure `modified-GW`$(q+)$. If S is the set under consideration we change $\chi(S)$ to white. This process continues until all such sets have been considered and their colors changed from black to white. Now χ is the same as χ_{q+}.

We now try to reduce the difference in the forests F, F_{q+}. The following claim relates the structure of trees in the forests F and F_{q+}. A set of vertices is *contiguous* in a forest, if the edges of the forest induce a tree on the set of vertices.

CLAIM 4.1. *Let $S \in \mathcal{S}$. Then the vertices of S occur contiguously in the forests F and F_{q+}.*

PROOF. Since the initial potentials $q-$ and $q+$ are only infinitesimally different from q any set that has a positive potential when the algorithm is run with an initial potential q will also have a positive potential when the initial potentials are $q-$ or $q+$. Hence the vertices of this set occur contiguously in the forests F and F_{q+}. □

Since the vertices of $S \in \mathcal{S}$ occur contiguously in forests F and F_{q+}, we can replace the edges induced by S in F with the edges that this set induces in F_{q+}. We perform these modifications on sets $S \in \mathcal{S}$ in the order that they were created in `modified-GW`$(q+)$. Let S be the set under consideration and suppose it was formed by the merging of components $S_1, S_2, \ldots, S_p, S_i \in \mathcal{S}$. To make the modification for the set S we only need to replace the edges that run between the components $S_1, S_2, \ldots, S_p$ in the tree F with the edges between these components in F_{q+}. This replacement is done one edge at a time. Thus we remove one such edge from F and add one such edge of F_{q+} that connects the two trees formed. After having considered all sets in $\mathcal{S}$, the forest F would be identical to F_{q+}. This completes our sequence of steps that takes the tuple $(\mathcal{S}_{q-}, \chi_{q-}, F_{q-})$ to the tuple $(\mathcal{S}_{q+}, \chi_{q+}, F_{q+})$.

After each step in the above process we prune the current forest F with respect to the coloring function χ and determine the size of the largest tree. Clearly, at some step, this quantity will become larger than k. It is this step that is of interest to us for determining the k vertices to pick.

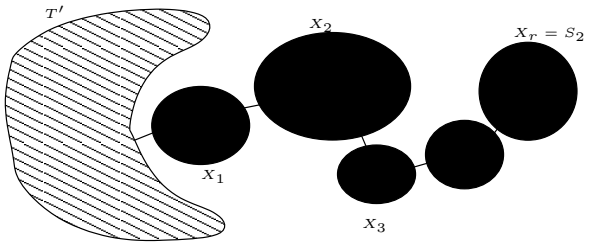

Figure 2: The tree H

4.1 Case I

We first consider the case when the step that causes the pruned forest to contain a tree spanning more than k vertices, is of the first kind i.e. including a set $S \in \mathcal{S}_{q+} \backslash \mathcal{S}_{q-}$ into the collection $\mathcal{S}$ and modifying F and χ suitably.

Let S_1, S_2 be the two sets that merged in `modified-GW`$(q+)$ to form the set S and let e be the edge that got tight and was included in F. Since e is not removed when we prune F with respect to χ, both $\chi(S_1), \chi(S_2)$ are white and this in turn implies that $\chi_q(S_1), \chi_q(S_2)$ are grey. Let H be the tree in the pruned forest that contains edge e. We restrict the sets in $\mathcal{S}$ to vertices of H and show how to pick k contiguous vertices in H.

Set $\chi(S_2)$ to black and prune the tree H. Then edge e would be pruned and as a consequence some other edges of H would also be pruned. It is the black sets due to which these additional edges are pruned that are of interest to us. Let $\mathcal{B} \subset \mathcal{S}$ denote the collection of black sets which cause the edge incident to them to be pruned; note that $S_2 \in \mathcal{B}$. $\mathcal{B}$ is a laminar family and let $X_1, X_2, \ldots, X_r$ be the maximal components in $\mathcal{B}$. After removing the vertices in $X_1, X_2, \ldots, X_r$ from H we would be left with a tree T' with less than k vertices. We call the vertices of T' *old*.

CLAIM 4.2. *The edges of H between the disjoint sets $T', X_1, X_2, \ldots, X_r$ form a path.*

We relabel the vertex sets $X_1, X_2, \ldots, X_r$ so that this is the order in which we encounter these sets when we traverse the path starting from T'; thus $S_2 = X_r$. Since the number of old vertices is less than k we need to pick some other vertices. We do so by picking all vertices in the components $X_1, X_2, \ldots$ in this order until we encounter a component, say X_d, such that picking all vertices of X_d would take us beyond k.

Let $e = (u, v)$ be the edge in H between components X_{d-1} and X_d and let $v \in X_d$. Let k' be the total number of vertices already picked. We now invoke the procedure **pick**$(X_d, v, k - k')$ to pick $k - k'$ contiguous vertices in X_d that include vertex v. Since vertex $u \in X_{d-1}$ has already been picked and (u, v) is and edge in H, including v ensures that the k vertices we pick are contiguous in H.

The procedure **pick**(S, w, l) picks l contiguous vertices, one of which is w, from a component S. Let S_1, S_2 be the two components that merged to form S. Let $e = (u, v)$ be the edge between S_1, S_2 in H with $u \in S_1$ and $v \in S_2$. Further, let $w \in S_1$. If $|S_1| = l$ then we pick all vertices of S_1, relabel vertex u as y and exit the procedure. If $|S_1| > l$ then we need to pick l vertices from S_1 which we do by invoking **pick**(S_1, w, l). If $|S_1| < l$ then we pick all vertices of S_1 and the remaining from S_2. This is done by invoking

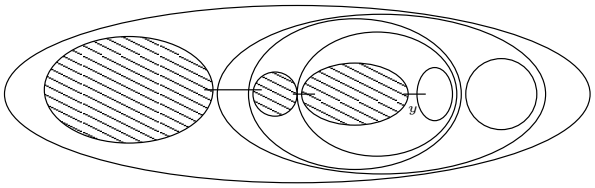

Figure 3: The shaded sets are the vertices picked by procedure pick.

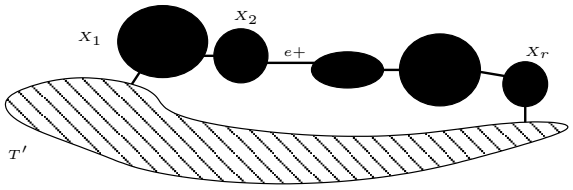

Figure 4: The graph H

pick$(S_2, v, l-|S_1|)$ where the requirement that v be included ensures that the set of vertices picked is contiguous. Note that all components that are picked partially (not all their vertices are included) contain the vertex labeled y.

We call the vertices in $X_1, X_2, \ldots, X_d$ that we have picked, *new* vertices. The remaining vertices in $X_d, X_{d+1}, \ldots, X_r$ are called *discarded* vertices. Thus the total number of old and new vertices is exactly k and the subgraph of H induced over these vertices is the tree, say T, whose length we would like to bound.

4.2 Case II

We now consider the case when, on changing $\chi(S)$ from black to white, the size of the largest tree, in the forest obtained by pruning F with respect to χ, exceeds k. Let H be the tree in the pruned forest containing at least k vertices. We will now show how to pick k contiguous vertices from H. For the rest of this section we restrict the sets in the collection $\mathcal{S}$ to the vertices in H.

We now prune H with respect to χ with $\chi(S)$ as black. Note that there is only one edge in H that is incident to S and this will be pruned. This might, in turn, lead to more edges of H getting pruned. This brings us to the same situation as considered in Case I; the set S now plays the role of S_2.

4.3 Case III

Let T_-, T_+ be the trees differing in one edge and satisfying the property that the largest tree obtained on pruning T_- has less than k vertices while the largest tree obtained on pruning T_+ has more than k vertices. Let e_-, e_+ be the edges that these trees differ in.

Consider the subgraph $T_- \cup \{e_+\}$ and prune it with respect to the coloring χ. Since e_-, e_+ lie on a cycle in $T_- \cup \{e_+\}$, on pruning this subgraph we would get a subgraph, say H, which includes both e_-, e_+. Note that H has at least k vertices. We will show how to pick k contiguous vertices in H and hence in this section we restrict the sets in $\mathcal{S}$ to the vertices of H.

We now prune $H \backslash \{e_+\}$ with respect to the coloring χ. Let $\mathcal{B}$ denote the collection of black sets which cause an edge to be pruned. $\mathcal{B}$ is a laminar family and let $X_1, X_2, \ldots, X_r$ be the inclusion-wise maximal components in $\mathcal{B}$. On removing the vertices in $X_1, X_2, \ldots, X_r$ from H we get a tree, say T'. Note that T' would be one of the trees obtained on pruning T_- and hence T' has less than k vertices.

CLAIM 4.3. *Consider the graph H and the $r+1$ disjoint subsets of vertices $T', X_1, X_2, \ldots, X_r$. If we were to shrink each subset into a single vertex then we would obtain a cycle on $r+1$ vertices.*

We relabel the vertex sets $X_1, X_2, \ldots, X_r$ so that this is the order in which we encounter these sets when we traverse the cycle. All the vertices of T' would be included in the k vertices that we pick; we call these vertices *old*. Since the number of old vertices is less than k we need to pick some other vertices. We do so by picking all vertices in the components $X_1, X_2, \ldots$ in this order till we encounter a component, say X_d, such that picking all vertices of X_d would take us beyond k. The remaining vertices from X_d are picked in the same manner as for Case I.

We call the vertices in $X_1, X_2, \ldots, X_d$ that we have picked, *new* vertices. The remaining vertices in $X_d, X_{d+1}, \ldots, X_r$ are called *discarded* vertices. Thus the total number of old and new vertices is exactly k and the subgraph of H induced over these vertices is the tree, say T, whose length we would like to bound.

5. BOUNDING THE LENGTH OF THE TREE

We first prove that the length of T is at most $2(q*k-\pi(Q))$ where Q is the inclusion wise minimal set in $\mathcal{S}$ containing all vertices of T.

All vertices in $Q - T$ are contained in subsets of Q that are black. Hence all these vertices have exhausted their initial potential before they became part of Q and so $q*k-\pi(Q)$ can be viewed as the total potential expended by the picked (new and old) vertices. We now identify sets S such that $y(S)$ was raised only by expending potential of picked vertices; we call such sets *paid*. Then the total potential expended by picked vertices is the same as the sum of the dual variables of such sets. Clearly, a set which contains only picked vertices is paid. However, if a set contains both picked and discarded vertices then we cannot always include its dual variable in our sum. But if some of the picked vertices of this set are old vertices then the dual variable of this set must have increased by expending the potential of the old vertices. This is because discarded and new vertices would have exhausted their potentials fully before joining any set with old vertices. Hence, sets containing old vertices are paid. Sets which contain only discarded vertices are not of interest to us since no edge of T crosses such sets. It follows, then, that the only sets which are not paid and of interest to us are those which contain no old vertex and contain at least one new and one discarded vertex. We note that every set that contains a new and a discarded vertex also contains the vertex labeled y.

In the following we argue that the extent to which the edges of T get tight in an iteration is at most twice the total increase in the dual variables of the paid sets.

Let t be the time in the growth phase of `modified-GW`(q) when X_d merged with some other active component. At any stage before time t, the component containing vertex y was a leaf in the tree T. All other leaf components of T were also

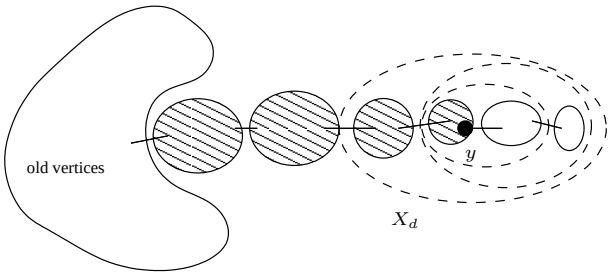

Figure 5: The shaded sets are the new vertices. The sets drawn with dashed lines contain both new and discarded vertices. The black dot is the vertex labeled y

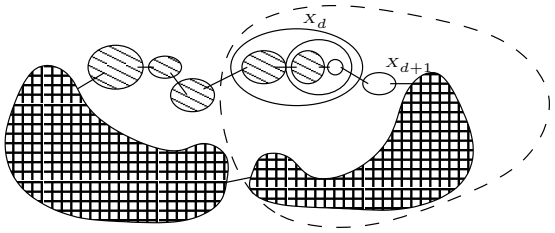

Figure 6: The setting when the edge between X_d, X_{d+1} goes tight. The set formed is shown with the dashed line. The sets shaded with horizontal and vertical lines contain only old vertices while the sets shaded with diagonal lines contain the new vertices.

leaf components in the forest F and so they must be active components. The component containing y may or may not be active. If it is active then it might contain both new and discarded vertices and so we cannot include the increase in the dual variable for this component. But since this is the only component that could either be an inactive leaf or an active component whose dual variable cannot be included in the sum, the Goemans-Williamson analysis continues to hold. We can now claim that for all iterations before time t the total extent to which the edges of T are tightened is at most twice the increase in the dual variables of the paid sets.

At time t, X_d merges with an active set. If the edge between X_{d-1}, X_d became tight at time t then the resulting set contains some old vertices and all of $X_1, X_2, \ldots X_d$. Hence this set is relevant. Since y is now in the same component as some old vertices, at any stage of the growth phase after time t the component containing y will not be an inactive leaf of T.

At time t the edge between X_d and X_{d+1} could also go tight. This, however can happen only in Case III. The new set formed contains some old vertices and all of X_d, $X_{d+1}, \ldots, X_r$; hence this set is relevant. The edges of tree T now induce a cycle on the components. Since only one cycle is formed we can still claim that the total degree, in T, of the active components is at most twice the number of active components and hence even for these iterations the total extent to which the edges of T are tightened is at most twice the increase in the dual variables of the relevant sets.This implies that the length of T is at most $2(qk - \pi(Q))$.

We now show that for any set $Q' \in \mathcal{S}$ that contains all vertices of T, we can find a tree T^* of length at most $2(qk - \pi(Q'))$. Note that $Q \subset Q'$ and the interesting case arises only when $\pi(Q') > \pi(Q)$. Consider the tree in the forest F which contains the vertices of Q'. The vertices of Q' are contiguous in this tree; we consider the restriction of this tree to the vertices of Q' and denote it by T_-.

When pruning T_- with respect to χ we do not consider sets containing Q as a subset. That is, if for some $S \in \mathcal{S}$, $\chi(S)$ is black and S has only one edge $e \in T_-$ incident to it, we do not prune edge e if $Q \subseteq S$. Let H be the tree spanning at least k vertices in the forest obtained after this pruning. We will now pick k contiguous vertices from H so that the length of the tree spanning these vertices is at most $2(q * k - \pi(Q')$. For the remainder of this section we restrict the sets in $\mathcal{S}$ to the vertices of H.

Suppose we now prune H with respect to χ. Let $\mathcal{B} \subset \mathcal{S}$ be the collection of black sets that cause the edge incident to them to be pruned. $\mathcal{B}$ is a laminar family and let $X_1, X_2, \ldots, X_r$ be the maximal components in $\mathcal{B}$. After removing the vertices in $X_1, X_2, \ldots, X_r$ from H we are left with a tree T'. Note that the minimal set in $\mathcal{S}$ that includes all vertices of T' has potential at least $\pi(Q')$. If T' has more than k vertices then, as shown earlier, we can find a subtree of T' of length at most $2(qk - \pi(Q'))$. If $|T'| < k$, we pick k contiguous vertices of H in the same manner as we did in Case I. Let T be the tree obtained and S the minimal set containing all vertices of T. We have argued before that T has length at most $2(q * k - \pi(S))$. Since $\pi(S) \geq \pi(Q')$, the length of T is at most $2(q * k - \pi(Q'))$ and we are done.

6. RUNNING TIME

As mentioned before, the procedures $\mathcal{Q}$,$\mathcal{T}$ have to be run at most n times. It is easy to see that both these procedures are polynomial time. A more accurate analysis of the running time of these procedures is left for the full paper.

7. ACKNOWLEDGMENTS

We acknowledge useful discussions with R. Ravi, Neha Mittal and Saumya Senapati.

8. REFERENCES

[1] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[2] S. Arora and G. Karakostas. A 2 + &egr; approximation algorithm for the k-mst problem. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 754–759. Society for Industrial and Applied Mathematics, 2000.

[3] S. Arya and H. Ramesh. A 2.5-factor approximation algorithm for the k-mst problem. *Information Processing Letters*, 65(3):117–118, 1998.

[4] B. Awerbuch, Y. Azar, A. Blum, and S. Vempala. Improved approximation guarantees for minimum weight k-trees and prize-collecting salesmen. In *Proceedings, ACM Symposium on Theory of Computing*, pages 277–283, 1995.

[5] N. Bansal, A. Blum, S. Chawla, and A. Meyerson. Approximation algorithms for deadline-tsp and vehicle routing with time-windows. In *Proceedings, ACM Symposium on Theory of Computing*, 2004.

[6] A. Blum, P. Chalasani, and S. Vempala. A constant-factor approximation for the k-mst problem in the plane. In *Proceedings, ACM Symposium on Theory of Computing*, pages 294–302, 1995.

[7] A. Blum, S. Chawla, D. Karger, T. Lane, A. Meyerson, and M. Minkoff. Approximation algorithms for orienteering and discounted-reward tsp. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, 2003.

[8] A. Blum, R. Ravi, and S. Vempala. A constant factor approximation for the k-mst problem. In *Proceedings, ACM Symposium on Theory of Computing*, pages 442–448, 1996.

[9] K. Chaudhuri, B. Godfrey, S. Rao, and K. Talwar. Paths, trees and minimum latency tours. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, 2004.

[10] D. Eppstein. Faster geometric k-point mst approximation. Technical Report 13, University of California, Irvine, CA, 1995.

[11] N. Garg. A 3-approximation for the minimum tree spanning k vertices. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, 1996.

[12] N. Garg and D. Hochbaum. An $O(\log k)$ approximation algorithm for the k minimum spanning tree problem in the plane. In *Proceedings, ACM Symposium on Theory of Computing*, 1994.

[13] D. Johnson, M. Minkoff, and S. Phillips. The prize collecting steiner tree problem: theory and practice. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms*, 2000.

[14] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999.

[15] J. S. B. Mitchell, A. Blum, P. Chalasani, and S. Vempala. A constant-factor approximation algorithm for the geometric k-MST problem in the plane. *SIAM Journal on Computing*, 28(3):771–781, 1999.

[16] S. Rajagopalan and V. V. Vazirani. Logarithmic approximation of minimum weight k trees. Unpublished Manuscript, 1995.

[17] R. Ravi, R. Sundaram, M. Marathe, D. Rosenkrantz, and S. Ravi. Spanning trees short and small. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms*, 1993.