Decomposition Methods for Stochastic Steiner Trees

M. Leitner[2] I. Ljubić[1] M. Luipersbeck[2] M. Sinnl[2]

[1] ESSEC Business School of Paris, France
[2] ISOR, University of Vienna, Austria

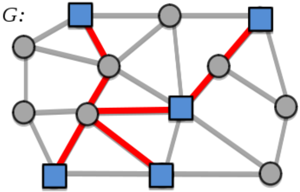
2nd European Conference on Stochastic Optimization (ECSO 2017)
September 20-22, 2017
Roma Tre University, Rome, Italy

Deterministic Steiner Tree Problem (STP)

Deterministic STP

- **Given:** undirected graph $G = (V, E)$, positive edge costs c_e, set of terminals $T \subset V$, $T \neq \emptyset$.
- **Objective:**

$$\min\{c(E_0) : E_0 \subset E, E_0 \text{ spans } R\}.$$



Decision problem NP-complete. Well studied, many applications, recent DIMACS Challenge (non-trivial graphs with 100 000's of nodes solved to optimality).

WHY DO WE STUDY STEINER TREES UNDER UNCERTAINTY?

Steiner Tree Problem (STP) Under Uncertainty

In practice, two sources of uncertainty:

- **Who are the terminals?** No precise knowledge of future customer demands.
- **What are the edge installation costs?** Future edge costs may be more expensive and prices are highly volatile ("wait and see" can be costly).

One possible approach: **Stochastic Optimization**

Estimate possible outcomes and derive scenarios:

- Each scenario k assumes terminals $T^k \subset V$ are given and edge costs c^k are specified.

Decision Process: **Two Stages**

- **First Stage**: ("now", Monday): buy cheap/profitable edges now. **Difficulty:** we only know possible outcomes and their probabilities.
- **Second Stage**: ("future", Tuesday, one scenario is realized): additional edges are purchased to make the solution feasible (**recourse action**).

SSTP: Formal Problem Definition

SSTP

- **Given:** Undirected graph $G = (V, E)$, root $r \in V$, positive edge costs c_e^0, $e \in E$. Set of scenarios K, s.t. $k \in K$:
  - probability $p^k > 0$,
  - edge costs c_e^k, $e \in E$,
  - set of terminals $T^k \subset V$, $r \in T^k$.
- **Objective:** Find $E^0 \subset E$ (purchased in the first-stage) and $E^k \subset E$ (purchased in the second-stage, **if scenario** k **is realized**), for all $k \in K$ such that expected solution cost is minimized, i.e.:

$$\min \sum_{e \in E^0} c_e^0 + \sum_{k \in K} p^k \sum_{e \in E^k} c_e^k$$

$$\text{s.t. } E^0 \cup E^k \text{ spans } T^k, \quad \forall k \in K$$

WHAT IS KNOWN ABOUT SSTP SO FAR?

Previous Work

- introduced by Gupta et al. [2007a] (approximation and complexity results)
- approximation algorithms [Gupta and Pál, 2005, Gupta et al., 2004, 2007b, Swamy and Shmoys, 2006]
  - In general, SSTP is NP-hard to approximate within a constant factor. Constant approximation possible only for special cases.
- fixed-parameter tractability [Kurz et al., 2013]
- heuristics [Hokama et al., 2014] (**genetic algorithm, DIMACS Challenge 2014**)
- exact two-stage branch-and-cut based on **Benders decomposition**:
  - stochastic STP [Bomze et al., 2010],
  - stochastic survivable network design [Ljubić et al., 2017],
  - PhD thesis Bernd Zey (upcoming 2017).

Our Contribution

- we introduce a new ILP formulation for the SSTP
  - strongest among existing formulations
- we design a solution framework based on this formulation
  - exploits the **decomposability** of the formulation in various ways

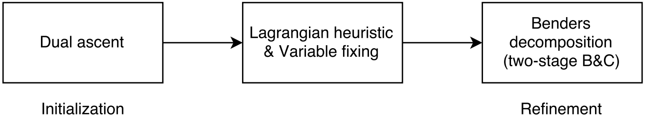


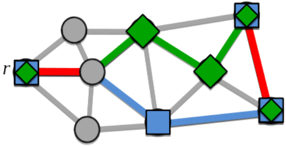
Figure: Algorithmic framework.

- we present a computational study comparing our approach with
  - state-of-the-art exact approach from [Bomze et al., 2010, Ljubić et al., 2017] (Benders decomposition based on two-stage branch-and-cut)
  - genetic algorithm from [Hokama et al., 2014]
- presented method significantly outperforms these approaches

STEP 1:
A STRONGER FORMULATION

Two Semi-Directed Models for SSTP [Bomze et al., 2010, Zey, 2016, Ljubić et al., 2017]

It is impossible to orient the first-stage solution, so we derive semi-directed formulations.



- **1st Stage:** undirected $\longrightarrow x_e^0$
- **2nd Stage:** directed $\longrightarrow y_{ij}^k$

$SSTP_{sd1}$

$$\min \sum_{e \in E} c_e^0 x_e^0 + \sum_k p^k \sum_{e=\{i,j\} \in E} c_e^k (y_{ij}^k + y_{ji}^k)$$

$$x^0(\delta(S)) + y^k(\delta^-(S)) \geq 1 \qquad \forall k, \forall S$$

$$x_e^0, y_{ij}^k \in \{0,1\}$$

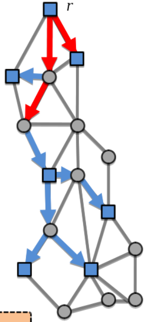
$$\min \sum_{e \in E} c_e^0 x_e^0 + \sum_k p^k \sum_{e=\{i,j\} \in E} c_e^k (y_{ij}^k + y_{ji}^k - x_e^0)$$

$$y_{ij}^k + y_{ji}^k \geq x_e^0 \qquad \forall e = \{i,j\} \in E$$

$$y^k(\delta^-(S)) \geq 1 \qquad \forall k, \forall S$$

$$x_e^0, y_{ij}^k \in \{0,1\}$$

$SSTP_{sd2}$



Hierarchy of Formulations

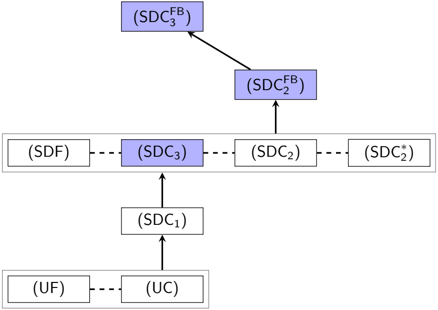


Figure: Directed arcs indicate that the target formulation is stronger than the source formulation. Blue boxes: the formulation has been introduced by us, all the others are from Bomze et al. [2010], Zey [2016]

Flow-Balance constraints (FB):

- strengthening: ensure, that only terminals can be leaf-nodes
- added to (SDC$_2$) from Bomze et al. [2010], Zey [2016] $\rightarrow$ (SDC$_2^{FB}$)
- added to our (SDC$_3$) $\rightarrow$ (SDC$_3^{FB}$)

(SDC_3): A Strong Formulation for SSTP

- **idea**: Steiner arborscence rooted at r for each $k \in K$, using arcs bought in first and second stage
  - binary $w_{ij}^k = 1$, iff arc (i,j) is selected in the first stage for scenario k
  - binary $z_{ij}^k = 1$, iff arc (i,j) is selected in the second stage for scenario k
  - binary $x_e = 1$, iff edge e is selected in the first stage
- $\mathcal{W}^k$: set of directed **Steiner cuts** for scenario k

$$\min \quad \sum_{e \in E} c_e^0 x_e + \sum_{k \in K} p^k \sum_{e=\{i,j\} \in E} c_e^k (z_{ij}^k + z_{ji}^k)$$

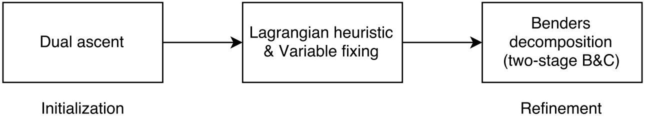
$$\text{s.t.} \quad w^k(\delta^-(W)) + z^k(\delta^-(W)) \geq 1 \qquad \forall W \in \mathcal{W}^k, \forall k \in K \qquad (SDC_3{:}1)$$

$$w_{ij}^k + w_{ji}^k \leq x_e \qquad \forall e = \{i,j\} \in E, \forall k \in K \qquad (SDC_3{:}2)$$

$$(\mathbf{x}, \mathbf{z}, \mathbf{w}) \in \{0,1\}^{|E|+2|A||K|} \qquad (SDC_3{:}3)$$

The Framework

Advantages of (SDC$_3$): It decomposes nicely, and gives the strongest bounds with (SDC$_3^{FB}$).



How does it work?

1. **Dual ascent:** greedy heuristic that changes dual multipliers λ while monotonically increasing LB. Gives also an UB.
2. **Lagrangian:** takes UB and final λ from DA to initialize the subgradient method. Improves UB and LB. Applies **reduction techniques.** Generates a collection of useful dual multipliers λ.
3. **Benders:** takes UB and optimality cuts associated to Langrangian λ found during the subgradient procedure.

OBSERVE: Steps 1 and 2 give valid LB and UB and are purely combinatorial (no MIP solver needed!) Step 3 is a branch-and-cut (CPLEX).

STEP 2: DUAL ASCENT

Dual Ascent

- let $\boldsymbol{\beta}$ and $\boldsymbol{\lambda}$ be the dual multipliers of (SDC_3:1) (connectivity) and (SDC_3:2) (linking)

$$(\mathrm{SDC}_3^D) \qquad \max \sum_{k \in K} \sum_{W \in \mathcal{W}^k} \beta_W^k$$

$$\sum_{k \in K} \lambda_e^k \leq c_e^0 \qquad \forall e \in E \qquad (\mathrm{SDC}_3^D{:}1)$$

$$\beta(\mathcal{W}_{ij}^k) \leq p^k c_e^k \qquad \forall (i,j) \in A, \forall k \in K, e = \{i,j\} \qquad (\mathrm{SDC}_3^D{:}2)$$

$$\beta(\mathcal{W}_{ij}^k) - \lambda_e^k \leq 0 \qquad \forall (i,j) \in A, \forall k \in K, e = \{i,j\} \qquad (\mathrm{SDC}_3^D{:}3)$$

$$(\boldsymbol{\beta}^k, \boldsymbol{\lambda}^k) \in \mathbb{R}_{\geq 0}^{|\mathcal{W}^k| + |E|} \qquad \forall k \in K$$

- dual ascent works similar to dual ascent for STP Wong [1984]
  - start from initial solution $\bar{\boldsymbol{\beta}} = \mathbf{0}$
  - each iteration: increase one dual variable $\beta_W^k = 0$ while preserving feasibility
  - **The worst-case time complexity:** $\mathcal{O}(\sum_{k \in K} |A| \min\{|A|, |T^k||V|\})$.

STEP 3: LAGRANGIAN HEURISTIC

Lagrangian Relaxation

- relax constraints (SDC$_3$:2) using **Lagrangian dual multipliers** $\boldsymbol{\lambda} \geq 0$
- we obtain the relaxation

$$L(\boldsymbol{\lambda}) := \min \Big\{ \sum_{e \in E} c_e^0 x_e + \sum_{k \in K} p^k \sum_{e=\{i,j\} \in E} c_e^k (z_{ij}^k + z_{ji}^k) + \sum_{k \in K} \sum_{e=\{i,j\} \in E} \lambda_e^k (w_{ij}^k + w_{ji}^k - x_e) : (\text{SDC}_3{:}1), (\text{SDC}_3{:}3) \Big\}$$

- define **Lagrangian cost** as $\tilde{c}_e := c_e^0 - \sum_{k \in K} \lambda_e^k, e \in E$
- problem decomposes into $|K| + 1$ independent subproblems
  - one in $\mathbf{x}$

$$L^0(\boldsymbol{\lambda}) := \min \Big\{ \sum_{e \in E} \tilde{c}_e x_e : \mathbf{x} \in \{0,1\}^{|E|} \Big\}$$

  - and one in $\mathbf{z}^k, \mathbf{w}^k$ for $k \in K$

$$L^k(\boldsymbol{\lambda}) := \min \Big\{ \sum_{e=\{i,j\} \in E} \Big[p^k c_e^k (z_{ij}^k + z_{ji}^k) + \lambda_e^k (w_{ij}^k + w_{ji}^k) \Big] : (\text{SDC}_3{:}1), (\mathbf{z}^k, \mathbf{w}^k) \in \{0,1\}^{2|A|} \Big\}$$

Lagrangian Relaxation

- the **Lagrangian dual problem** is

$$(\mathrm{SDC}_3^{LD}) \qquad \max_{\boldsymbol{\lambda} \geq \mathbf{0}} \left\{ L^0(\boldsymbol{\lambda}) + \sum_{k \in K} L^k(\boldsymbol{\lambda}) \right\}$$

- $L^0(\boldsymbol{\lambda})$ can be computed by inspection
- $L^k(\boldsymbol{\lambda})$: solving an instance of the Steiner arborescence problem (SAP)

Theorem

$v(LP\text{-}SDC_3^{FB}) \leq v(SDC_3^{LD}) = v(SDC_3)$

- we solve (SDC_3^{LD}) using a subgradient scheme
- **dual variables at the end of the dual ascent are used to initialize $\boldsymbol{\lambda}$**
- subproblems $L^k(\boldsymbol{\lambda})$ are solved heuristically
  - using a dual ascent for SAP together with a primal heuristic
- two different heuristics to calculate high-quality feasible solutions
- we designed **reduction tests** to fix nodes and edges

STEP 4: BENDERS DECOMPOSITION

Benders Decomposition

- in the spirit of the two-stage B&C approach introduced in Bomze et al. [2010] for (SDC$_2$).
- **Benders master problem** is stated as follows

$$(\text{SDC}_3^B)\ \min \quad \sum_{e \in E} c_e^0 x_e + \sum_{k \in K} p^k \theta^k$$

$$\text{s.t.} \qquad \theta^k \geq \Phi^k(\mathbf{x}) \quad \forall k \in K \qquad (\text{SDC}_3^B{:}1)$$

$$\mathbf{x} \in \{0,1\}^{|E|}, \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^{|K|}$$

- variables **z** and **w** associated to the second stage projected out
- $\boldsymbol{\theta}^k \geq 0$: second-stage cost for each scenario
- for each $k \in K$ and first-stage solution $\bar{\mathbf{x}}$, the recourse function $\Phi^k(\bar{\mathbf{x}})$ gives the corresponding second-stage cost
- dynamically separated **fractional** and **integral Benders optimality cuts** are used in order to underestimate the value of $\Phi^k(\bar{\mathbf{x}})$

Benders Decomposition

- **Benders subproblem** is another Steiner arborescence problem
- Benders cuts

$$\theta^k \geq \sum_{W \in \mathcal{W}^k} \bar{\beta}^k_W - \sum_{e \in E} \bar{\lambda}^k_e x_e \qquad \forall k \in K \qquad (\mathrm{SDC}^B_3\text{:FRAC})$$

  where $\bar{\lambda}^k$ and $\bar{\beta}^k$ are (optimal) dual multipliers of the LP-relaxation of the Benders subproblem.
- **Lagrangian optimality cuts:**
  - initialize the master problem using optimality cuts derived from high-quality Lagrangian multipliers ($\bar{\lambda}^k = \lambda^k$ and $\bar{\beta}^k = \frac{1}{p_k}\beta^k$)
- **Integer optimality cuts**
  - $\Phi^k(\bar{\mathbf{x}})$ is an STP, solved using the exact solver by Fischetti et al. [2017]
  - let $E^0_S = \{e \in E : \bar{x}_e = 1\}$, optimality cuts are defined as

$$\theta^k \geq \Phi^k(\bar{\mathbf{x}}) - \sum_{e \in E \setminus E^0_S} c^k_e x_e \qquad \forall k \in K \qquad (\mathrm{SDC}^B_3\text{:INT})$$

COMPUTATIONAL RESULTS

Implementation Details and Benchmark Instances

- implemented in C++
- Benders decomposition: CPLEX 12.7 is used as a ILP solver
- single-threaded on an Intel Xeon CPU E5-2670v2 (2.5 GHz)
- time limit of one hour and a memory limit of 6 GB
- instances from the [SSTPLib] (used in the 11th DIMACS Implementation Challenge); denoted as **SMALL**
- also generated new large-scale benchmark instances from real-world STP instances [Leitner et al., 2014]; denoted as **LARGE**

Table: Basic properties of our benchmark instances.

		$\|V\|$			$\|E\|$			$\|K\|$		
dataset	**inst**[#]	*min*	*avg*	*max*	*min*	*avg*	*max*	*min*	*avg*	*max*
K100	154	22	31	45	64	115	191	5	272	1000
P100	70	66	77	91	163	194	237	5	272	1000
LIN01-10	140	53	190	321	80	318	540	5	272	1000
WRP	196	10	194	311	149	363	613	5	272	1000
VIENNA	40	1991	5756	9574	3176	9347	16208	5	21	50

Effects of the Dual Ascent Initialization

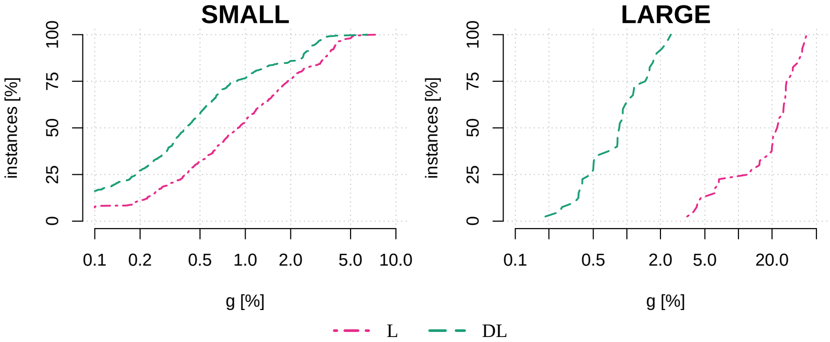


Figure: Optimality gap charts for SMALL and LARGE instances with dual ascent initialization of the subgradient algorithm (DL) and without (L).

Effects of the Benders Decomposition

- gap at the end of the root node

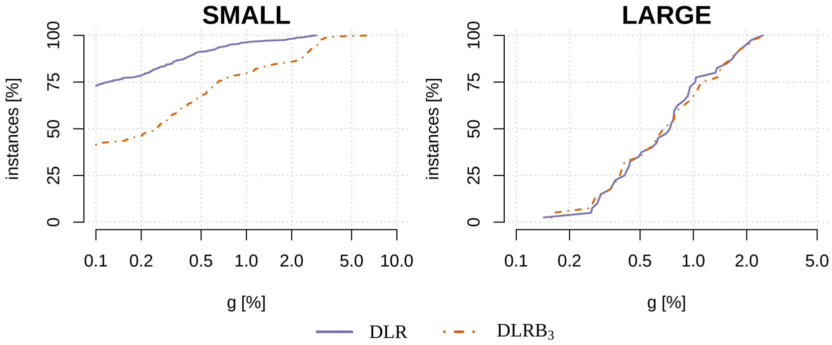


Figure: Optimality gap charts at the end of the root node for SMALL and LARGE with ($DLRB_3$) and without (DLR) Benders decomposition applied as a refinement procedure.

Comparison with the State-of-the-Art

- re-implemented Benders approach of Bomze et al. [2010], denoted as B_2

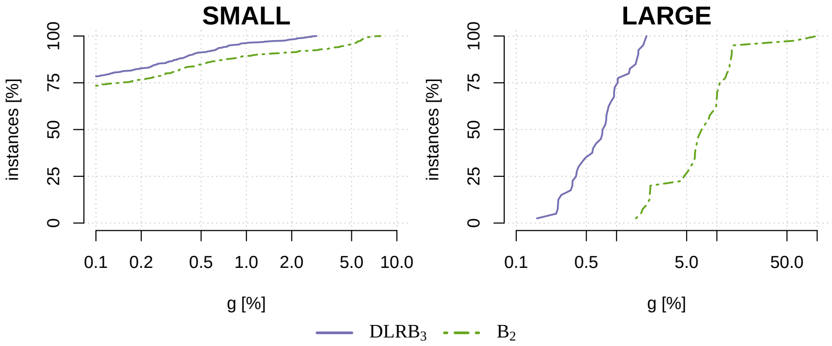


Figure: Optimality gap charts comparing $DLRB_3$ and B_2.

Comparison with the State-of-the-Art

- H: heuristic of Hokama et al. [2014] are denoted
  - done in C++; obtained on an Intel Xeon CPU E3-1230 V2, (3.30GHz)
- *Pg*: primal gap, t_b: time to best solution

Table: Results on datasets K100 (all solved to optimality by $DLRB_3$ and B_2, columns *Pg*[%] are thus omitted).

	t[s]		Pg[%]	t_b[s]		
$\|K\|$	$DLRB_3$	B_2	H	$DLRB_3$	B_2	H
5	1	1	2.31	**0**	1	1
10	1	1	0.86	1	1	1
20	2	2	0.68	1	1	2
50	3	3	0.81	2	2	5
75	4	5	0.55	**2**	4	8
100	5	5	0.58	**3**	4	11
150	9	**8**	0.57	6	6	16
200	13	**12**	0.52	**8**	9	23
250	**15**	16	0.55	**6**	11	28
300	19	**17**	0.88	**9**	14	30
400	27	**22**	0.72	**15**	18	40
500	32	**28**	0.60	18	18	57
750	**44**	47	0.66	**26**	36	93
1000	68	**61**	0.82	**32**	35	121

Further Reading

References:

- M. Leitner, I. Ljubić, M. Luipersbeck, M. Sinnl, **Decomposition methods for the two-stage stochastic Steiner tree problem**, technical report, 2017
  `http://homepage.univie.ac.at/ivana.ljubic/research/publications/da-TR.pdf`

Our additional work on dual ascent for Steiner trees:

- M. Leitner, I. Ljubić, M. Luipersbeck, M. Sinnl, **A dual-ascent-based branch-and-bound framework for the prize-collecting Steiner tree and related problems**, INFORMS Journal on Computing, 2017, to appear
- code available at `https://github.com/mluipersbeck/dapcstp`

Literature I

I. Bomze, M. Chimani, M. Jünger, I. Ljubić, P. Mutzel, and B. Zey. Solving two-stage stochastic Steiner tree problems by two-stage branch-and-cut. In **International Symposium on Algorithms and Computation**, pages 427–439. Springer, 2010.

M. Fischetti, M. Leitner, I. Ljubić, M. Luipersbeck, M. Monaci, M. Resch, D. Salvagnin, and M. Sinnl. Thinning out Steiner trees: a node-based model for uniform edge costs. **Mathematical Programming Computation**, 9(2): 203–229, 2017. ISSN 1867-2957. doi: 10.1007/s12532-016-0111-0. URL http://dx.doi.org/10.1007/s12532-016-0111-0.

A. Gupta and M. Pál. Stochastic Steiner trees without a root. **Automata, Languages and Programming**, pages 100–100, 2005.

A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In **Proceedings of the thirty-sixth annual ACM symposium on Theory of computing**, pages 417–426. ACM, 2004.

A. Gupta, M. Hajiaghayi, and A. Kumar. Stochastic Steiner tree with non-uniform inflation. In **Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques**, pages 134–148. Springer, 2007a.

Literature II

A. Gupta, R. Ravi, and A. Sinha. LP rounding approximation algorithms for stochastic network design. **Mathematics of Operations Research**, 32(2): 345–364, 2007b.

P. Hokama, M. C. San Felice, E. C. Bracht, and F. L. Usberti. A heuristic approach for the stochastic Steiner tree problem. 11th DIMACS Challenge workshop, 2014.

D. Kurz, P. Mutzel, and B. Zey. Parameterized algorithms for stochastic Steiner tree problems. In **Mathematical and Engineering Methods in Computer Science**, volume 7721 of **LNCS**, pages 143–154. Springer, 2013.

M. Leitner, I. Ljubic, M. Luipersbeck, M. Prossegger, and M. Resch. New real-world instances for the Steiner tree problem in graphs. Technical report, 2014.

I. Ljubić, P. Mutzel, and B. Zey. Stochastic survivable network design problems: Theory and practice. **European Journal of Operational Research**, 256(2): 333–348, 2017.

SSTPLib. `https://ls11-www.cs.uni-dortmund.de/staff/zey/sstp/`. Accessed at: 2017-04-24.

Literature III

C. Swamy and D. B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. **ACM SIGACT News**, 37(1):33–46, 2006.

R. T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. **Mathematical Programming**, 28(3):271–287, 1984. ISSN 0025-5610.

B. Zey. ILP formulations for the two-stage stochastic Steiner tree problem. 2016.