Some Approximation Results in Multicasting

Prabhu Manyem
Graduate Program in Operations Research
North Carolina State University
Raleigh, NC 27695-7913, U.S.A.

Matthias F. M. Stallmann
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, U.S.A.

Abstract

The approximability characteristics of the constrained Steiner tree (CST) problem and some if its special cases are considered here. APX is the class of problems for which it is possible to have polynomial time heuristics that guarantee a constant approximation bound. We first show that two special cases of CST, height-constrained spanning tree and height-constrained Steiner tree with unit-weight edges, cannot be in APX. This implies that CST cannot be in APX either. We then show that a more restricted special case of CST, height-constrained spanning tree with edge weights 1 or 2, cannot have a polynomial time approximation scheme unless P=NP.

Key words: Approximability, Constrained Steiner tree, E-reduction, Multicasting.

1 Multicasting

The optimization problems considered here arise in the context of multicast routing in communication networks. Multicasting is the transmission of data from a source to a given set of destinations. Specifically, it involves sending messages to a subset of the nodes in a network, as opposed to broadcasting, where all nodes in the network except the source receive messages.

A positive cost and delay are associated with each link connecting a pair of nodes in the network. The delay at a destination is the sum of the delays in each edge on the path from the source to the destination. In general, minimum delay (at the destinations) and minimum cost cannot be achieved at the same time. The minimum cost solution can be obtained if the problem is formulated as a Steiner tree problem with a cost parameter assigned to the edges. The minimum delay solution is obtained by modeling the problem as a shortest path problem from the source.

1.1 Constrained Steiner Tree and its special cases

The (delay) constrained Steiner tree problem (CST) and its special cases are studied here. See Fig.1. CST can be described as follows: Given an undirected graph $G = (V, E)$, a subset S of V, a vertex $s \in S$ called the *source* (and all vertices in $S - \{s\}$ are called *destinations*), each edge (i, j) having two parameters: a cost $c(i, j)$ and a delay $d(i, j)$, and a positive parameter Δ, the problem is to find a tree connecting vertices in S such that

- the cost of the tree is minimal, and
- for each destination $v \in S - \{s\}$, the sum of the edge delays in the (unique) path from s to v is at most Δ.

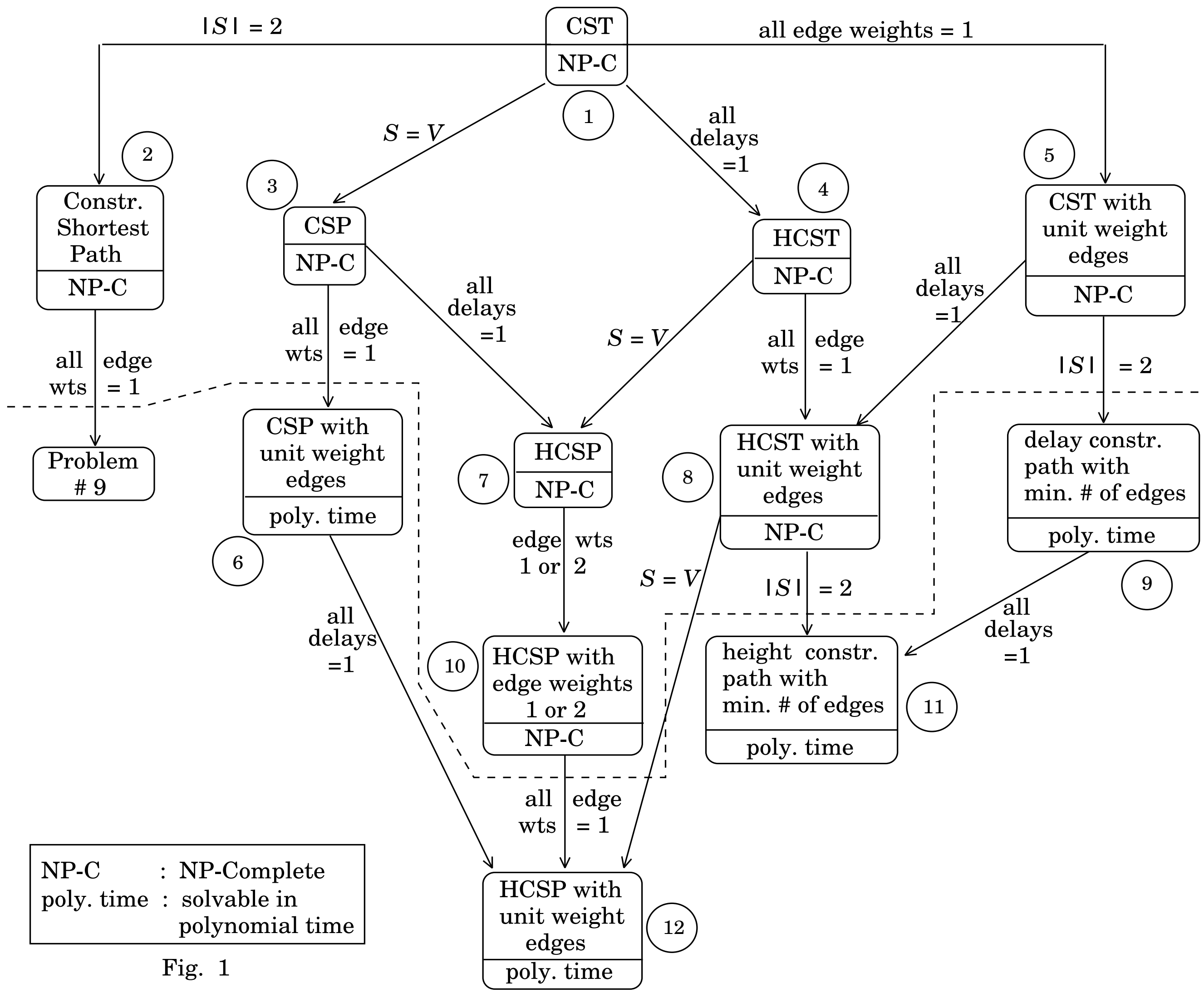
CST
NP-C
|S| = 2
all edge weights = 1
S = V
all delays =1
1
2
3
4
5
Constr. Shortest Path
NP-C
CSP
NP-C
HCST
NP-C
CST with unit weight edges
NP-C
all wts
edge = 1
all delays =1
S = V
all wts
edge = 1
all delays =1
|S| = 2
all wts
edge = 1
Problem # 9
CSP with unit weight edges
poly. time
HCSP
NP-C
HCST with unit weight edges
NP-C
delay constr. path with min. # of edges
poly. time
6
7
8
9
edge 1 or
wts 2
S = V
|S| = 2
all delays =1
all delays =1
10
HCSP with edge weights 1 or 2
NP-C
height constr. path with min. # of edges
poly. time
11
all wts
edge = 1
HCSP with unit weight edges
poly. time
12
NP-C : NP-Complete
poly. time : solvable in polynomial time

Fig. 1

CST and several of its special cases are NP-Complete. Kompella et.al.[7] have given a heuristic for CST. CST and its special cases are shown in Fig.1. The problems above the dotted line are NP-Complete, and the ones below have polynomial time algorithms. We have used the following abbreviations in addition:

CSP	constrained spanning tree
HCSP	height constrained spanning tree
HCSP-12	height constrained spanning tree with edge weights 1 or 2 (all edge weights are not the same)
CST-1	constrained Steiner tree with unit weight edges
HCST	height constrained Steiner tree
HCST-1	height constrained Steiner tree with unit weight edges

Note that for HCSP-12, if all edge weights are equal, then the problem has an efficient solution: breadth first search.

2 Approximation Classes

An NP optimization problem (or NPO problem) is one whose decision version is in NP. All NPO problems considered in this paper will be studied from an approximability point of view. A minimization algorithm M is an ϵ-approximation algorithm if for all x,

$$\frac{c(M(x)) - opt(x)}{c(M(x))} \leq \epsilon$$

where x is an instance of the given problem, $M(x)$ is the solution obtained by M, $c(M(x))$ is the cost of the solution $M(x)$, and $opt(x)$ is the value of an optimal solution to instance x. All costs are assumed to be positive.

The ratio $\frac{c(M(x))}{opt(x)}$ is defined as the approximation ratio for minimization problems.

APX is the class of NPO problems for which it is possible to have polynomial time heuristics that guarantee a constant approximation bound, i.e., the approximation ratio is bounded by a constant for all instances of the problem. In other words, for any problem in APX, an ϵ-approximation algorithm exists for some fixed ϵ less than one. Sahni et.al.[12] have shown that the Travelling salesperson problem is not in APX. Kou et.al.[8] showed that the Steiner tree problem is in APX. HCSP-12 is obviously in APX: any feasible solution has a bound of two on the approximation ratio. CST, CST-1, HCST, and HCST-1 are NP-Complete, since the Steiner tree problem with unit weight edges is NP-Complete[3]. Using reduction from VERTEX COVER, Salama et.al.[13] showed that HCSP-12 (and hence HCSP as well as CSP too) is NP-Complete. Hassin[5] gave an FPTAS (fully polynomial time approximation scheme) for the constrained shortest path problem.

In this paper, we first show that HCSP is not in APX, thereby proving that CSP is not in APX either. Secondly, we show that HCST-1 is not in APX, which also shows that HCST, CST-1, and CST are not in APX. Finally, we show that HCSP-12 cannot have a PTAS (polynomial time approximation scheme) unless P=NP.

3 E-reductions

Ordinary reductions, like the ones used in NP-Completeness proofs, are insufficient from the approximability point of view. Any reduction from a source problem to a target problem does not necessarily imply that the target problem is as hard to approximate as the source problem. The reduction from INDEPENDENT SET to VERTEX COVER is a case in point[10]. The reductions have to be refined to make them *approximation preserving* . For this purpose, Papadimitriou et.al.[11] first came up with *L-reductions*, followed by Khanna et.al.[6] with their *E-reductions*. We shall use E-reductions (E for *error preserving*) in all our proofs. If problem A E-reduces to problem B, then problem B is as hard to approximate as problem A, i.e. E-reductions apply at all levels of approximability. For example, if the source problem has been shown not to have a PTAS unless P=NP, then the same is true for the target problem too.

Definition 1. **(E-reduction)** [6]: A problem A E-reduces to a problem B, or A $\leq_E$ B, if there exist polynomial time functions f, g, and a constant β such that

- f maps an instance I of A to an instance J of B, and
- g maps solutions T of J to solutions S of I such that

$$\varepsilon(I,S) \;\leq\; \beta\varepsilon(J,T), \tag{1}$$

where $\varepsilon(I,S)$ is the *error* term defined below:

Definition 2. **(Error)** [6]: A solution S to an instance I of an NPO problem Π has error $\varepsilon(I,S)$ if

$$\frac{1}{1+\varepsilon(I,S)} \;\leq\; \frac{V(I,S)}{opt(I)} \;\leq\; 1+\varepsilon(I,S), \tag{2}$$

where $V(I,S)$ is the value of a solution S to instance I, and $opt(I)$ is the value of an optimal solution to I. Note that the above definition applies to both maximization and minimization problems.

Lund et.al.[9] showed that SET COVER cannot be guaranteed to have an approximation ratio better than $\theta(\log n)$ unless P=NP, meaning that SET COVER is not in APX. We show in sections 4 and 5 that SET COVER can be E-reduced to HCSP and HCST-1, and thus conclude that these two problems are not in APX. A problem Π is said to be APX-Complete if in addition to Π being in APX, any problem in APX can be E-reduced to Π. Arora et.al.[1] have proved that no APX-Complete problem can have a PTAS unless P=NP. Halldorsson[4] showed that the problem B-SET COVER is APX-Complete. B-SET COVER is a special case of SET COVER, where every given subset has at most b elements, b is a constant greater than or equal to one.

SET COVER:

Given: $X = \{x_i | 1 \leq i \leq p\}$,

$Y = \{y_j | 1 \leq j \leq q\}$, each y_j is a subset of X.

Find : a subset Y' of Y such that

$$\bigcup_{y_j \in Y'} y_j = X$$

Minimize: $|Y'|$, the cardinality of Y', or the number of subsets in Y'.

In B-SET COVER, $|y_j| \leq b,\ 1 \leq j \leq q$, and b is a constant. By showing that B-SET COVER E-reduces to HCSP-12, one can conclude that HCSP-12 cannot have a PTAS unless P=NP. This is what we show in section 6.

4 HCSP is not in APX

We show in this section that SET COVER can be reduced to HCSP (height constrained spanning tree). SET COVER is defined in section 3. In HCSP, a spanning tree needs to be determined whose cost is minimal and whose the number of edges in the path from a vertex $s \in V$ (the *source*) to every other vertex in V is less than or equal to Δ.

Create an instance of HCSP as follows (see fig.2):
For each $x_i \in X$ and $y_j \in Y$ in SET COVER, create a vertex. Create two additional vertices s and t. The edges in E in the instance $G = (V, E)$ of HCSP are:

$E_1 : \ (s, t),$

$E_2 : \ (s, y_j), \ 1 \leq j \leq q,$

$E_3 : \ (t, y_j), \ 1 \leq j \leq q,$

$E_4 : \ (y_j, x_i), \ \text{ if } x_i \in y_j, \ 1 \leq i \leq p, \ 1 \leq j \leq q,$

$E = E_1 \cup E_2 \cup E_3 \cup E_4$

The edge weights are assigned as

$$w(e) = 1, \text{if } e \in E_1 \cup E_3 \cup E_4,$$
$$= \ n = \ p + q + 2, \ \text{if } e \in E_2$$

Let $\Delta =$ height constraint $=$ 2. Note that the distance (the least number of edges for a path between s and any x_i) is two.

4.1 Feasible solution (FS)

As the height constraint is two, all x_i's have to be leaves in any FS. The parents of the x_i's in a FS have to be y_j's, and these y_j's should in turn have s (not t) as their parent. The vertex t may or may not be in the (unique) path from a y_j to s. If it is, then the y_j cannot be the parent of any x_i

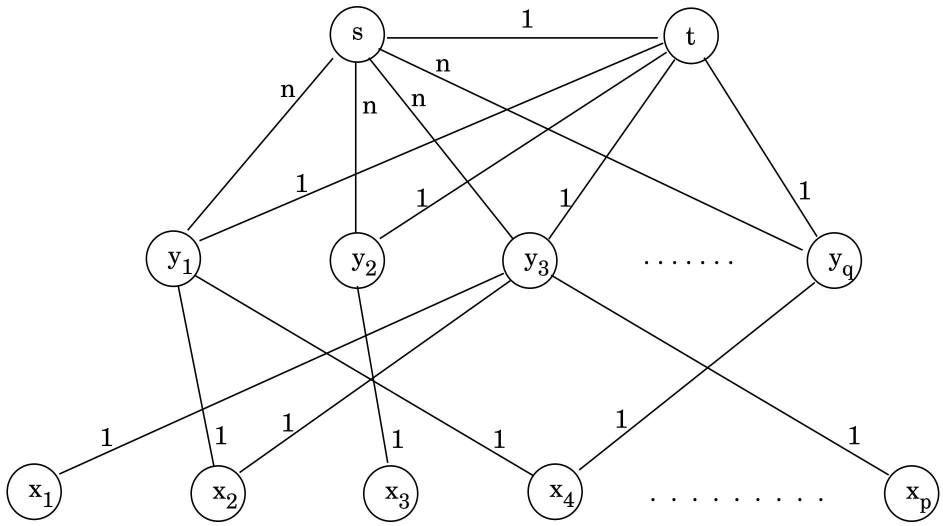

Fig. 2. HCSP instance (reduced from SET COVER)

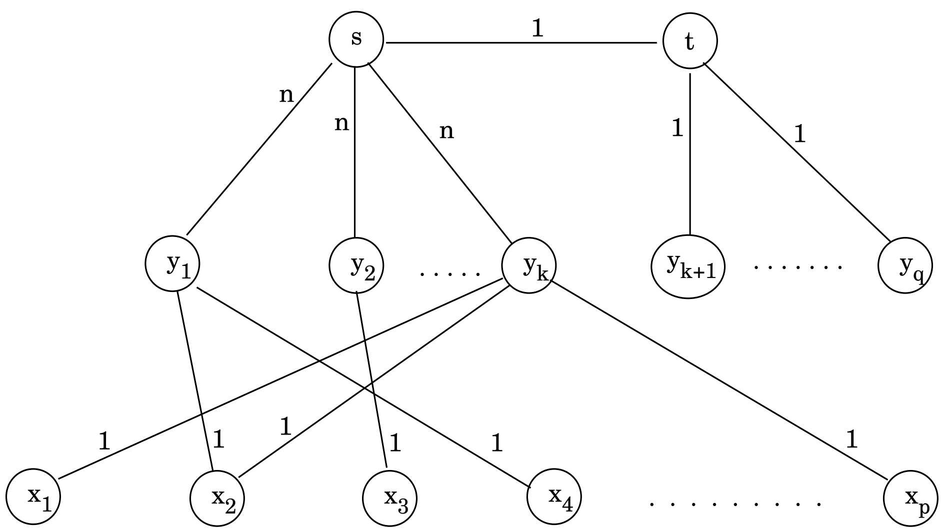

Fig. 3. Feasible solution for our instance of HCSP

due to the height constraint. If not, then it's possible for this y_j to be the parent of some x_i in a FS; suppose we call such a y_j a *covering* y_j and the rest as *non-covering* y_j's. The covering y_j's together form a *cover* to the x_i's — these y_j's may or may not be leaves in a FS. The non-covering y_j's will be leaves.

Let the cover size (the number of covering y_j's) be k. If in fig.2, we move the cover to the left (the y_j's can be renumbered in such a way that y_1 through y_k cover all elements in X), a feasible solution as described above will look like the one in fig.3.

The cost of the spanning tree of fig.3 is

$$\begin{aligned} &= kn + (q-k) + 1 + p \\ &= k(n-1) + p + q + 1 \\ &= k(n-1) + n - 1 \\ &= (n-1)(k+1) \end{aligned}$$

The cost then, can be described as a function $C(k)$ of k, where

$$C(k) = (n-1)(k+1)$$

Note that by construction, there is a one-to-one correspondence between feasible solutions in SET COVER and our instance of HCSP: a set of covering y_j's in our HCSP instance can also be used as a cover in SET COVER. In the other direction, a cover in SET COVER can be transformed to a set of covering y_j's in our HCSP instance; these will have s as their parent in a FS. The (other non-covering) y_j's will have t as their parent, and the x_i's will have the covering y_j's as their parent(s).

Further note that the reduction from SET COVER is in polynomial time in the size of SET COVER. To complete the proof that this is an E-reduction, we only need to show that the error condition (1) is satisfied for some constant β.

For this reduction, I is a SET COVER instance, S is a solution to I, J is our instance of HCSP corresponding to I, and T is a solution to J. Let k be the value of any feasible solution to SET COVER, and l be that of the optimal solution. Therefore,

$$\varepsilon(I,S) \;=\; \frac{k}{l} - 1 \;=\; \frac{k-l}{l}$$

and

$$\varepsilon(J,T) \;=\; \frac{C(k)}{C(l)} - 1 \;=\; \frac{(k+1)(n-1)}{(l+1)(n-1)} - 1 \;=\; \frac{k-l}{l+1}$$

We need to find a constant β such that

$$\beta\varepsilon(I,S) \;\geq\; \varepsilon(I,S)$$

or

$$\beta\frac{k-l}{l+1} \;\geq\; \frac{k-l}{l}$$

or

$$\beta \;\geq\; 1 + \frac{1}{l}$$

As $l \geq 1$, choose $\beta = 2$. This completes the proof of E-reduction.

5 HCST-1 is not in APX

This proof is similar to the one for HCSP. The vertex t is not necessary — it can be deleted. In edge set E_2, replace each edge (s, y_j) by a path of n edges ($n = p + q + 1$), each with unit weight, between s and y_j. See fig.4. Set the height constraint Δ to $n+1$. Let S (the set of vertices that need to be connected) be $s \cup X$.

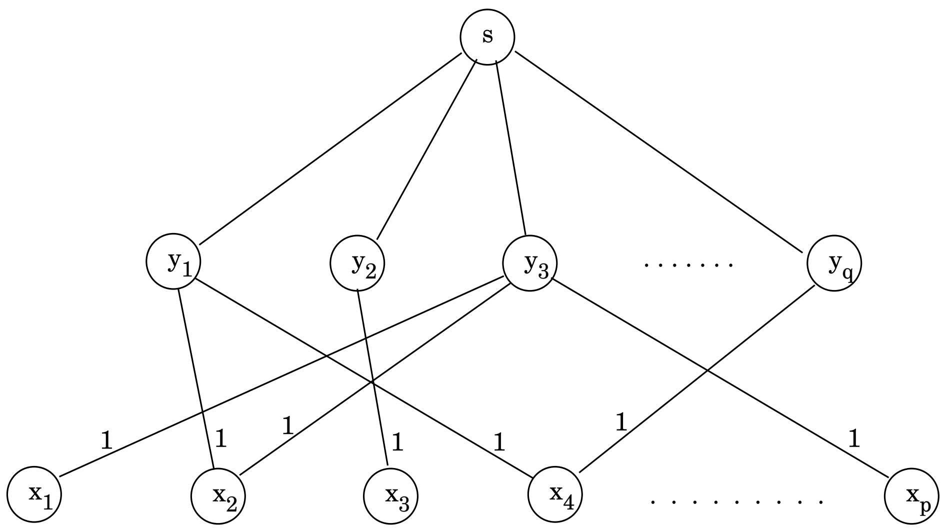

Fig. 4. HCST-1 instance (reduced from SET COVER)
Each edge (s, y_j) shown above is in fact a path of n edges

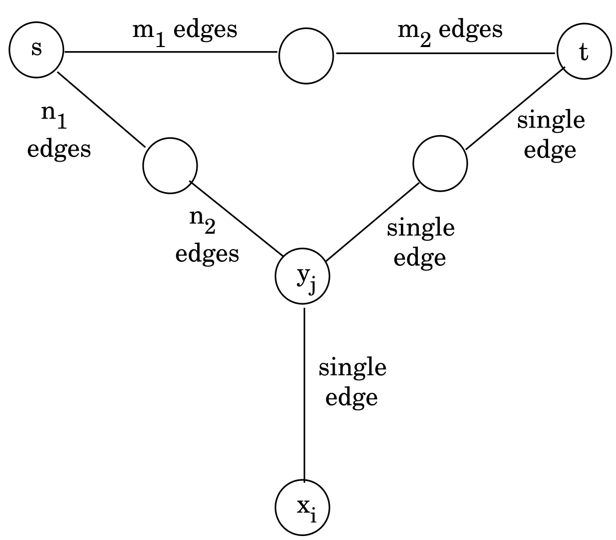

Fig. 5. HCSP-12 instance (reduced from B-SET COVER)

5.1 Feasible solution (FS)

But for a few changes, the FS here is the same as in section 4. Only the covering y_j's need to appear in a FS (which is a Steiner tree) ; the rest of the y_j's (the non-covering ones) need not.

If k is the size of the cover, then the cost of the Steiner tree is $kn + p$. As before, it is easy to see that a cover of size k in our HCST-1 instance of HCST-1 corresponds to a k-size cover in SET COVER and vice-versa. This is a polynomial time $O(n^2)$ reduction. The error condition (1) is satisfied as shown below:

$$\varepsilon(I, S) = \frac{k}{l} - 1 = \frac{k-l}{l}$$

and

$$\varepsilon(J, T) = \frac{kn+p}{ln+p} - 1 = \frac{n(k-l)}{ln+p}$$

We need a β where

$$\beta\varepsilon(I, S) \geq \varepsilon(I, S)$$

or

$$\beta\frac{n(k-l)}{ln+p} \geq \frac{k-l}{l}$$

or

$$\beta \geq 1 + \frac{p}{ln}$$

Since $l \geq 1$ and $p \leq n$, the right side of the above inequality is at most two. Choose β equal to 2. This completes the proof.

6 HCSP-12 is harder than B-SET COVER

It is shown in this section that HCSP-12, the height constrained spanning tree problem with edge weights 1 or 2 (all edge weights not the same) cannot have a PTAS unless P=NP. The proof technique is similar to the ones used in the previous sections: we E-reduce B-SET COVER to HCSP-12. Recall the definition of B-SET COVER from section 3. We choose $b = 3$ for this reduction (if $b = 2$, then B-SET COVER reduces to EDGE COVER, which is polynomially solvable).

The instance of HCSP-12 we will create here is very similar to the HCSP instance created from SET COVER in section 4: Replace the edge (s, t) by a path $P(s, t)$ consisting of m_1 edges of unit weight and m_2 edges of weight 2. For $1 \leq j \leq q$, replace the edge (s, y_j) by a path $P(s, y_j)$ with n_1 unit-weighted edges and n_2 edges of weight 2. Further, for $1 \leq j \leq q$, replace the edge (t, y_j) by a path $P(t, y_j)$ with two edges, each with unit weight. See fig.5. Let

$$\begin{aligned} l_1 &= \text{length of } P(s, y_j), \quad 1 \leq j \leq q, \\ l_2 &= \text{length of } P(t, y_j), \quad 1 \leq j \leq q, \\ l_3 &= \text{path length from } s \text{ to } t, \\ l_4 &= \text{length of } P(x_i, y_j), \quad 1 <= j <= q, \quad 1 <= i <= p, \\ & \qquad\qquad\qquad\qquad\qquad \text{for } (x_i, y_j) \in E_4 \end{aligned}$$

This gives

$$\begin{aligned} l_1 &= n_1 + n_2, \\ l_2 &= 2, \\ l_3 &= m_1 + m_2, \text{ and} \\ l_4 &= 1 \end{aligned} \tag{3}$$

Let W_i be the weight of the path corresponding to l_i, $1 \leq i \leq 4$. Thus,

$$\begin{aligned} W_1 &= n_1 + 2n_2, \\ W_2 &= 2, \\ W_3 &= m_1 + 2m_2, \text{ and} \\ W_4 &= 1 \end{aligned} \tag{4}$$

Finally, set the height constraint Δ as

$$\Delta = l_1 + l_4 \tag{5}$$

6.1 Feasible solution (FS)

The HCSP-12 instance can be constructed such that

$$\begin{aligned} l_i &\leq \Delta,\ i = 1, 2, 3, 4, & (6) \\ l_1 + l_2 &> \Delta, \text{ and} & (7) \\ l_2 + l_3 &= \Delta & (8) \end{aligned}$$

From (6) and (7), it follows that the path $P(s, t)$ will be a part of any FS. Hence for any y_j, exactly one of $P(s, y_j)$ and $P(t, y_j)$ will be in a FS. Consider the two cases:

- If $P(t, y_j)$ is in the FS, then from (8) it follows that there cannot be an $x_i \in X$ such that $P(x_i, y_j)$ is in the FS. Such a y_j is a non-covering y_j.
- On the other hand, if $P(s, y_j)$ is in the FS, for any $x_i \in X$, it is possible for $P(x_i, y_j)$ to be in the FS as long as such a path $P(x_i, y_j)$ exists in the given graph, since Δ is equal to $l_1 + l_4$. Such a y_j is a covering y_j and all such y_j's form a cover to X.

Since each subset y_j has at most 3 elements in 3-SET COVER, for a feasible solution to our HCSP instance, we can assume that

$$p \leq 3q \tag{9}$$

Suppose we let

$$\begin{aligned} m_1 &= 6q - p, \\ m_2 &= 0, \\ n_1 &= 6q - 2p, \\ n_2 &= p + 1 \end{aligned} \tag{10}$$

(3) and (10) give

$$\begin{aligned} l_1 &= 6q - p + 1, \\ W_1 &= 6q + 2, \\ l_3 &= 6q - p, \\ W_3 &= 6q - p \end{aligned} \tag{11}$$

(3) and (11) satisfy (6), (7), and (8). (9) satisfies the natural requirement that m_1 and n_1 be non-negative. W, the weight of a FS, is

$$\begin{aligned} W &= (k)W_1 + (q-k)W_2 + W_3 + pW_4 \\ &= q(6k+8) \end{aligned}$$

W is $q(6k+8)$ if and only if the size of the cover to X is k. As in the previous sections, a k-size cover in our HCSP-12 instance corresponds to a k-size cover in B-SET COVER and vice-versa. The reduction from B-SET COVER to HCSP-12 is easily seen to be polynomial in the size of B-SET COVER. The proof for a constant bound in the error in along the same lines as before:

$$\varepsilon(I,S) = \frac{k}{l} - 1 = \frac{k-l}{l}$$

and

$$\varepsilon(J,T) = \frac{q(6k+8)}{q(6l+8)} - 1 = \frac{6(k-l)}{6l+8}$$

We need a β where

$$\beta\varepsilon(I,S) \geq \varepsilon(I,S)$$

or

$$\beta\frac{q(6k+8)}{q(6l+8)} \geq \frac{k-l}{l}$$

or

$$\beta \geq 1 + \frac{4}{3l}$$

Since $l \geq 1$, choose β equal to 3. This completes the proof of E-reduction. It follows that HCSP-12 cannot have a PTAS unless P=NP.

7 Conclusion

We have shown that all problems in fig.1 except problems 2 and 8 are not in APX. Problem(8) cannot have a PTAS unless P=NP; it would be interesting to determine if it can have a constant approximation that is strictly less than two.

8 Acknowledgements

We thank Hussein Salama for his useful suggestions and comments. The compendium of NP-optimization problems[2] has been a big help in our research.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. *Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 14–23, 1992.

[2] P. Crescenzi and V. Kann. *A Compendium OF NP Optimization Problems.* Dynamic online survey available by anonymous ftp at `nada.kth.se` as `Theory/Viggo-Kann/compendium.ps.Z` , 1995.

[3] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman(New York), 1979.

[4] M.M. Halldorsson. Approximating set cover via local improvements. *Technical Report IS-RR-95-0002F, School of Information Science, Japan Advanced Institute of Science and Technology, Hokuriku*, 1995.

[5] R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1), February 1992.

[6] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 819–830, 1994.

[7] V.P. Kompella, J.C. Pasquale, and G.C. Polyzos. Multicast routing for multimedia communication. *IEEE Transactions on Networking*, pages 286–292, 1993.

[8] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner trees. *Acta Informatica*, 15:141–145, 1981.

[9] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Proc. of the 25th ACM Symposium on the Theory of Computing*, pages 286–295, 1993.

[10] C.H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.

[11] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Proc. of the 20th ACM Symposium on the Theory of Computing*, pages 229–234, 1988.

[12] S. Sahni and T. Gonzalez. P-complete approximation problems. *J. of the Association for Computing Machinery*, 23:555–565, 1976.

[13] H.F. Salama, Y. Viniotis, and D.S. Reeves. An Efficient Delay-Constrained Minimum Spanning Tree Heuristic. Submitted to the *Fifth*

International Conference on Computer Communications and Networks, October 1996.