Reduction Tests for the Prize-Collecting Steiner Problem

Eduardo Uchoa

August, 2004

Abstract

The Prize-Collecting Steiner Problem (PCSP) is a generalization of the classical Steiner Problem in Graphs (SPG) where instead of terminal vertices that must be necessarily connected, one have profits associated to the vertices that must be balanced against the connection costs. This problem is gaining much attention in the last years due to its practical applications on network design. This article introduces a proper redefinition of the concept of bottleneck Steiner distance on the PCSP context. This allows the application of known STP reduction tests on the PCSP in their full power.

1 Introduction

The *Prize-Collecting Steiner Problem* (PCSP) consists of given a connected graph $G = (V, E)$, positive edge costs $c(e)$ (also denoted by $c(u, v)$ when $e = (u, v)$), and non-negative vertex profits $p(i)$; find a subtree (V', E') of G maximizing $\sum_{e \in V'} p(i) - \sum_{e \in E'} c(e)$. This quantity is certainly non-negative, for the empty subtree gives a valid answer. The vertices with positive profits, those in the set $T = \{i \in V \,|\, p(i) > 0\}$, are called terminals. The Steiner Problem in Graphs (SPG) can be viewed as the particular case of the PCSP where terminal profits are high enough to assure that all of them must be any optimal solution.

The practical efficiency of exact algorithms for the STP increased dramatically in recent years. The sequence of results reported in [11, 13, 14, 15, 10] on the main benchmark instances from the literature [7] improve by two or three orders of magnitude upon the results from previous articles, like [6]. The strength of those new algorithms lies on a complex combination of reduction tests, primal heuristics, dual heuristics and branch-and-cut. Reduction tests are a key part of those combos.

The first reduction tests for the STP were introduced by Beasley [2] and by Balakrishnan and Patel [1]. They were generalized by Duin and Volgenant [5] in a work that established a set of "classical" reduction tests widely used in the 1990's decade. The most important such tests are based on the concept of bottleneck Steiner distances. More recently, Uchoa et al. [15] enhanced Duin and Volgenant's tests with the idea of expansion. This idea was further developed by Polzin and Vahdati [12]. Anyway, those new reduction tests with expansion still rely on Bottleneck Steiner distances.

Current exact algorithms for the PCSP are not as advanced as their STP counterparts. This is statement is certainly true when one considers reduction tests. Recent algorithms like [9], [3] and [8] apply weakened versions of Duin and Volgenant's tests obtained by replacing the bottleneck Steiner distances by standard distances. This weakening makes those tests, originally devised for the STP, valid for the PCSP, but makes them much less effective.

This article proposes a new definition of bottleneck Steiner distances on the PCSP context. This redefinition allows the application of Duin and Volgenant's tests (and their enhancements with the idea of expansion) in their full power on the PCSP, just like they are now applied on the STP. There is only one important difference. Computing exact bottleneck distances on the SPG can be done in polynomial time. On the other hand, it is shown here that computing bottleneck distances on the PCSP is NP-hard. This point should not hinder the practical use of the new tests. Most current STP codes only use fast and efficient heuristics to compute bottleneck distances. Similar heuristics can also be used on the PCSP.

2 Reduction tests for the the STP

Reduction tests are procedures devised to transform an original instance into a smaller equivalent instance. An edge $e \in E$ is said to be *choosable* if there is at least one optimal solution containing e and *redundant* if there is at least one optimal solution not containing e. Some reduction tests try to identify choosable and redundant edges. Once a choosable edge $e = (u, v)$ is identified, it can be forced into the solution and its endpoints u and v may be contracted. A redundant edge is simply deleted from the graph. In either case, the size of the instance is reduced. Other reduction tests lead to more complex graph transformations. Reduction tests may be successively applied to already reduced graphs, until no further reduction is possible. Some very simple tests are:

Test 1 Non-terminal with degree 1 (NTD1) - *A non-terminal vertex with degree 1 and its adjacent edge may be deleted.*

Test 2 Non-terminal with degree 2 (NTD2) - *A non-terminal vertex u with degree 2 and its adjacent edges (u, v) and (u, w) may be replaced by a single edge (v, w) with cost $c(u, v) + c(u, w)$.*

Test 3 Terminal with degree 1 (TD1) - *If $|T| \geq 2$, the edge adjacent to a terminal vertex with degree 1 is choosable.*

In order to introduce more complex tests, we review the definition of bottleneck Steiner distance on the SPG context, as introduced by Duin and Volgenant [5].

Let u and v be two distinct vertices in V. Let $\mathcal{P}(u, v)$ denote the set of all simple paths joining u to v. The standard *distance* between vertices u and v is defined as

$$d(u, v) = \min\{c(P) \mid P \in \mathcal{P}(u, v)\}, \tag{1}$$

where $c(P)$ denotes the sum of the costs of the edges in path P.

For $P \in \mathcal{P}(u, v)$, let $T(P)$ be $\{u, v\} \cup (T \cap P)$, i.e. the vertex-set formed by u, v and the terminals in P. Two vertices x and y in $T(P)$ are said to be *consecutive* if the subpath from x to y in P contains no other vertices in $T(P)$. The *Steiner distance* $SD(P)$ is the length of the longest subpath in P joining two consecutive vertices in $T(P)$. The *bottleneck Steiner distance* between vertices u and v is defined as

$$B(u, v) = \min\{SD(P) \mid P \in \mathcal{P}(u, v)\}. \tag{2}$$

A nice interpretation for bottleneck distances (that appears, for instance in [4, 6]) is to consider G as a road system, T as petrol stations and a driver who wants to go from u to v. Then $B(u, v)$ is the minimum distance he must be able to drive without refilling in order to reach his destination.

The bottleneck Steiner distance without passing through a given edge e is defined as

$$B(u, v)^{-e} = \min\{SD(P) \mid P \in \mathcal{P}(u, v);\ e \notin P\}. \tag{3}$$

If e disconnects u and v, $B(u, v)^{-e} = \infty$.

The key concept of bottleneck distance allowed Duin and Volgenant [5] to propose stronger reduction tests, generalizing some ideas which appeared earlier in the literature [2, 1]. The most important such tests, in terms of graph reductions obtained, are the SD and the NTD-3 tests.

Test 4 Special distance (SD) - *Let (u, v) be an edge in E. If $B(u, v)^{-(u,v)} \leq c(u, v)$, then edge (u, v) can be removed.*

Test 5 Non-terminal degree 3 (NTD-3) - *Let u be a non-terminal vertex with degree 3, adjacent to vertices v, w, and z. If*

$$\min\{B(v,w)+B(v,z), B(w,v)+B(w,z), B(z,v)+B(z,w)\} \leq c(u,v)+c(u,w)+c(u,z),$$

then there is an optimal solution where the degree of u is at most 2. Therefore, u and its three adjacent edges can be replaced by the following three edges: (v,w) with cost $c(u,v)+c(u,w)$, (v,z) with cost $c(u,v)+c(u,z)$, and (w,z) with cost $c(u,w)+c(u,z)$.

The graph transformation given by the NTD3 test is not much advantageous by itself, since the number of edges remains the same. But some of the newly created edges are quite likely to be immediately eliminated by the SD test. Duin and Volgenant actually introduced a general test NTD-k for non-terminals with any degree k, based on bottleneck distances. This test is not much effective for values of $k > 3$.

There is also another test proposed by Duin and Volgenant that not uses bottleneck distances.

Test 6 Terminal distance (TDist) - *Let $[W, \bar{W}]$ be a partition of the vertices in V such that the subgraphs induced by W and $\bar{W}$ are both connected, with $W \cap T \neq \emptyset$ and $\bar{W} \cap T \neq \emptyset$. Let $\delta(W)$ be the cut induced by this partition. If $\delta(W) = \{e\}$, then e is choosable. If $|\delta(W)| \geq 2$, let $e = \text{argmin}_{e' \in \delta(W)} c(e')$ and $f = \text{argmin}_{f' \in \delta(W) \setminus \{e\}} c(f')$ be respectively a shortest and a second shortest edge in the cut. Suppose $e = (u,v)$ with $u \in W$ and $v \in \bar{W}$. If*

$$\min\{d(u,t_1) \,|\, t_1 \in T \cap W\} + c(e) + \min\{d(v,t_2) \,|\, t_2 \in T \cap \bar{W}\} \;\leq\; c(f),$$

then e is choosable.

The new tests introduced in Uchoa et al. [14, 15] are enhancements of the SD and the NTD-k tests with the idea of expansion. Loosely speaking, this idea means probing the instance to dynamically build a chain of logical implications of the kind "if edge e appear in some optimal solution R then edge f must also be in R" in order to prove that some graph reduction can be indeed performed. Such probing is heavily based on the concept of bottleneck distances.

3 Reduction tests for the PCSP

Consider a PCSP instance. Again, let u and v be two distinct vertices in V and let $\mathcal{P}(u,v)$ denote the set of all simple paths joining u to v. Consider a path $P \in \mathcal{P}(u,v)$. Define $P(x,y)$ as the subpath of P between two given vertices x and y in P. Define the *Steiner distance* associated to this subpath as

$$SD(P(x,y)) = \sum_{e \in P(x,y)} c(e) - \sum_{i \in P(x,y) \setminus \{x,y\}} p(i),$$

i.e. the sum of the edge costs in this subpath, minus the sum of the profits of vertices in the interior of this subpath. The *Steiner distance* associated to the whole path P is:

$$SD(P) = max_{x,y \in P} SD(P(x,y))$$

Finally, the *bottleneck Steiner distance* between vertices u and v is defined as

$$B(u,v) = \min\{SD(P) \mid P \in \mathcal{P}(u,v)\}. \tag{4}$$

The new definition of bottleneck distances for the PCSP can be interpreted as follows. Consider G as a road system, $c(i,j)$ as the number of units of fuel needed to drive from i to j and the vertices as petrol stations that have only $p(i)$ units of fuel available. Suppose that a driver wants to go from u to v, starting with a full tank. Then $B(u,v)$ is the minimum tank capacity (in units of fuel) necessary to reach v. In the SPG case, terminals can be viewed as petrol stations with many units of fuel in stock, so the vehicle tank is always completely refilled.

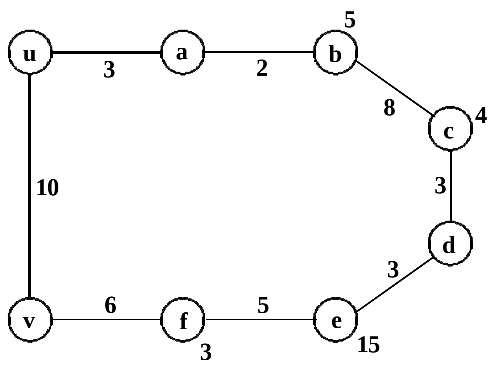

Figure 1: Part of a PCSP instance. Edge (u,v) can be eliminated by the SD test.

Figure 1 depicts part of a PCSP instance, where edge costs are indicated and vertex profits are the following, $p(a) = 0, p(b) = 5, p(c) = 4, p(d) = 0, p(e) = 15, p(f) = 3$. The Steiner distance associated to path $\{u,a,b,c,d,e,f,v\}$ is 10, the same of the subpath $\{b,c,d,e\}$. Going back to the vehicle analogy, we can say that a tank capacity of 10 units is enough to travel from u to v by that path. The vehicle starts at u with a full tank, reaches b with 5 units left, completes the tank with more 5 units, reaches c with 2 units left, puts more 4 units, reaches e with the tank empty, fill it with 10 units (from the 15 available), reaches f with 5 units left, puts more 3 units and finally arrives at v with 2 units left.

Theorem 1 *Test SD with the new definition of bottleneck Steiner distance is valid for the PCSP.*

Proof: Suppose that $B(u,v)^{-(u,v)} \leq c(u,v)$. Let $P \in \mathcal{P}(u,v)$ be a path not using edge (u,v) such that $SD(P) = B(u,v)$. Let R be a solution tree using edge (u,v). Removing (u,v) from R creates two subtrees. Let R_u be the one containing vertex u and R_v be the other, containing v. Pick two vertices x and y from P such that $x \in R_u$, $y \in R_v$ but no other vertices in $P(x,y)$ are in R. Since $SD(P(x,y)) \leq B(u,v)^{-(u,v)} \leq c(u,v)$, then $R' = R_u \cup R_v \cup P(x,y)$ is a solution tree at least as good as R. Therefore, there is an optimal solution that does not use (u,v). ■

Tests NTD-1 and NTD-2 are clearly valid for the PCSP.

Theorem 2 *Test NTD-3 with the new definition of bottleneck Steiner distance is valid for the PCSP.*

Proof: Without loss of generality, suppose that $B(v,w) + B(v,z) \leq c(u,v) + c(u,w) + c(u,z)$, since the other two cases are similar. Let $P_1 \in \mathcal{P}(v,w)$ be a path such that $SD(P_1) = B(v,w)$ and $P_2 \in \mathcal{P}(v,z)$ be a path such that $SD(P_2) = B(v,z)$. Let R be a solution tree using vertex u with degree 3. Removing u and its adjacent edges from R, we obtain the subtrees R_v, R_w, and R_z. Pick two vertices x_1 and y_1 from P_1 such that $x_1 \in R_w$, $y_1 \in R_v$ but no other vertices in the subpath $P_1(x_1,y_1)$ are in R. Similarly, pick two vertices x_2 and y_2 from P_2 such that $x_2 \in R_z$, $y_2 \in R_v$ but no other vertices in the subpath $P_2(x_2,y_2)$ are in R. Since $SD(P_1(x_1,y_1)) \leq B(v,w)$ and $SD(P_2(x_2,y_2)) \leq B(v,z)$, $SD(P_1(x_1,y_1)) + SD(P_2(x_2,y_2)) \leq c(u,v) + c(u,w) + c(u,z)$. Therefore $R' = R_v \cup R_w \cup R_z \cup P_1(x_1,y_1) \cup P_2(x_2,y_2)$ is a solution tree at least as good as R. Vertex u has degree at most 2 in R' (otherwise $B(v,w) + B(v,z) > c(u,v) + c(u,w) + c(u,z)$). ■

The applicability of tests NTD-k can be increased by considering as non-terminals not only vertices with zero profit. One can consider a vertex u with positive profit as a "non-terminal" if it can be shown that u would never appear as a leaf in some optimal solution. For instance, if $p(u)$ is less or equal to the cost of the cheapest edge adjacent to u. The graph transformations produced by the tests must be slightly changed when $p(u) > 0$. On test NTD-2, (u,v) and (u,w) must be replaced by an edge (v,w) with cost $c(u,v) + c(u,w) - p(u)$. A similar change must be done on NTD-3.

The new tests with expansion can also be applied on the PCSP with the new definitions of bottleneck distances and "non-terminals". We do not prove this here, but it is quite easy to adapt the proofs found in [15] to this new context.

Test TD-1 can be applied on the PCSP if one consider a vertex u as a "terminal" if it can be shown that u appears in some optimal solution with more than two vertices. On test TDist, it is necessary to show that both vertices t_1 and t_2 appear in some optimal solution. Those two tests are not able to yield significant graph reductions on most STP instances. It is unlikely that they would be effective on the PCSP with the additional complication of having to identify "terminals". In fact, except in the cases where some vertices have profits really high (with respect to edge costs), this would be very hard to do.

4 Computing Bottleneck Steiner Distances

A table with the exact bottleneck Steiner distances for all pairs of vertices in a STP instance can be computed in $O(|V|^3)$ time [4]. Computing exact bottleneck Steiner distances on a PCSP instance can be much harder. Define this problem in a more formal way.

Prize-Collecting Bottleneck Distance

Instance: Graph $G = (V, E)$, positive integers c associated to the edges, non-negative integers p associated to vertices, vertices u, v in V and integer b.

Question: Is $B(u, v) \leq b$?

Theorem 3 *Prize-Collecting Bottleneck Distance is NP-hard.*

Proof: The Hamiltonian path problem, find a simple path visiting all vertices in a graph, is widely known to be NP-hard. The following version of the problem, find a simple path between two given vertices visiting all the other vertices in a graph, is easily shown to be NP-hard too. This version is formally defined as follows.

Hamiltonian Path

Instance: Graph $G' = (V', E')$, vertices u', v' in V.

Question: Is there a simple path from u' to v' visiting all the other vertices in G' ?

Given an instance of Hamiltonian Path, produce an instance of Prize-Collecting Bottleneck Distance as follows. Graph $G = (V' \cup \{u, v\}, E' \cup \{(u, u'), (v, v')\})$. Costs are 1 for the edges in E', $c(u, u') = c(v, v') = |V'|$. The profit of all vertices are 2, except for u', $p(u') = 1$. Define b as equal to $|V'|$. It can be seen that $B(u, v) = b$ on that instance if and only if there is a hamiltonian path from u' to v' on the original instance. ■

The above theorem rules out the computation of exact bottleneck distances on reduction tests for the PCSP. One must use heuristics instead. Such kind of heuristics are widely used when applying reduction test on STP, see [4, 6, 14, 10], since the $O(n^3)$ time for exact computation is considered excessive. Those heuristics are fast and very effective, they yield upper bounds on the true bottleneck distances so tight that the amount of graph reduction obtained by the tests barely changes. This is possible because the tests (even those with expansion) always ask for distances between vertices that are very close in the graph. Only a few terminals in that neighborhood are likely to be relevant in that computation. Similar ideas can be used to compute upper bounds for bottleneck distances on the PCSP.

5 Conclusion

This work have proposed a redefinition of the concept of bottleneck Steiner distances on the PCSP context. The reduction tests that are more effective on the STP, the SD and the NTD-k (and their enhancements with expansion), can then be applied on the PCSP in their full power.

The practical performance of the new tests on a set of benchmark instances still have to be assessed, it basically depends on the implementation of good heuristics to calculate the new bottleneck distances. It is worthy to mention that the practical applications mentioned in Lucena and Resende [9] (telecommunications network design) and in Ljubić et al. [8] (electricity and gas distribution) lead to instances where most of the vertices have positive profits. On those instances, bottleneck Steiner instances may be significantly smaller than standard distances, leading to large graph reductions.

References

[1] A. Balakrishnan and N. Patel, "Problem reduction methods and a tree generation algorithm for the Steiner network problem", *Networks* 17, 65–85, 1987.

[2] J. Beasley, "An algorithm for the Steiner problem in graphs", *Networks* 14, 147–159, 1984.

[3] A. Cunha, A. Lucena, N. Maculan and M. Resende, "A relax and cut algorithm for the prize collecting Steiner problem in graphs", Proceedings of Mathematical Programming in Rio, 72–78, Bzios, Brazil, 2003.

[4] C. Duin, "Steiner's problem in graphs", PhD thesis, University of Amsterdam, 1993.

[5] C. Duin and A. Volgenant, "Reduction tests for the Steiner problem in graphs", *Networks* 19, 549–567, 1989.

[6] T. Koch and A. Martin, "Solving Steiner tree problems in graphs to optimality", *Networks* 32, 207–232, 1998.

[7] T. Koch, A. Martin and S. Voß, "SteinLib: An updated library on Steiner tree problems in graphs", Konrad-Zuse-Zentrum für Informationstechnik Berlin, ZIB-Report 00-37, 2000 (online document at `http://elib.zib.de/steinlib`).

[8] I. Ljubić, R. Weiskircher, U. Pferschy, G. Klau, P. Mutzel and M. Fischetti, "Solving the prize-collecting Steiner tree problem ro optimality", Report TR-186-1-04-01, Technische Universität Wien, May 2004.

[9] A. Lucena and M. Resende, "Strong lower bounds for the prize collecting Steiner problem in graphs", *Discrete Applied Mathematics* 141, 277–294, 2004.

[10] T. Polzin, "Algorithms for the Steiner Problem in Networks", PhD thesis, Universität des Saarlandes, 2003.

[11] T. Polzin and S. Vahdati, "Improved algorithms for the Steiner problem in networks", *Discrete Applied Mathematics* 112, 263–300, 2001.

[12] T. Polzin and S. Vahdati, " Extending Reduction Techniques for the Steiner Tree Problem", Proceedings of the ESA 2002, 795–807, 2002.

[13] M. Poggi de Aragão, E. Uchoa and R.F. Werneck, "Dual heuristics on the exact solution of large Steiner problems", *Electronic Notes in Discrete Mathematics* 7 , 2001 (online document at `http://www.elsevier.nl/locate/endm`).

[14] E. Uchoa, "Algoritmos para problemas de Steiner com aplicação em projeto de circuitos VLSI", PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro , 2001.

[15] E. Uchoa, M. Poggi de Aragão and C.C. Ribeiro, "Preprocessing Steiner problems from VLSI layout", *Networks* 40, 38–50, 2002.