A Branch-and-cut-and-price algorithm for the Stackelberg Minimum Spanning Tree Game

Vinicius Morais, Alexandre Salles da Cunha, Philippe Mahey

Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 52 (2016) 309–316 www.elsevier.com/locate/endm

A Branch-and-cut-and-price algorithm for the Stackelberg Minimum Spanning Tree Game

Vinicius Morais [1]

Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
vwcmorais@dcc.ufmg.br

Alexandre Salles da Cunha [2]

LIMOS, UMR 6158-CNRS Université Blaise-Pascal, BP 10125, F-63173 Aubiére, Cedex, France and Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
acunha@dcc.ufmg.br

Philippe Mahey

LIMOS, UMR 6158-CNRS Université Blaise-Pascal, BP 10125, F-63173 Aubiére, Cedex, France
philippe.mahey@isima.fr

Abstract

The Stackelberg Minimum Spanning Tree Game (StackMST) is defined in terms of a graph $G = (V, B \cup R)$, with two disjoint sets of edges, blue B and red R, and costs $\{c_e \geq 0 : e \in R\}$ defined for the red edges. Once the leader of the game defines prices $\{p_e : e \in B\}$ to the blue edges, the follower chooses a minimum weight spanning tree (V, E_T), at cost $\sum_{e \in B \cap E_T} p_e + \sum_{e \in R \cap E_T} c_e$. The goal is to find prices to maximize the revenue $\sum_{e \in B \cap E_T} p_e$ collected by the leader. We introduce a reformulation and a Branch-and-cut-and-price algorithm for StackMST. The reformulation is obtained

http://dx.doi.org/10.1016/j.endm.2016.03.041

after applying KKT optimality conditions to a StackMST non-compact Bilevel Linear Programming formulation and is strengthened with a partial rank-1 RLT and with valid inequalities from the literature. We also implemented a Branch-and-cut algorithm for an extended formulation derived from another in the literature. A preliminary computational study comparing both methods is also presented.

Keywords: Stackelberg Games, Spanning Trees, Branch-and-cut-and-price.

1 Introduction

The Stackelberg Minimum Spanning Tree Game (StackMST) is defined in terms of a undirected graph $G = (V, E)$, with two sets of edges, blue B and red R ($E = B \cup R$, $B \cap R = \emptyset$) and fixed costs $\{c_e \geq 0 : e \in R\}$ assigned to the red edges. Once the leader of the game defines prices $\{p_e : e \in B\}$ to the blue edges, the follower chooses a minimum weight spanning tree (MST) (V, E_T) of G, at cost $\sum_{e \in B \cap E_T} p_e + \sum_{e \in R \cap E_T} c_e$. StackMST aims at finding prices that maximize the *revenue* $\sum_{e \in B \cap E_T} p_e$ collected by the leader. It is assumed that (V, R) is connected (otherwise StackMST would be unbounded) and that given two edges $b \in B$ and $r \in R : p_b = c_r$, the follower prefers picking the blue one (thus, prices being fixed, the revenue in all MSTs are the same [1]). StackMST is a spanning tree analog to the Stackelberg Shortest Path Game [4] introduced in [1] (see also [2]).

Cardinal et al. [1] proved that StackMST is NP-hard and APX-hard, proposed an approximation algorithm and formulated StackMST as an integer program whose Linear Programming (LP) relaxations have integrality gaps bounded by the same approximation factor.

We present a reformulation and a Branch-and-cut-and-price (BCP) algorithm for StackMST. Our reformulation strategy follows three steps: we replace the inner optimization problem of a non-compact Bilevel LP Problem (BLPP) model for StackMST by Karush-Kuhn-Tucker (KKT) optimality conditions, obtaining a non-linear single level formulation [4]. We then replace non-linear terms by conveniently defined binary quadratic ones and linearize the model. Finally, the formulation is strengthened with a partial application of the Reformulation by Linearization Technique (RLT) and with StackMST valid inequalities in [1]. We also implement a Branch-and-cut (BC) algorithm

[1] Vinicius Morais is funded by CAPES BEX 7461/14-3.
[2] Alexandre Salles da Cunha is partially funded by CNPq grants 200493/2014-0, 305423/2012-6, 471464/2013-9 and FAPEMIG CEX-PPM-00164-13.

for an extended formulation that derives from that in [1]. We present computational experiments that suggest that BCP is a promising solution approach.

2 A StackMST Reformulation

For any function $f : Q \to \mathbb{R}$ defined over a finite domain Q and $Q' \subseteq Q$, $f(Q') = \sum_{q \in Q'} f_q$. Given $S \subset V, S \neq \emptyset$, $E(S) = \{\{i,j\} \in E : i,j \in S\}$ and $\delta(S) = \{\{i,j\} \in E : i \in S, j \notin S\}$. Denote by $\{c^k : k = 1,\ldots,K\} = \{c_e : e \in R\}$ $(c^1 < c^2 < \ldots < c^K)$ the set of $K \geq 2$ distinct red costs. Assume that $\mathcal{T}$ denotes the set of all spanning trees of G and that Z_{ST} denotes the set of vectors in $\mathbb{B}^{|E|}$ $(\mathbb{B} = \{0,1\})$ that are support vectors of subgraphs in $\mathcal{T}$.

Let $\{p_e \in \mathbb{R} : e \in B\}$ and $\{z_e \in \mathbb{B} : e \in E\}$ respectively denote decision variables used to indicate the prices set by the leader and the edges included in the MST (V, E_T) picked by the follower. If $e \in E_T$, $z_e = 1$; otherwise, $z_e = 0$. Denote by $\mathcal{P}_{ST} = \{z \in \mathbb{R}_+^{|E|} : z(E) = |V|-1, z(E(S)) \leq |S|-1, S \subset V, S \neq \emptyset\}$ the convex hull of Z_{ST}. Note that $\mathcal{P}_{ST}$ is represented in terms of exponentially many subtour elimination constraints (SECs).

A BLPP formulation for StackMST is:

$$\max \left\{ \sum_{e \in B} p_e z_e : z \in \arg\min \left\{ \sum_{e \in B} p_e z_e + \sum_{e \in R} c_e z_e : z \in \mathcal{P}_{ST} \right\} \right\}. \tag{1}$$

In order to reformulate (1) as a single level program, let α and $\{\mu_S \geq 0 : S \subset V, S \neq \emptyset\}$ be dual variables respectively assigned to $z(E) = |V| - 1$ and to the SECs in the inner LP of (1). Alongside with constraints defining $\mathcal{P}_{ST}$, KKT optimality conditions include $\{\mu_S \geq 0 : S \subset V, S \neq \emptyset\}$ and:

$$\sum_{e \in B} p_e z_e + \sum_{e \in R} c_e z_e \leq \alpha(|V| - 1) - \sum_{S \subset V, S \neq \emptyset} \mu_s(|S| - 1), \tag{2}$$

$$\alpha - \sum_{S : e \in E(S)} \mu_s \leq p_e, \qquad e \in B \tag{3}$$

$$\alpha - \sum_{S : e \in E(S)} \mu_s \leq c_e, \qquad e \in R \tag{4}$$

For reasons that will become clear later on, LP strong duality condition (2) is stated in inequality form. Indeed, that can be done since LP weak

duality rules out the satisfaction of (2) slack. A result in [1] states that in every optimal price function, the prices assigned to the blue edges that actually appear in any follower's MST belong to $\{c^k : k = 1, \dots, K\}$. Define one binary variable x_e^k to each $e \in B$ and cost index k. If $p_e = c^k$, $x_e^k = 1$. Otherwise, $x_e^k = 0$. As long as $\{z_e \in \mathbb{B} : e \in B\}$ and (5) are enforced, $p_e = \sum_{k=1}^{K} c^k x_e^k$ and the revenue can be written as $\sum_{k=1}^{K} \sum_{e \in B} c^k z_e x_e^k$.

$$\sum_{k=1}^{K} x_e^k = 1, \; e \in B. \tag{5}$$

Let $\{l_e^k = z_e x_e^k, e \in B, k = 1, \dots, K\}$ denote linearization variables. StackMST can then be formulated as:

$$\max \left\{ \sum_{k=1}^{K} \sum_{e \in B} c^k l_e^k : (z_B, z_R, l, x, \alpha, \mu) \in \mathcal{P} \cap (\mathbb{B}^{|B|}, \mathbb{R}^{|R|}, \mathbb{B}^{K|B|}, \mathbb{B}^{K|B|}, \mathbb{R}, \mathbb{R}_+^o) \right\} \tag{6}$$

where $o = 2^{|V|} - |V| - 1$, $z = (z_B, z_R)$ is partitioned in its blue ($z_B \in \mathbb{B}^{|B|}$) and red ($z_B \in \mathbb{R}^{|R|}$) components and $\mathcal{P}$ is the intersection of $\mathcal{P}_{ST}$, (4)-(5) and:

$$\sum_{k=1}^{K} \sum_{e \in B} c^k l_e^k + \sum_{e \in R} c_e z_e \leq \alpha(|V| - 1) - \sum_{S \subset V, S \neq \emptyset} \mu_s(|S| - 1) \tag{7}$$

$$\alpha - \sum_{S : e \in E(S)} \mu_s \leq \sum_{k=1}^{K} c^k x_e^k, \qquad e \in B \tag{8}$$

$$z_e + x_e^k - l_e^k \leq 1, \qquad e \in B, \; k = 1, ..., K \tag{9}$$

$$\sum_{k=1}^{K} l_e^k = z_e \qquad e \in B \tag{10}$$

$$0 \leq l_e^k \leq x_e^k, \qquad e \in B, \; k = 1, ..., K \tag{11}$$

Since $l_e^k \geq 0$, constraints (10) imply $\{l_e^k \leq z_e : e \in B, k = 1, \dots, K\}$. They can be proven valid by applying a partial RLT: multiply (5) by $z_e, e \in B$ and replace the products $z_e x_e^k$ by l_e^k. The model can be further strengthened by:

$$\sum_{e \in P \cap B} (1 - z_e) \geq \sum_{t=k}^{K} l_f^t, \; f = \{a, b\} \in B, k = 2, \dots, K, \; P_{ab} \in G^{k-1}, \tag{12}$$

where, for a given $k = 2, \ldots, K$ and blue edge $f = \{a, b\}$, P_{ab} denotes any path that starts at a and ends at b in the graph $G^{k-1} = (V, B \setminus \{f\} \cup \{e \in R : c_e \leq c^{k-1}\})$. Inequalities (12) come from [1] and rely on a nice characterization of admissible prices introduced in that reference. They state that if a cycle of G has all its red edges with cost at most c^{k-1} and some blue edge f in the cycle belongs to the MST with $p_f \geq c^k$, then the cycle includes at least another blue edge that does not belong to the MST. In the remainder, denote by $\mathcal{P}^+$ the intersection of $\mathcal{P}$ with (12). Given any linear integer programming formulation $\mathcal{U}$ for StackMST, denote by $v(\mathcal{U})$ its LP relaxation upper bound.

3 A Branch-and-cut-and-price algorithm

To show how $v(\mathcal{P}^+)$ is evaluated, assume that a Restricted Linear Programming Master Problem (RLPM) is obtained after relaxing all SECS and (12), and considering only those variables $\{\mu_S : |S| = 2, S = \{i, j\} \in E\}$ among $\{\mu_S : S \subset V, S \neq \emptyset\}$. Assume that such a RLPM is solved to optimality and that $\overline{z} \in [0,1]^{|E|}, \overline{x}, \overline{l} \in [0,1]^{K|B|}, \overline{\alpha} \in \mathbb{R}$ and $\overline{\mu} \in \mathbb{R}_+^{|E|}$ denote its optimal solution. SECs are separated with the heuristic and exact algorithms in [3]. The most violated SEC is always added to the RLPM. The others are added if they are orthogonal enough to the most violated one. Sufficiently orthogonal inequalities are those whose inner product (evaluated after normalizing the inequalities with the Euclidean norm) does not exceed 0.04. Constraints (12) are also separated in polynomial time. For each $f = \{a, b\} \in B : \sum_{t=1}^{K} \overline{l}_f^t > 0$ and $k = 2, \ldots, K$, we compute a shortest path connecting a to b in G^{k-1}. For the shortest path computations, Dijkstra's algorithm is used; red edges have length equal to 0 and blue edges have length equal to $1 - \overline{z}_e$. Inequalities (12) are only separated when no violated SECs are found.

Let us now discuss how the column generation problem is solved. Assume that $\overline{z}$ satisfies all SECs and that dual variables associated to constraints (7), (4) and (8) in the LP dual of the LP relaxation implied by $\mathcal{P}^+$ are respectively defined by $\theta \geq 0$, $\{\varphi_e \geq 0 : e \in R\}$ and $\{\gamma_e \geq 0 : e \in B\}$. Additionally, let $\overline{\theta}$, $\{\overline{\varphi}_e : e \in R\}$ and $\{\overline{\gamma}_e : e \in B\}$ be the corresponding optimal solutions to the LP dual of RLPM. To check whether or not $(\overline{z}, \overline{x}, \overline{l}, \overline{\alpha}, \overline{\mu})$ also solves the LP relaxation of $\mathcal{P}^+$, we must check if any constraint $\sum_{e \in R \cap E(S)} \varphi_e + \sum_{e \in B \cap E(S)} \gamma_e \leq \theta(|S| - 1)$, $\forall S \subset V$, $|S| \geq 3$ of the dual of the LP relaxation of $\mathcal{P}^+$ is violated by $(\overline{\theta}, \overline{\varphi}, \overline{\gamma})$. If $\overline{\theta} = 0$, we must have $\overline{\gamma}_e = 0, e \in B$ and $\overline{\varphi}_e = 0, \forall e \in R$ and they are all satisfied. If $\overline{\theta} > 0$, they read as $\sum_{e \in R \cap E(S)} \frac{\overline{\varphi}_e}{\overline{\theta}} + \sum_{e \in B \cap E(S)} \frac{\overline{\gamma}_e}{\overline{\theta}} \leq (|S| - 1), \forall S \subset V, |S| \geq 3$. Since all variables $\{\mu_S : |S| =$

$2, S = \{i, j\} \in E\}$ were included in the very first RLMP, $\frac{\overline{\varphi}_e}{\overline{\theta}} \leq 1, \forall e \in R$ and $\frac{\overline{\gamma}_e}{\overline{\theta}} \leq 1, \forall e \in B$ apply. Thus, the column generation subproblem is solved by the SEC separation algorithms in [3], having $\{\frac{\overline{\varphi}_e}{\overline{\theta}} : e \in R\}$ and $\{\frac{\overline{\gamma}_e}{\overline{\theta}} : e \in B\}$ as the point to be separated. When no violated constraints nor columns with positive reduced costs are found, the objective function value of RLPM matches $v(\mathcal{P}^+)$.

If there are fractional $\overline{x}_e^k$ values for the LP relaxation solution at a given node, BCP branches on generalized upper bound (GUB) constraints (5). Among edges $\{e \in B : \overline{x}_e^k \in (0, 1), \text{ for any } k\}$, it applies the strong branching approach to define the most promising one. Assuming that $e \in B$ denotes that edge, two child nodes are created, enforcing $\sum_{k=1}^{\overline{K}} x_e^k = 1$ and $\sum_{k=\overline{K}+1}^{K} x_e^k = 1$ respectively in each of them ($\overline{K} = \min\{t : \sum_{k=1}^{t} \overline{x}_e^k \geq 0.5\}$). Otherwise, if $\overline{x}_e^k \in \mathbb{B}, \forall e \in B, k = 1, \ldots, K$, BCP then branches on fractional $z_e : e \in B$, once again, using strong branching. BCP implements a best-first search and only uses CPLEX as its LP solver.

BCP is initialized with the following valid lower bounds. After computing a minimum weight spanning tree (V, R_T) for the subgraph (V, R), each vertex $v \in V$ is labelled $l_v = \min\{c_e : e \in \delta(\{v\}) \cap R_T\}$. Then, a MST for $(V, R \cup B)$ under prices $\{p_e = \min\{l_i, l_j\} : e = \{i, j\} \in B\}$ is computed. The revenue collected in that MST is an initial valid StackMST lower bound. When the LP relaxation $(\overline{z}, \overline{x}, \overline{l}, \overline{\alpha}, \overline{\mu})$ of a given BCP node is computed, we define $K^e = \{k = 1, \ldots, K : \overline{l}_e^k > 0\}$, $\overline{B} = \{e \in B : K^e \neq \emptyset\}$, set $\{p_e = \max\{c^k : k \in K^e\} : e \in \overline{B}\}$ and compute the MST for graph $(V, R \cup \overline{B})$. Then, every edge $e \in \overline{B}$ has its prize set to c^k for every $k \in K^b$, one edge with one new price at a time, and the MST is re-computed. For each of these spanning trees, the revenue is evaluated, attempting to update the best lower bound.

Since no computational evaluation for StackMST algorithms is available in the literature, we compare BCP to a BC algorithm based on an extended formulation that derives from that in [1]. The formulation uses variables $\{z_e \in \{0, 1\} : e \in B\}$ and the set of decision variables $\{u_e^k \in \mathbb{B} : e \in B, k = 1, \ldots, K\}$ used in [1]. According to [1], $u_e^k = 1$ if edge e is included in the MST with price $p_e \geq c^k$. If $p_e \leq c^{k-1}$ or if e is not included in the MST, $u_e^k = 0$. The formulation behind BC is $\max\{\sum_{k=1}^{K} \sum_{e \in B} (c^k - c^{k-1}) u_e^k : (z, u) \in \mathcal{C} \cap (\mathbb{R}_+^{|E|} \times \mathbb{B}^{K|E|})\}$, where polytope $\mathcal{C}$ is given by the intersection of $\mathcal{P}_{ST}$, $\{\sum_{e \in P \cap B} (1 - u_e^1) \geq u_f^k, f = \{a, b\} \in B, k = 2, \ldots, K, P_{ab} \in G^{k-1}\}$, $\{z_e = u_e^1, e \in B\}$ and $\{u_e^{k-1} \geq u_e^k, e \in B, k = 2, \ldots, K\}$ (note $\mathcal{C}$ requires cost $c^0 = 0$ to be defined). Variables z are used so BC separates SECs instead of the polytope-wise equivalent partition inequalities in [1]. BC uses the same separation

algorithms for SECs and (12) described before, embedded as callback routines for CPLEX MIP solver, that manages the search tree. BC also uses the heuristic used to initialize BCP, described above. We conjecture that $\mathcal{C}$ and the formulation in [1] provide the same LP bounds, but a proof remains to be given.

4 Computational experiments and future work

BCP and BC were implemented in `C++` and tested with a 2.4GHz Intel XEON E5645 machine, with 32 GB of RAM, under Linux Operating System. The StackMST instances are generated as follows. Given a desired integer $K \in \{3, 5, 7\}$, K integer red costs $\{c^1, \ldots, c^K\}$ are randomly chosen in $[1, 150]$, with uniform probability. Then, given a desired number $n \in \{20, 30, 50, 70\}$ of vertices, we randomly generate a tree $(V, \hat{E})$ spanning them. Let E_c denote the set of edges of a complete graph with vertex set V. We then initialize: $E \leftarrow \hat{E}$, $R \leftarrow \hat{E}$, $B \leftarrow \emptyset$. Additional edges are then randomly picked from $E_c \setminus \hat{E}$, until a desired graph density $d \in \{10\%, 20\%, 30\%, 50\%\}$ is obtained. If a given edge in $e \in E_c \setminus \hat{E}$ is added to E, we then choose, with equal probability, if $R \leftarrow R \cup \{e\}$ or $B \leftarrow B \cup \{e\}$. In the former case, the cost assigned to e is randomly chosen from $\{c^1, \ldots, c^K\}$, with uniform probability. Each instance name clearly indicates the corresponding values for n, d and K. To illustrate, for instance $n30d50k3$, $n = |V| = 30, d = 50\%$ and $K = 3$.

Computational results are presented in Table 1, whose first column identifies the instance being considered. The next three columns provide LP upper bounds: $v(\mathcal{C}), v(\mathcal{P})$ and $v(\mathcal{P}^+)$. We then provide computational results for BC and BCP. For each algorithm, we provide: the best upper (BUB) and lower (BLB) bounds found during the search, the corresponding duality gap, and the total CPU time, $t(s)$, in seconds. Whenever an instance was not solved to proven optimality within a time limit of 5 hours, an indication "tl" is provided.

As one could expect, StackMST instances become very difficult with the increase of K and the graph density. BC solved 4 out of 16 instances to proven optimality, one of them also being solved by BCP. For the remaining three cases, BCP manages to find the optimal solution, but fails in providing an optimality certificate. That happens despite the fact that BCP dual bounds at the root node are much stronger than BC counterparts. Note that, in many cases, the best upper bound BCP outputs at the end of the time limit is essentially the root upper bound $v(\mathcal{P}^+)$. Nevertheless, these figures are always much stronger than the best upper bounds provided by BC, under the same time limit, for instances they do not solve. As a result, BCP provides smaller

duality gaps for almost all instances left unsolved by both methods. These results suggest that BCP is a promising solution approach. We should investigate other branching strategies, so that BCP benefits more from the stronger upper bounds it relies on. We also intend to proceed with the characterization of StackMST valid inequalities to separate them into BCP and BC.

				BC				BCP based on $\mathcal{P}^+$			
	$v(\mathcal{C})$	$v(\mathcal{P})$	$v(\mathcal{P}^+)$	BUB	BLB	gap(%)	$t(s)$	BUB	BLB	gap(%)	$t(s)$
n20d30k7	787	597	597	541	541	-	44	597	541	9.5	t.l.
n20d50k3	190	190	190	190	190	-	0	190	190	-	0
n20d50k5	599	467	467	575	407	29.2	t.l.	467	395	15.4	t.l.
n30d30k3	493	427	425	413	413	-	25	425	413	3.0	t.l.
n30d50k3	2134	1862	1862	2134	1798	15.7	t.l.	1862	1830	1.7	t.l.
n30d50k5	1870	1320	1320	1854	1320	28.8	t.l.	1320	1254	5.0	t.l.
n30d50k7	813	524	524	752	498	33.8	t.l.	524	497	5.2	t.l.
n50d10k5	1701	1592	1588	1470	1470	-	0	1588	1470	7.5	t.l.
n50d10k7	861	853	828	838	713	15.0	t.l.	828	732	11.6	t.l.
n50d20k3	2778	2301	2301	2778	2115	23.9	t.l.	2301	2239	2.7	t.l.
n50d20k7	1156	799	760	1127	570	49.4	t.l.	760	582	23.5	t.l.
n70d10k3	4807	4714	4694	4718	4641	1.6	t.l.	4694	4599	2.0	t.l.
n70d10k7	2342	2023	2002	2301	1825	20.7	t.l.	2002	1604	19.9	t.l.
n70d20k3	780	763	763	780	759	2.7	t.l.	763	759	0.5	t.l.
n70d20k5	1873	1173	1173	1873	835	55.4	t.l.	1173	934	20.4	t.l.
n70d30k5	1995	1227	1227	1995	1167	41.5	t.l.	1227	1167	4.9	t.l.

Table 1: Computational results. Upper bounds are rounded down.

References

[1] Cardinal, J., Demaine, E.D., Fiorini, S., Gwenael, J., Langerman, S., Newman, I. and Weimann, O. The stackelberg minimum spanning tree game. *Algorithmica*, 59(2):129–144, 2011.

[2] Cardinal, J., Demaine, E.D., Fiorini, S., Gwenael, J., Newman, I. and Weimann, O. The stackelberg minimum spanning tree game on planar and bounded-treewidth graphs. *Journal of Combinatorial Optimization*, 25(1):19–46, 2013.

[3] Gendron, B., Lucena, A., da Cunha, A.S. and Simonetti, L. Benders decomposition, branch-and-cut, and hybrid algorithms for the minimum connected dominating set problem. *INFORMS Journal on Computing*, 26(4):645–657, 2014.

[4] Labbé, M. and Violin, A. Bilevel programming and price setting problems. *4OR-Q J Oper Res*, 11(1):1–30, 2013.