Journal of Advances in Science and Technology

Vol. VIII, Issue No. XV, November-2014, ISSN 2230-9659

AN ANALYSIS ON VARIOUS TECHNIQUES, PROPERTIES AND ALGORITHMS OF LINEAR PROGRAMMING IN COMBINATORIAL OPTIMIZATION

Journal of Advances in Science and Technology
Vol. VIII, Issue No. XV, November-2014, ISSN 2230-9659

An Analysis on Various Techniques, Properties and Algorithms of Linear Programming in Combinatorial Optimization

Pal Dilip Kumar

Research Scholar, Sai Nath University, Ranchi, Jharkhand

Abstract – We study techniques, approximation algorithms, structural properties and lower bounds related to applications of linear programs in combinatorial optimization. The following Steiner tree problem is central: given a graph with a distinguished subset of required vertices, and costs for each edge, find a minimum-cost subgraph that connects the required vertices. We also investigate the areas of network design, multicommodity flows, and packing/covering integer programs. All of these problems are NP-complete so it is natural to seek approximation algorithms with the best provable approximation ratio. Overall, we show some new techniques that enhance the already-substantial corpus of LP-based approximation methods, and we also look for limitations of these techniques.

---------------------------♦-----------------------------

INTRODUCTION

What is the power of computers? How can mathematics help us understand the power of computers, and how can computers help us understand mathematics? These are the sort of theoretical questions that motivate this thesis, at a high level; we return to the interplay between computers and mathematics later in the introduction.

The concrete problem which motivates most of this thesis is the *Steiner tree problem,* which is as follows. You are given some required vertices (points) and some optional vertices, and want to build a graph (network) to connect the required vertices. You can purchase an edge (direct link) between any two points *u* and *v:* the cost of this edge, which you are given as part of the problem statement, is some dollar value c_{uv} that depends on *u* and *v.* The Steiner tree problem is then, *what is the cheapest way to purchase edges so that between any two required vertices, there is a path of edges?* Optional points can be included or excluded from the graph as you prefer.

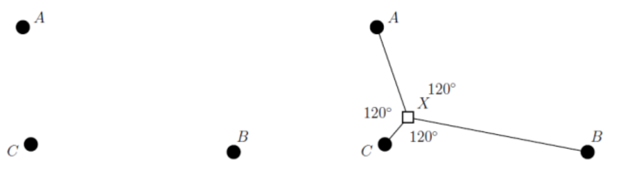

Figure 1: Left: an instance of the Steiner tree problem where there are three required vertices in the plane. Right: the solution for this instance uses one optional point.

In Figure 1 we give an example of how an optional point helps. There are three required points A *C* in the two-dimensional plane, every other point in the plane is an optional point, and the cost of connecting two points is the same as their distance; so the Steiner tree problem asks the shortest total length of line segments to connect .4, *B, C.* This very special case was investigated by the classical mathematicians Fermat and Torricelli in the mid-1600s; they found that the optimal network consists of three edges *AX, BX,CX* where the point *X* satisfies $\angle AXB = \angle BXC = \angle CXA = 120°$ (unless this *X* lies outside the triangle *ABC*, in which case the optimal solution is just the two shorter sides of the triangle.) We refer the reader to for these historical references.

We now skip forward a few centuries to the 1900s. Electronic computers were developed over the course of this century and so was a formal mathematical model of computation. Computers can be programmed to perform a variety of different tasks and can do basic mathematical operations much faster than a human. The Steiner tree problem has applications in industry (for example, the network could be for telecommunication, transportation, or chip layout) and so one wonders: just how quickly can a computer can solve the Steiner tree problem? The mainstream notion in theoretical computer science is the following abstract mathematical expression of speed (i.e. one that is independent of whether you use a Mac or a PC): we seek an algorithm (abstract computer program) with smallest *time complexity* (number of basic operations performed) as a function of the input size n.

Another mainstream notion is that a *fast* running time is any running time of the form at most n^D for constant D, so-called *polynomial time complexity.* In the 1970s, the complexity-theoretic notion of N P-completeness was developed by Cook and Levin; then Karp showed that Steiner tree is "NP-hard," so a fast algorithm for the Steiner tree problem is would imply fast algorithms for all "NP-complete"' problems. Moreover, there are lots of well-known NP-complete problems and the best known algorithms for them have running time like C^n for some constant $C > 1$. Since $n^D < C^n$ for large enough n, we cannot use any known algorithms solve the Steiner tree problem this quickly — there is a $1,000,000 conjecture that in fact no such algorithm exists.

It is therefore sensible to look at *approximation algorithms*: find a fast algorithm which outputs some valid answer that is *nearly* optimal, i.e. the output has cost within a factor *a* of the best possible, for some *a.* Such an algorithm is called an *a-approximation algorithm.*

LINEAR PROGRAMS

Algorithmic techniques based on *linear programs* (LPs) have driven many modern developments in combinatorial optimization problems related to the Steiner tree problem. The linear programming approach is to relax a discrete problem (in Steiner tree there are two discrete choices per connection, purchase it or don't purchase it) into a continuous variant (we now allow each edge e to be "purchased" to any fractional extent $0 \leq x_e \leq 1$). Then one must overcome two inter-related technical challenges: first, modeling the problem by linear constraints; second, recovering an integral solution from the fractional relaxation without increasing the cost too much.

It has been 10 years since the last improvement to the best-known ratio for the Steiner tree problem. Moreover, the Steiner tree problem is an example where LP methods have *not* driven innovation: none of the long list of approximation algorithms were developed by LP methods. The best LP-based result known is that an alternative 2-approximation algorithm can be obtained using LP technology. Nonetheless, the overall breadth and depth of LP methods has developed over time, and there is no negative result suggesting that LP methods will remain forever ineffective in this setting. In addition, LP methods are useful in practice for solving large- scale instances of the Steiner tree problem, by using integer programming software. Therefore, a large part of this thesis is devoted to developing and understanding modern LP technology for use in the Steiner tree problem. We also successfully develop LP-based approximation algorithms, with better approximation ratios than were previously known, for several other problems in combinatorial optimization.

LP PRELIMINARIES

While a substantial number of different successful techniques are known in the literature on LP methods, there is no hard-and-fast rule telling whether a given LP is useful or not. Some guidelines are known, including *small integrality gap* and *uncrossability.*

The *integrality gap V* of a linear program is a quantitative measure related to the suitability of an LP for use in designing approximation algorithms. We discuss it for minimization problems here, but analogous definitions hold for maximization problems. The integrality gap is defined as the worst-case cost ratio between the integral optimum cost(x_I^*)(i.e., min Steiner tree cost) and the fractional optimum $\mathcal{L}^*$(i.e., LP optimal value). The proof method of most LP-based approximation algorithms (e.g. *rounding* and *primal-dual)* is to find a feasible *x* and prove that $\texttt{cost}(x) \leq \alpha\mathcal{L}^*$, which guarantees an a-approximation since $\texttt{cost}(x) \leq \alpha\mathcal{L}^* \leq \alpha\texttt{cost}(x_I^*)$.

But if the integrality gap satisfies $\Gamma > \alpha$, we also have

$$\texttt{cost}(x) \geq \texttt{cost}(x_I^*) \geq \Gamma\mathcal{L}^* > \alpha\mathcal{L}^*,$$

and so $\texttt{cost}(x) \leq \alpha\mathcal{L}^*$is impossible. Conversely, approximation algorithms of this type, which we will call *LP-relative* approximation algorithms[2], prove that $\Gamma \leq \alpha$. One reason to demarcate the property of being LP-relative is that it is specifically necessary in some settings. The notion of an LP-relative approximation algorithm is already ubiquitous in the literature; but we are not aware of a name for it as such.

Even if an LP has a large integrality gap, one may still find it useful in designing an approximation algorithm. For example, Carr et al. gave a $\frac{21}{10}$-approximation algorithm for *edge-dominating set* using the natural LP. They also showed an integrality gap of $\frac{21}{10}$ for that LP, which precluded any better LP-relative ratio for that LP. But by strengthening the natural LP with additional constraints, two groups obtained another LP with integrality gap of 2 and also obtained a 2-approximation algorithm for the problem.

STEINER TREE LPS: AN OVERVIEW

In the *Steiner tree* problem, we are given an undirected graph *G* = (V, £), non-negative costs c_e for all edges $e \in E$, and a set of *terminal* vertices $R \subseteq V$. The goal is to find a minimum-cost tree *T* spanning /?, and possibly some *Steiner vertices* from *V* \ *R.* The problem takes a central place in the theory of combinatorial optimization and has numerous practical applications. The problem is one of original 21 NP-hard problems, and Chlebik and Chlebikova show that no (96/95 —ϵ)-approximation algorithm can exist for any positive *e* unless P=NP 15]. The same authors also show that it is NP-hard to obtain an approximation ratio better than $\frac{128}{127}$ for *quasi-bipartite* instances of the Steiner tree problem; these are instances in which no two Steiner vertices are adjacent in the underlying graph *G.*

One of the first approximation algorithms for the Steiner tree problem is the well-known *minimum-spanning tree heuristic* which is widely attributed to Moore . Moore's algorithm has a performance ratio of 2 for the Steiner tree problem and this remained the best known until the 1990s, when Zelikovsky suggested computing Steiner trees with a special structure, so called *r-restricted Steiner trees* based on the *full component decomposition.* Nearly all of the Steiner tree algorithms developed since then use r-restricted Steiner trees.

Given a Steiner tree T, a *full component* of *T* is a maximal subtree of *T* all of whose leaves are terminals and all of whose internal nodes are Steiner nodes. The edge set of any Steiner tree can be partitioned in a *unique* way into full components by splitting at internal terminals.

The above observation leads to the following hypergraph view of the Steiner tree problem which was first made explicit by Warme and Promel and Steger. Let $\mathcal{K}$ be

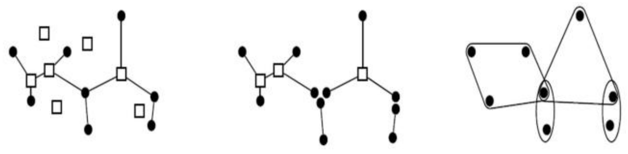

Figure 2: Black nodes are terminals and white nodes are Steiner nodes. Left: a Steiner tree for this instance. Middle: the Steiner tree's edges are partitioned into full components; there are four full components. Right: the hyperedges corresponding to these full components.

the set of all nonempty subsets of terminals (*hyperedges*). We associate with each $K \in \mathcal{K}$ a fixed min-cost full component spanning the terminals in K, and let C_K be its cost. The problem of finding a minimum-cost Steiner tree spanning R now reduces to that of finding a minimum-cost hyper-spanning tree in the hypergraph $(R, \mathcal{K})$.

The basic terminology for hypergraphs is as follows. A hyperedge (called just an edge sometimes) is any nonempty subset of the vertex set (usually of size at least 2). Vertices v, v' are connected if there is a sequence $v = v_1 \in e_1 \ni v_2 \in e_2 \ni \cdots \ni v_l = v'$ of vertices and hyperedges in the hypergraph. A connected hypergraph is one in which all vertices are connected. A simple cycle is a sequence of distinct vertices and hyperedges with $r_1 \in K_1 \ni r_2 \in K_2 \cdots \ni r_\ell \in K_\ell \ni r_1$ and $\ell > 1$. A hyper-spanning tree may then be defined as a connected hypergraph with no simple cycles: see the right of Figure 2 for an example.

LP Approaches & Background on MST - Development of combinatorial algorithms has been paralleled by extensive LP-based research to understand the Steiner tree problem. Such a polyhedral study often leads to better exact and approximate algorithms (although this has not yet actually happened in the particular case of Steiner trees). There is vast literature on various LP relaxations for the Steiner tree problem, e.g. One offshoot of better LP relaxations is to achieve vast improvements in the area of integer programming-based exact algorithms for the Steiner tree problem.

Despite their apparent strength, none of the above LP relaxations has been proven to exhibit an integrality gap smaller than 2 in general. In particular, the best LP-based approximation algorithm for the Steiner treeproblem is due to Goemans and Bertsimas, and has a performance guarantee of 2 —$\frac{2}{|R|}$. This algorithm uses the weakest among the Steiner tree LP formulations cited above — the *undirected cut relaxation* — whose integrality gap is precisely 2 — $\frac{2}{|R|}$.

Improved LP-based approximation algorithms have so far only been obtained for structured instances of the Steiner tree problem. Notably, Chakrabarty, Devanur, and Vazi- rani recently showed that the *bidirected cut relaxation* has an integrality gap of at most 4/3 for quasi-bipartite instances of the problem, improving upon an earlier bound of 3/2. (Nonetheless, RZ achieves an approximation ratio better than $\frac{4}{3}$ for these graphs.) The worst known example is due to Goemans and exhibits an integrality gap of only. The bidirected cut relaxation is widely conjectured to have an integrality gap smaller than 2, but a proof remains elusive.

LP INTERPRETATIONS

In this paper we provide algorithmic evidence that the primal-dual method is useful for the Steiner tree problem. We first present a novel LP relaxation for the Steiner tree problem that generalizes the LP$(\mathcal{P})$presented earlier. We then show that the algorithm RZ of Robins and Zelikovsky can be analyzed as an iterated primal-dual algorithm using this relaxation.

Robins and Zelikovsky showed that, for every fixed r, the performance ratio of RZ is $1.279\rho_r$ in quasi-bipartite graphs, and $1.55\rho_r$ in general graphs. We prove an interpolation of these results. For a Steiner vertex v, define its *Steiner neighbourhood* to be the collection of vertices that are in the same connected component as v in *G\R*. A graph is *b-quasi-bipartite* if all of its Steiner neighbourhoods have cardinality at most *b*. We prove the following, where opt_rdenotes the optimal cost after *r*-preprocessing:

Theorem 1. *Given an undirected, b-quasi-bipartite graph G = (V_yE), terminals*$R \subseteq V$, *and a fixed constant*$r \geq 2$, *Algorithm* RZ *returns a feasible Steiner tree T s.t.*

$$c(T) \leq \begin{cases} 1.279 \cdot \mathrm{opt}_r & : \quad b = 1 \\ (1 + \frac{1}{e}) \cdot \mathrm{opt}_r & : \quad b \in \{2,3,4\} \\ (1 + \frac{1}{2} \ln (3 - \frac{2}{b}))\, \mathrm{opt}_r & : \quad b \geq 5. \end{cases}$$

Note that fe-quasi-bipartite graphs are a natural interpolation between quasi-bipartite graphs (6=1) and general graphs$(b \leq |V \backslash R|)$, hence Theorem 1 interpolates the two main results of Robins and Zelikovsky.

Unfortunately, Theorem 1 does not imply that our new relaxation has a small integrality gap. Nonetheless, we obtain the following bounds, when *G* is 6-quasi-bipartite:

Theorem 2. *Our new relaxation has an integrality gap between* $\frac{8}{7}$ *and* $\frac{2b+1}{b+1}$ We remark that *filtering* approach of Chakrabarty et al. , can be applied to improve the gap upper bound to $\frac{2b-1}{b}$ for $b \geq 2$.

A New LP Relaxation for Steiner Trees - In this paper, we work on r-preprocessed graphs as introduced— so for example $\mathcal{K}$represents element-disjoint full components, one for every possible set of up to r terminals.

We also use a type of metric assumption which holds without loss of generality. The standard one is that the costs satisfy the triangle inequality and *G* is a complete graph. Here we instead work under a weaker metric assumption that is also without loss of generality for the purposes of computing optimal Steiner trees. The assumption is that, for every triple u, v, w of nodes with v a Steiner node such that $uv, vw \in E$, we have $uw \in E$ and $c_{uw} \leq c_{uv} + c_{vw}$. This preserves the Steiner neighbourhoods (unlike the standard metric assumption) and has the important property that for every K, we may assume that the min-cost full component (tree) with leaf set *K* has degree at least 3 in all internal (Steiner) nodes. Consequently, for example, every r-restricted instance is also (r —2)-quasi-bipartite.

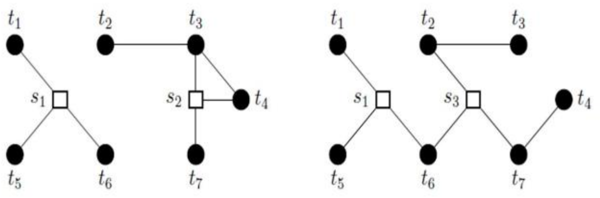

Figure 3: Left: a collection *S* = *{{t_1,t_5,t_6}, { t_3,t_4,t_7 }, { t_2,t_3*}, { *t_3,t_4*}} of 4 full components. Right: a Steiner tree with $\mathcal{S}$-decomposition *({t_1s_1, t_5s_1,t_6s_1,t_2t_3}, {{t_2,t_6,t_7}, {t_4,t_7}}).*

Whenever$\mathcal{S} \subset \mathcal{K}$, we will abuse notation and identify the symbol $\mathcal{S}$with the graph obtained by deleting everything except the full components in$\mathcal{S}$. For example, $E(\mathcal{S})$ denotes the set of all edges of the full components in$\mathcal{K}$. We will always have$\binom{R}{2} \subset \mathcal{S}$, therefore the graph $\mathcal{S}$always contains the induced subgraph $G[R]$

INTEGRAL MULTICOMMODITY FLOW IN TREES

In the max-weight integral multicommodity flow problem (WMF), we are given an undirected *supply* graph *G* = (V, *E)*, terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$ where $s_i, t_i \in V$, non-negative weights $w_1, \ldots, w_k$ and non-negative integral capacities. We distinguish between three versions of the problem: in edge-WMF each edge $e \in E$has a capacityc_e; in arc-WMF each of the *2\E\* directed arcs (u, v) with $\{u, v\} \in E$has a capacity c_{uv}; in vertex-WMF each vertex $v \in V$has a capacity c_v. The goal is to simultaneously route integral s_i-t_i flows of value y_i, subject to the capacities, so as to maximize the weight$\sum w_i y_i$. Note that edge-WMF can be reduced to vertex-WMF by subdividing each edge, moving that edge's capacity onto the new vertex, and setting all other vertex capacities to be infinite. When we make statements that apply to all three versions, we simply say WMF.

The single-commodity version (k = 1) of WMF is well-known to be solvable in polynomial time. If we drop the integrality restriction the problem can be solved in polynomial time via linear programming for any k. However, when integrality is required, even the 2-commodity unit-capacity, unit-weight arc- and edge-versions are NP-complete — see Even, Itai, and Shamir . Let $n := |V|$. Recent results on the *edge-disjoint paths* prob lem show more strongly that arc-WMF isNP-hard to $|E|^{\frac{1}{2}-\epsilon}$-approximate, and edge-WMF cannot be approximated better than $\log^{\frac{1}{2}-\epsilon}(n)$unless NP ⊂ ZPTIME($n^{\text{polylog}(n)}$)i. Random ized rounding gives a $(1+\epsilon)$-approximation algorithm for WMF when all capacities are at least $\log n/\epsilon^2$, for suitably small ϵ.

An easier and significant special case of WMF is where the supply graph *G* is a tree, which we denote by WMFT. Garg, Vazirani and Yannakakis considered the unit- weight case of edge-WMFT and showed APX-hardness even if *G's* height is at most 3 and all capacities are 1 or 2; but on the positive side, they gave a 2-approximate polynomial- time primal-dual algorithm. Techniques of Garg et al. show that WMFT can be solved in polynomial time when *G* has unit capacity (using dynamic programming and matching) or is a star (this problem is essentially equivalent to 6-matching). The case where *G* is a path (so-called *interval packing)* is also polynomial-time solvable, e.g. by linear programming since the natural LP has a totally unimodular constraint matrix. Arc- WMF on unit-capacity bidirected trees admits a $(\frac{5}{3}+\epsilon)$-approximation algorithm.

In general, the best result for edge- and arc-WMFT is a 4-approximation of Chekuri, Mydlarz and Shepherd . Vertex-WMFT has not been explicitly studied as far as we are aware, although we observe later (Proposition 6.11) that techniques of yield a 5-approximation.

CONCLUSION

The first and foremost question would be to improve the best known approximation ratio for the Steiner tree problem, as well as integrality gaps for the hypergraphic and bidirected LPs. After this thesis was submitted to its examiners, a preprint of Byrka et al. was We noted that the hypergraphic relaxations can be solved in polynomial time on *r*- restricted instances, and approximately solved in general, but it is open whether we can solve them exactly in general. One of our results is that the bidirected and hypergraphic LPs have the same value on quasi-bipartite instances.

REFERENCES

- A.Agarwal and M. Charikar. On the advantage of network coding for improving network throughput. In *Proc. IEEE Information Theory Workshop*, pages 105–109, 2004. 12, 14, 63, 74
- A.Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem in networks. *SIAM J. Comput.*, 24:440–456, 1995. Preliminary version appeared in *Proc. 23rd STOC*, pages 134-144, 2001. 3, 12, 79
- approximation algorithm for a generalization of the weighted edge-dominating set problem. *J. Comb. Optim.*, 5(3):317–326, 2001. Preliminary version appeared in *Proc. 8th ESA*, pages 132–142, 2000. 4
- Ernst Althaus, Tobias Polzin, and Siavash Vahdati Daneshmand. Improving linear programming approaches for the Steiner tree problem. In *Proc. 2nd WEA*, pages 620–621, 2003. 3
- Gruia Calinescu, Amit Chakrabarti, Howard J. Karloff, and Yuval Rabani. Improved approximation algorithms for resource allocation. In *Proc. 9th IPCO*, pages 401–414, 2002. 104
- H.J. Pr¨omel and A. Steger. *The Steiner Tree Problem — A Tour through Graphs, Algorithms, and Complexity*. Vieweg Verlag, Braunschweig-Wiesbaden, 2002. 2, 12
- M.X. Goemans and D. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Math. Programming*, 60:145–166, 1993. Preliminary version appeared in *Proc. 1st SODA*, pages 388-396, 2000. 3, 12, 44, 79, 148, 149, 150
- R.M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, NY, 1972. 2, 9
- Robert D. Carr, Toshihiro Fujito, Goran Konjevod, and Ojas Parekh.
- S.Hougardy and H. J. Pr¨omel. A 1.598 approximation algorithm for the Steiner problem in graphs. In *Proc. 10th SODA*, pages 448–453, 2009. 2, 3, 5, 12

- Xiuzhen Cheng and Ding-Zhu Du, editors. *Steiner Trees in Industry*, volume 11 of *Combinatorial Optimization*. Springer, 2002. 2

- Y.P. Aneja. An integer linear programming approach to the Steiner problem in graphs. *Networks*, 10:167–178, 2002. 5, 12