Models and Algorithms for the Balance-Constrained Stochastic Bottleneck Spanning Tree Problem

Jue Wang[1] Siqian Shen[1] Murat Kurt[2]

[1]Department of Industrial and Operations Engineering
University of Michigan

[2]Department of Industrial and Systems Engineering
University at Buffalo (State University of New York)

The 13th INFORMS Computing Society Conference
January 08, 2013

Outline

- Introduction
- Basic and MINLP formulations for the BCSBSTP
- SOS1- and SOS2-based formulations and algorithm
- SAA-based MILP formulation
- Computational results

What is the BCSBSTP?

- BCSBSTP: Balance-Constrained Stochastic Bottleneck Spanning Tree Problem (a stochastic MST problem)
- Each edge weight is characterized by a probability distribution; all weights are independently distributed.
- Goal: minimize an upper bound imposed on the maximum edge weight in a spanning tree with certain probability.
- "Balanced-Constrained" implies an additional chance constraint on the minimum edge weight in a spanning tree.
- SBSTP: A special case of the BCSBSTP without bounding the minimum edge weight.

Applications

- Telecommunication, e.g., wireless sensor networks
- Post-disaster relief
- Epidemic spread
- Network reliability

Previous work

- Ishii and Nishida (1983) studied the SBSTP with normally and independently distributed edge weights.
- Ishii and Shiode (1995) continued to discuss variants and extensions of the SBSTP.
- Kurt (2012) proposed a polynomial-time approximation for solving the generalized SBSTP and showed that
  1. the exact optimal solution can be obtained when edge weights have the same distribution type,
  2. BCSBSTP is in general NP-Complete.

Notation

- Graph Configuration

$G = (V, E)$	An undirected connected graph.
$\mathcal{T}(G)$	Set of all spanning trees of graph G.
$T = (V, E_T)$	A spanning tree of G.
w_j	Random edge weight for every edge $e_j \in E$.

Notation

- Graph Configuration

$G = (V, E)$	An undirected connected graph.
$\mathcal{T}(G)$	Set of all spanning trees of graph G.
$T = (V, E_T)$	A spanning tree of G.
w_j	Random edge weight for every edge $e_j \in E$.

- Decision Variable

ℓ	an upper bound variable on the maximum edge weight.

Notation

- Graph Configuration

$G = (V, E)$	An undirected connected graph.
$\mathcal{T}(G)$	Set of all spanning trees of graph G.
$T = (V, E_T)$	A spanning tree of G.
w_j	Random edge weight for every edge $e_j \in E$.

- Decision Variable

ℓ	an upper bound variable on the maximum edge weight.

- Parameters

κ	a given lower bound on the minimum edge weight.
α, β	probability levels associated with the upper and lower bound chance constraints, respectively.

Basic formulation for the BCSBSTP

$$Q := \min_{T\in \mathcal{T}(G)} \left\{ \ell : \Pr\left(\max_{j:e_j\in E_T} w_j \leq \ell \right) \geq \alpha,\ \Pr\left(\min_{j:e_j\in E_T} w_j \geq \kappa \right) \geq \beta \right\}, \qquad (1)$$

Basic formulation for the BCSBSTP

$$Q := \min_{T \in \mathcal{T}(G)} \left\{ \ell : \Pr\left(\max_{j: e_j \in E_T} w_j \leq \ell \right) \geq \alpha,\ \Pr\left(\min_{j: e_j \in E_T} w_j \geq \kappa \right) \geq \beta \right\}, \qquad (1)$$

Because all distributions are independent, we have

$$\Pr\left(\max_{j:\ e_j \in E_T} w_j \leq \ell \right) \geq \alpha \Leftrightarrow \prod_{j:\ e_j \in E_T} F_j(\ell) \geq \alpha \Leftrightarrow \sum_{j:\ e_j \in E_T} \log F_j(\ell) \geq \log \alpha, \quad \text{and}$$

$$\Pr\left(\min_{j:\ e_j \in E_T} w_j \geq \kappa \right) \geq \beta \Leftrightarrow \prod_{j:\ e_j \in E_T} \left[1 - F_j(\kappa)\right] \geq \beta \Leftrightarrow \sum_{j:\ e_j \in E_T} \log \left[1 - F_j(\kappa)\right] \geq \log \beta,$$

Basic formulation for the BCSBSTP

$$Q := \min_{T \in \mathcal{T}(G)} \left\{ \ell : \Pr\left(\max_{j: e_j \in E_T} w_j \leq \ell \right) \geq \alpha,\ \Pr\left(\min_{j: e_j \in E_T} w_j \geq \kappa \right) \geq \beta \right\}, \qquad (1)$$

Because all distributions are independent, we have

$$\Pr\left(\max_{j:\ e_j \in E_T} w_j \leq \ell \right) \geq \alpha \Leftrightarrow \prod_{j:\ e_j \in E_T} F_j(\ell) \geq \alpha \Leftrightarrow \sum_{j:\ e_j \in E_T} \log F_j(\ell) \geq \log \alpha, \quad \text{and}$$

$$\Pr\left(\min_{j:\ e_j \in E_T} w_j \geq \kappa \right) \geq \beta \Leftrightarrow \prod_{j:\ e_j \in E_T} \left[1 - F_j(\kappa)\right] \geq \beta \Leftrightarrow \sum_{j:\ e_j \in E_T} \log\left[1 - F_j(\kappa)\right] \geq \log \beta,$$

which transform Problem Q into an equivalent nonlinear problem:

$$Q' := \min_{T \in \mathcal{T}(G)} \left\{ \ell : \sum_{j: e_j \in E_T} \log F_j(\ell) \geq \log \alpha,\ \sum_{j: e_j \in E_T} \log\left[1 - F_j(\kappa)\right] \geq \log \beta \right\}. \qquad (2)$$

MINLP formulation for the BCSBSTP

Introduce new decision variables : $x_j = \begin{cases} 1 & \text{if edge } e_j \in E_T, \\ 0 & \text{otherwise.} \end{cases}$

$$
\begin{aligned}
\text{min:} \quad & \ell \\
\text{s.t.} \quad & \sum_{j:e_j \in E} x_j \log F_j(\ell) \geq \log \alpha && \text{(3a)} \\
& \sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta && \text{(3b)} \\
& \sum_{j:e_j \in E} x_j = n - 1 && \text{(3c)} \\
& \sum_{j:e_j \in E_{V_\mathrm{s}}} x_j \leq |V_\mathrm{s}| - 1 \quad \forall V_\mathrm{s} \subset V, \; V_\mathrm{s} \neq \emptyset && \text{(3d)} \\
& x_j \in \{0, 1\} \quad \forall e_j \in E && \text{(3e)}
\end{aligned}
$$

SOS1-based formulation

Special Ordered Sets of type 1 (SOS1): a set of variables, at most one of which can take a strictly positive value with all others being at 0.

SOS1-based formulation

Special Ordered Sets of type 1 (SOS1): a set of variables, at most one of which can take a strictly positive value with all others being at 0.

Define binary variables

$$z_k = \begin{cases} 1 \\ 0 \end{cases}$$

$$\sum_{k=1}^{n} z_k = 1 \tag{4c}$$

$$z_k \in \{0, 1\} \quad \forall k = 1, \ldots, n. \tag{4d}$$

SOS1-based formulation

Special Ordered Sets of type 1 (SOS1): a set of variables, at most one of which can take a strictly positive value with all others being at 0.

Define binary variables

$$z_k = \begin{cases} 1 & \text{if } \ell = \ell_k, \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{min:} \quad \sum_{k=1}^{n} z_k \ell_k$$

$$\sum_{k=1}^{n} z_k = 1 \tag{4c}$$

$$z_k \in \{0, 1\} \quad \forall k = 1, \ldots, n. \tag{4d}$$

SOS1-based formulation

Special Ordered Sets of type 1 (SOS1): a set of variables, at most one of which can take a strictly positive value with all others being at 0.

Define binary variables

$$z_k = \begin{cases} 1 & \text{if } \ell = \ell_k, \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned}
\text{min:} \quad & \sum_{k=1}^{n} z_k \ell_k \\
\text{s.t.} \quad & \text{(3c)–(3e)} \\
& \sum_{j: e_j \in E} x_j \log F_j(\ell) \geq \log \alpha && \text{(4a)} \\
& \sum_{j: e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta && \text{(4b)} \\
& \sum_{k=1}^{n} z_k = 1 && \text{(4c)} \\
& z_k \in \{0, 1\} \quad \forall k = 1, \ldots, n. && \text{(4d)}
\end{aligned}$$

SOS1-based formulation

Special Ordered Sets of type 1 (SOS1): a set of variables, at most one of which can take a strictly positive value with all others being at 0.

Define binary variables

$$z_k = \begin{cases} 1 & \text{if } \ell = \ell_k, \\ 0 & \text{otherwise.} \end{cases}$$

$$
\begin{aligned}
\text{min:} \quad & \sum_{k=1}^{n} z_k \ell_k \\
\text{s.t.} \quad & \text{(3c)–(3e)} \\
& \sum_{j:e_j \in E} \sum_{k=1}^{n} z_k x_j \log F_j(\ell_k) \geq \log \alpha \quad \text{(4a)} \\
& \sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta \quad \text{(4b)} \\
& \sum_{k=1}^{n} z_k = 1 \quad \text{(4c)} \\
& z_k \in \{0, 1\} \quad \forall k = 1, \ldots, n. \quad \text{(4d)}
\end{aligned}
$$

SOS1-based formulation

Compute the upper bound $\overline{\ell}$ and lower bound $\underline{\ell}$ of ℓ *a priori*, dissect the whole interval equally, treat each sample point as parameter.

$$
\begin{aligned}
\text{min:} \quad & \sum_{k=1}^{n} z_k \ell_k \\
\text{s.t.} \quad & \text{(3c)–(3e)} \\
& \sum_{j:e_j \in E} \sum_{k=1}^{n} z_k x_j \log F_j(\ell_k) \geq \log \alpha \\
& \sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta \\
& \sum_{k=1}^{n} z_k = 1 \\
& z_k \in \{0, 1\} \quad \forall k = 1, \ldots, n.
\end{aligned}
$$

SOS1-based formulation

Compute the upper bound $\overline{\ell}$ and lower bound $\underline{\ell}$ of ℓ *a priori*, dissect the whole interval equally, treat each sample point as parameter.

$$
\begin{aligned}
\text{min:} \quad & \sum_{k=1}^{n} z_k \ell_k \\
\text{s.t.} \quad & \text{(3c)–(3e)} \\
& \sum_{j:e_j \in E} \sum_{k=1}^{n} z_k x_j \log F_j(\ell_k) \geq \log \alpha \\
& \sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta \\
& \sum_{k=1}^{n} z_k = 1 \\
& z_k \in \{0, 1\} \quad \forall k = 1, \ldots, n.
\end{aligned}
$$

SOS1-based formulation

Compute the upper bound $\overline{\ell}$ and lower bound $\underline{\ell}$ of ℓ *a priori*, dissect the whole interval equally, treat each sample point as parameter.

$$
\begin{aligned}
\text{min:} \quad & \sum_{k=1}^{n} z_k \ell_k \\
\text{s.t.} \quad & \text{(3c)–(3e)} \\
& \sum_{j:e_j \in E} \sum_{k=1}^{n} z_k x_j \log F_j(\ell_k) \geq \log \alpha \\
& \sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta \\
& \sum_{k=1}^{n} z_k = 1 \\
& z_k \in \{0, 1\} \quad \forall k = 1, \ldots, n.
\end{aligned}
$$

SOS1-based formulation

Compute the upper bound $\overline{\ell}$ and lower bound $\underline{\ell}$ of ℓ *a priori*, dissect the whole interval equally, treat each sample point as parameter.

$$
\begin{aligned}
\text{min:} \quad & \sum_{k=1}^{n} z_k \ell_k \\
\text{s.t.} \quad & \text{(3c)–(3e)} \\
& \sum_{j:e_j \in E} \sum_{k=1}^{n} z_k x_j \log F_j(\ell_k) \geq \log \alpha \\
& \sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta \\
& \sum_{k=1}^{n} z_k = 1 \\
& z_k \in \{0,1\} \quad \forall k = 1, \ldots, n.
\end{aligned}
$$

SOS1-based formulation

Compute the upper bound $\bar{\ell}$ and lower bound $\underline{\ell}$ of ℓ *a priori*, dissect the whole interval equally, treat each sample point as parameter.

$$
\begin{aligned}
\text{min:} \quad & \sum_{k=1}^{n} z_k \ell_k \\
\text{s.t.} \quad & \text{(3c)–(3e)} \\
& \sum_{j:e_j \in E} \sum_{k=1}^{n} z_k x_j \log F_j(\ell_k) \geq \log \alpha \\
& \sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa) \right] \geq \log \beta \\
& \sum_{k=1}^{n} z_k = 1 \\
& z_k \in \{0, 1\} \quad \forall k = 1, \ldots, n.
\end{aligned}
$$

SOS1-based formulation

Introduce o_{kj} to replace bilinear terms $z_k x_j, \forall k = 1, \ldots, n, \; e_j \in E$;

$$\text{min:} \quad \sum_{k=1}^{n} z_k \ell_k$$

s.t. (3c)–(3e),(4c),(4d)

$$\sum_{j:e_j \in E} \sum_{k=1}^{n} z_k x_j \log F_j(\ell_k) \geq \log \alpha \tag{6a}$$

$$\sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta \tag{6b}$$

SOS1-based formulation

Introduce o_{kj} to replace bilinear terms $z_k x_j$, $\forall k = 1, \ldots, n,\ e_j \in E$;

$$\begin{aligned} \text{min:} \quad & \sum_{k=1}^{n} z_k \ell_k \\ \text{s.t.} \quad & \text{(3c)–(3e),(4c),(4d)} \\ & \sum_{j:e_j \in E} \sum_{k=1}^{n} o_{kj} \log F_j(\ell_k) \geq \log \alpha && \text{(6a)} \\ & \sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta && \text{(6b)} \end{aligned}$$

SOS1-based formulation

Introduce o_{kj} to replace bilinear terms $z_k x_j$, $\forall k = 1, \ldots, n, \; e_j \in E$;
Use McCormic Inequalities to linearize o_{kj}.

$$\text{min:} \quad \sum_{k=1}^{n} z_k \ell_k$$

s.t. (3c)–(3e),(4c),(4d)

$$\sum_{j:e_j \in E} \sum_{k=1}^{n} o_{kj} \log F_j(\ell_k) \geq \log \alpha \tag{6a}$$

$$\sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta \tag{6b}$$

SOS1-based formulation

Introduce o_{kj} to replace bilinear terms $z_k x_j$, $\forall k = 1, \ldots, n,\ e_j \in E$;
Use McCormic Inequalities to linearize o_{kj}.

$$\begin{aligned}
\text{min:} \quad & \sum_{k=1}^{n} z_k \ell_k \\
\text{s.t.} \quad & \text{(3c)–(3e),(4c),(4d)} \\
& \sum_{j:e_j \in E} \sum_{k=1}^{n} o_{kj} \log F_j(\ell_k) \geq \log \alpha && \text{(6a)} \\
& \sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta && \text{(6b)} \\
& o_{kj} \leq z_k \quad \forall k = 1, \ldots, n,\ \forall e_j \in E && \text{(6c)} \\
& o_{kj} \leq x_j \quad \forall k = 1, \ldots, n,\ \forall e_j \in E && \text{(6d)} \\
& o_{kj} \geq z_k + x_j - 1 \quad \forall k = 1, \ldots, n, \forall e_j \in E && \text{(6e)} \\
& o_{kj} \geq 0 \quad \forall k = 1, \ldots, n,\ \forall e_j \in E. && \text{(6f)}
\end{aligned}$$

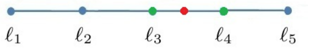
SOS2-based formulation

Special Ordered Sets of type 2 (SOS2) an ordered set of variables, of which at most two can be non-zero, and if two are non-zero these must be consecutive in their ordering.

SOS2-based formulation

Special Ordered Sets of type 2 (SOS2) an ordered set of variables, of which at most two can be non-zero, and if two are non-zero these must be consecutive in their ordering.

Building on SOS1...



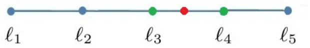
SOS2-based formulation

Special Ordered Sets of type 2 (SOS2) an ordered set of variables, of which at most two can be non-zero, and if two are non-zero these must be consecutive in their ordering.

Define binary variables

$$y_k = \begin{cases} 1 & \text{if } \ell \in [\ell_k, \ell_{k+1}], \\ 0 & \text{otherwise.} \end{cases}$$

Building on SOS1...



$$\sum_{k=1}^{n-1} y_k = 1 \tag{7d}$$

$$y_k \in \{0, 1\} \quad \forall k = 1, \ldots, n \tag{7h}$$

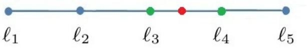
SOS2-based formulation

Special Ordered Sets of type 2 (SOS2) an ordered set of variables, of which at most two can be non-zero, and if two are non-zero these must be consecutive in their ordering.

Define binary variables

$$y_k = \begin{cases} 1 & \text{if } \ell \in [\ell_k, \ell_{k+1}], \\ 0 & \text{otherwise.} \end{cases}$$

Building on SOS1...



$$\sum_{k=1}^{n-1} y_k = 1 \tag{7d}$$

$$\rho_1 \leq y_1 \tag{7e}$$

$$\rho_i \leq y_i + y_{i-1} \quad \forall i = 2, \ldots, n-1 \tag{7f}$$

$$\rho_n \leq y_{n-1} \tag{7g}$$

$$y_k \in \{0, 1\} \quad \forall k = 1, \ldots, n \tag{7h}$$

$$\rho_k \geq 0 \quad \forall k = 1, \ldots, n. \tag{7i}$$

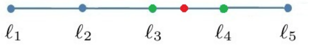
SOS2-based formulation

Special Ordered Sets of type 2 (SOS2) an ordered set of variables, of which at most two can be non-zero, and if two are non-zero these must be consecutive in their ordering.

Define binary variables

$$y_k = \begin{cases} 1 & \text{if } \ell \in [\ell_k, \ell_{k+1}], \\ 0 & \text{otherwise.} \end{cases}$$

Building on SOS1...



If $y_k = 1$, then
$\ell = \rho_k \ell_k + \rho_{k+1} \ell_{k+1}$,
$\rho_k + \rho_{k+1} = 1,\ \rho_k, \rho_{k+1} \geq 0$.

$$\sum_{k=1}^{n-1} y_k = 1 \tag{7d}$$

$$\rho_1 \leq y_1 \tag{7e}$$

$$\rho_i \leq y_i + y_{i-1} \quad \forall i = 2, \ldots, n-1 \tag{7f}$$

$$\rho_n \leq y_{n-1} \tag{7g}$$

$$y_k \in \{0, 1\} \quad \forall k = 1, \ldots, n \tag{7h}$$

$$\rho_k \geq 0 \quad \forall k = 1, \ldots, n. \tag{7i}$$

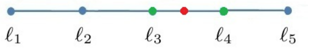
SOS2-based formulation

Special Ordered Sets of type 2 (SOS2) an ordered set of variables, of which at most two can be non-zero, and if two are non-zero these must be consecutive in their ordering.

Define binary variables

$$y_k = \begin{cases} 1 & \text{if } \ell \in [\ell_k, \ell_{k+1}], \\ 0 & \text{otherwise.} \end{cases}$$

Building on SOS1...



If $y_k = 1$, then
$\ell = \rho_k \ell_k + \rho_{k+1} \ell_{k+1}$,
$\rho_k + \rho_{k+1} = 1, \; \rho_k, \rho_{k+1} \geq 0.$

$$\text{min:} \quad \sum_{k=1}^{n} \rho_k \ell_k$$

$$\sum_{k=1}^{n} \rho_k = 1 \tag{7c}$$

$$\sum_{k=1}^{n-1} y_k = 1 \tag{7d}$$

$$\rho_1 \leq y_1 \tag{7e}$$

$$\rho_i \leq y_i + y_{i-1} \quad \forall i = 2, \ldots, n-1 \tag{7f}$$

$$\rho_n \leq y_{n-1} \tag{7g}$$

$$y_k \in \{0, 1\} \quad \forall k = 1, \ldots, n \tag{7h}$$

$$\rho_k \geq 0 \quad \forall k = 1, \ldots, n. \tag{7i}$$

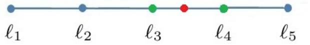
SOS2-based formulation

Special Ordered Sets of type 2 (SOS2) an ordered set of variables, of which at most two can be non-zero, and if two are non-zero these must be consecutive in their ordering.

Define binary variables

$$y_k = \begin{cases} 1 & \text{if } \ell \in [\ell_k, \ell_{k+1}], \\ 0 & \text{otherwise.} \end{cases}$$

Building on SOS1...



If $y_k = 1$, then
$\ell = \rho_k \ell_k + \rho_{k+1} \ell_{k+1}$,
$\rho_k + \rho_{k+1} = 1, \ \rho_k, \rho_{k+1} \geq 0$.

$$\text{min:} \quad \sum_{k=1}^{n} \rho_k \ell_k$$

$$\text{s.t.} \quad \text{(3c)–(3e)}$$

$$\sum_{j:e_j \in E} \sum_{k=1}^{n} \rho_k x_j \log F_j(\ell_k) \geq \log \alpha \quad (7a)$$

$$\sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta \quad (7b)$$

$$\sum_{k=1}^{n} \rho_k = 1 \quad (7c)$$

$$\sum_{k=1}^{n-1} y_k = 1 \quad (7d)$$

$$\rho_1 \leq y_1 \quad (7e)$$

$$\rho_i \leq y_i + y_{i-1} \quad \forall i = 2, \ldots, n-1 \quad (7f)$$

$$\rho_n \leq y_{n-1} \quad (7g)$$

$$y_k \in \{0, 1\} \quad \forall k = 1, \ldots, n \quad (7h)$$

$$\rho_k \geq 0 \quad \forall k = 1, \ldots, n. \quad (7i)$$

SOS2-based formulation

Similarly, introduce q_{kj} to replace bilinear terms $\rho_k x_j$, $\forall k = 1, \ldots, n,\ e_j \in E$;
Use McCormic Inequalities to linearize q_{kj}.

$$\text{min:} \quad \sum_{k=1}^{n} z_k \ell_k$$

$$\text{s.t.} \quad \text{(3c)–(3e),(7c)–(7i)}$$

$$\sum_{j:e_j \in E} \sum_{k=1}^{n} q_{kj} \log F_j(\ell_k) \geq \log \alpha \tag{8a}$$

$$\sum_{j:e_j \in E} x_j \log \left[1 - F_j(\kappa)\right] \geq \log \beta \tag{8b}$$

$$q_{kj} \leq \rho_k \quad \forall k = 1, \ldots, n, \ \forall e_j \in E \tag{8c}$$

$$q_{kj} \leq x_j \quad \forall k = 1, \ldots, n, \ \forall e_j \in E \tag{8d}$$

$$q_{kj} \geq \rho_k + x_j - 1 \quad \forall k = 1, \ldots, n, \ \forall e_j \in E \tag{8e}$$

$$q_{kj} \geq 0 \quad \forall k = 1, \ldots, n, \ \forall e_j \in E. \tag{8f}$$

Compute the upper and lower bounds

$$\Pr\left(\max_{j:\ e_j \in\ E_T} w_j \leq \ell\right) \geq \alpha \Leftrightarrow \prod_{j:\ e_j \in\ E_T} F_j(\ell) \geq \alpha.$$

Compute the upper and lower bounds

$$\Pr\left(\max_{j:\ e_j \in E_T} w_j \leq \ell\right) \geq \alpha \Leftrightarrow \prod_{j:\ e_j \in E_T} F_j(\ell) \geq \alpha.$$

Proposition

Let ℓ^* and T^* be the optimal objective value and a corresponding spanning tree to Problem Q. Then

$$\prod_{j:\ e_j \in E_{T^*}} F_j(\ell^*) = \alpha, \tag{9}$$

for any continuous cumulative distribution functions $F_j(\cdot)$ of edge weights w_j, $\forall e_j \in E$.

Compute the upper and lower bounds

Define $F_j^{-1}(\cdot)$ as the inverse cumulative distribution function of edge weight w_j. For both SOS1- and SOS2-based formulations, let

$$\underline{\ell} = \min_{j:e_j \in E} \left\{ F_j^{-1}(\alpha^{1/(|V|-1)}) \right\}, \text{ and} \tag{10a}$$

$$\overline{\ell} = \max_{j:e_j \in E} \left\{ F_j^{-1}(\alpha^{1/(|V|-1)}) \right\}, \tag{10b}$$

then

$$\underline{\ell} \leq \ell^* \leq \overline{\ell}, \text{ and}$$

$$\prod_{j:\ e_j \in E_{T^*}} F_j(\ell^*) = \alpha.$$

Algorithm for SOS1- and SOS2-based formulations

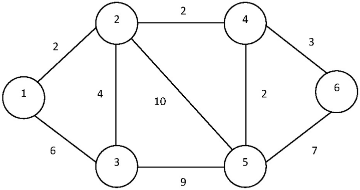
1: Setup a connected undirected graph $G(V, E)$, number of intervals n, probability level α and β and error tolerance Δ.
2: Set the current iteration $t := 0$.
3: Compute $\underline{\ell}^t := \min_{j:e_j \in E} \left\{ F_j^{-1}(\alpha^{1/(|V|-1)}) \right\}$ and $\overline{\ell}^t := \max_{j:e_j \in E} \left\{ F_j^{-1}(\alpha^{1/(|V|-1)}) \right\}$.
4: **repeat**
5: Generate an equally distributed sequence $\{\ell_1^t, \ldots, \ell_n^t\}$ in between interval $[\underline{\ell}^t, \overline{\ell}^t]$.
6: Compute $\log F_j(\ell_k^t)\ \forall e_j \in E,\ k = 1, \ldots, n$.
7: Solve SOS1- or SOS2-based formulation and record the current optimal objective value ℓ^{t^*}.
8: For SOS1, if $\ell^{t^*} = \ell_{k^t}^t$, set $\underline{\ell}^{\mathrm{t+1}} := \ell_{k^t-1}^t$ and $\overline{\ell}^{\mathrm{t+1}} := \ell_{k^t+1}^t$.
For SOS2, if $\ell^{t^*} \in [\ell_{k^t}^t, \ell_{k^t+1}^t]$, set $\underline{\ell}^{\mathrm{t+1}} := \ell_{k^t}^t$ and $\overline{\ell}^{\mathrm{t+1}} := \ell_{k^t+1}^t$.
9: Set $t := t + 1$.
10: **until** $|\ell^{t-1^*} - \ell^{t^*}| \leq \Delta$

An example (using SOS2-based formulation)

Assume that each edge weight in the network follows an exponential distribution such that $w_j \sim \text{Exp}(\lambda_j)$, $j = 1, \ldots, 9$. The number alongside each edge in the figure represents the value of λ_j.

Set $n = 6, \alpha = 0.95$, and error tolerance $\Delta = 0.01$.

$F^{-1}(\lambda_j) = -\ln(1 - 0.95^{1/5})/\lambda_j; \quad F^{-1}(2) \approx 2.292; \quad F^{-1}(10) \approx 0.458.$



Iteration t	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ^{t^*}
0	0.458	0.825	1.192	1.559	1.926	2.292	1.077
1	0.825	0.899	0.972	1.045	1.119	1.192	1.021
2	0.972	0.987	1.001	1.016	1.031	1.045	1.019

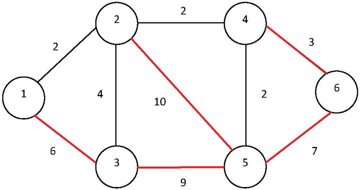
$|1.019 - 1.021| = 0.002 \leq \Delta = 0.01$

An example (using SOS2-based formulation)

Assume that each edge weight in the network follows an exponential distribution such that $w_j \sim \text{Exp}(\lambda_j)$, $j = 1, \ldots, 9$. The number alongside each edge in the figure represents the value of λ_j.

Set $n = 6, \alpha = 0.95$, and error tolerance $\Delta = 0.01$.

$F^{-1}(\lambda_j) = -\ln(1 - 0.95^{1/5})/\lambda_j$; $F^{-1}(2) \approx 2.292$; $F^{-1}(10) \approx 0.458$.



Iteration t	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ^{t^*}	$E_T^{t^*}$
0	0.458	0.825	1.192	1.559	1.926	2.292	1.077	(1, 3) (2, 5) (3, 5) (4, 6) (5, 6)
1	0.825	0.899	0.972	1.045	1.119	1.192	1.021	(1, 3) (2, 5) (3, 5) (4, 6) (5, 6)
2	0.972	0.987	1.001	1.016	1.031	1.045	1.019	(1, 3) (2, 5) (3, 5) (4, 6) (5, 6)

$|1.019 - 1.021| = 0.002 \leq \Delta = 0.01$

An SAA-Based Integer Programming Approximation

Parameters

Ω	a finite set of scenarios.
$\xi = \{w_1, \ldots, w_{\lvert E \rvert}\}$	a random vector, characterized by distributions of w_j, $\forall e_j \in E$.
$\xi^s = \{w_1^s, \ldots, w_{\lvert E \rvert}^s\}$	the realization of ξ in scenario $s \in \Omega$, where values w_j^s are generated from distributions of w_j, $\forall e_j \in E$.

Decision Variables

$\zeta_s \quad \forall s \in \Omega$	$\zeta_s = 1$ if $\max_{j:e_j \in E_T} w_j^s > \ell$, and 0 otherwise.
$\phi_s \quad \forall s \in \Omega$	$\phi_s = 1$ if $\min_{j:e_j \in E_T} w_j^s < \kappa$, and 0 otherwise.

An SAA-Based Integer Programming Approximation

$$Q := \min_{T \in \mathcal{T}(G)} \left\{ \ell : \Pr\left(\max_{j: e_j \in E_T} w_j \leq \ell \right) \geq \alpha,\ \Pr\left(\min_{j: e_j \in E_T} w_j \geq \kappa \right) \geq \beta \right\}.$$

The two chance constraints are rewritten as

$$\Pr\left(\max_{j:\ e_j \in E_T} w_j \leq \ell \right) \geq \alpha \Leftrightarrow \Pr\left(\max_{j:\ e_j \in E_T} w_j > \ell \right) \leq 1 - \alpha \Leftrightarrow \sum_{s \in \Omega} Prob^s \zeta_s \leq (1 - \alpha), \quad \text{and}$$

$$\Pr\left(\min_{j:\ e_j \in E_T} w_j \geq \kappa \right) \geq \beta \Leftrightarrow \Pr\left(\min_{j:\ e_j \in E_T} w_j < \kappa \right) \leq \beta \qquad \Leftrightarrow \sum_{s \in \Omega} Prob^s \phi_s \leq (1 - \beta).$$

An SAA-Based Integer Programming Approximation

Letting $u_s = \max_{j:e_j \in E_T} w_j^s$ and $v_s = \min_{j:e_j \in E_T} w_j^s$ for a spanning tree $E_T = \{e_j \in E : x_j = 1\}$, the SAA-based reformulation of Problem Q is

$$
\begin{aligned}
\text{min:} \quad & \ell \\
\text{s.t.} \quad & \text{(3c)–(3e)} \\
& \sum_{s \in \Omega} Prob^s \zeta_s \le (1-\alpha) && \text{(11a)} \\
& u_s - w_{\max}^s \zeta_s \le \ell \quad \forall s \in \Omega && \text{(11b)} \\
& u_s \ge w_j^s x_j \quad \forall e_j \in E,\ s \in \Omega && \text{(11c)} \\
& \sum_{s \in \Omega} Prob^s \phi_s \le (1-\beta) && \text{(11d)} \\
& v_s + w_{\max}^s \phi_s \ge \kappa \quad \forall s \in \Omega && \text{(11e)} \\
& v_s \le w_j^s x_j \quad \forall e_j \in E,\ s \in \Omega && \text{(11f)} \\
& \zeta_s,\ \phi_s \in \{0,1\} \quad \forall s \in \Omega, && \text{(11g)}
\end{aligned}
$$

An SAA-Based Integer Programming Approximation

Letting $u_s = \max_{j:e_j \in E_T} w_j^s$ and $v_s = \min_{j:e_j \in E_T} w_j^s$ for a spanning tree $E_T = \{e_j \in E : x_j = 1\}$, the SAA-based reformulation of Problem Q is

$$
\begin{aligned}
\text{min:} \quad & \ell \\
\text{s.t.} \quad & \text{(3c)–(3e)} \\
& \sum_{s \in \Omega} Prob^s \zeta_s \leq (1-\alpha) && \text{(11a)} \\
& u_s - w_{\max}^s \zeta_s \leq \ell \quad \forall s \in \Omega && \text{(11b)} \\
& u_s \geq w_j^s x_j \quad \forall e_j \in E,\ s \in \Omega && \text{(11c)} \\
& \sum_{s \in \Omega} Prob^s \phi_s \leq (1-\beta) && \text{(11d)} \\
& v_s + w_{\max}^s \phi_s \geq \kappa \quad \forall s \in \Omega && \text{(11e)} \\
& v_s \leq w_j^s x_j \quad \forall e_j \in E,\ s \in \Omega && \text{(11f)} \\
& \zeta_s,\ \phi_s \in \{0,1\} \quad \forall s \in \Omega, && \text{(11g)}
\end{aligned}
$$

Computational Results

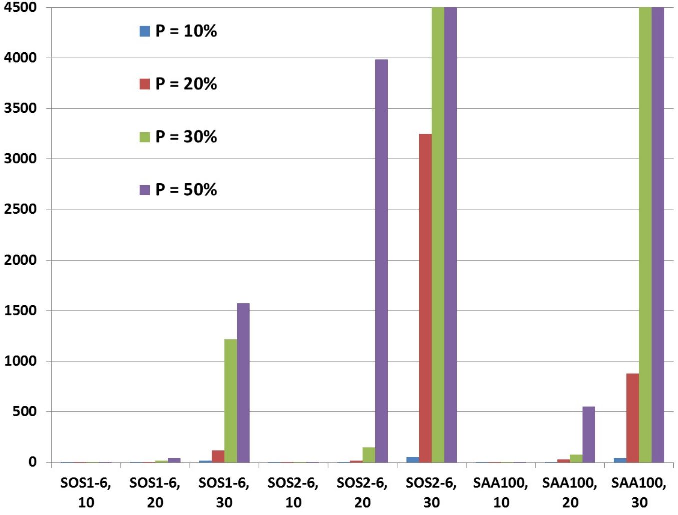
- We reports the computational efficacy of solving the SBSTP by using SOS1, SOS2 and SAA, and solving the BCSBSTP by using SOS1.
- For the SBSTP, we test 12 parameter combinations of graphs, i.e., $\{|V|\} \times \{P\} = \{10, 20, 30\} \times \{10\%, 20\%, 30\%, 50\%\}$, where $|V|$ is the number of nodes in the graph, and P is the graph density.
- For the BCSBSTP, we test one graph type, i.e., graph with 20 nodes and density of 50%, and with varied values of κ.
- We set $\alpha = \beta = 0.95$ and $\Delta = 0.01$.
- All models and algorithms use CPLEX 12.2 via ILOG Concert Technology with C++, and computations are performed on a HP Workstation Z210 Windows 7 machine with Intel(R) Xeon(R) CPU 3.20 GHz, and 8GB memory.
- For each parameter combination, we solve 10 instances.

The SBSTP with different distribution types

Tested Distribution Types and Parameters.

Type	1	2	3
Distribution	Normal	Normal	Normal
Setting	$w_j \sim \mathcal{N}(10, 1)$	$w_j \sim \mathcal{N}(10, 1.5)$	$w_j \sim \mathcal{N}(10, 2)$
Type	4	5	6
Distribution	Exponential	Exponential	Exponential
Setting	$w_j \sim \text{Exp}(0.4)$	$w_j \sim \text{Exp}(0.5)$	$w_j \sim \text{Exp}(0.6)$
Type	7	8	9
Distribution	Uniform	Uniform	Uniform
Setting	$w_j \sim \text{U}(0, 10)$	$w_j \sim \text{U}(0, 12)$	$w_j \sim \text{U}(0, 14)$
Type	10	11	12
Distribution	Chi-Squared	Chi-Squared	Chi-Squared
Setting	$w_j \sim \chi^2(2)$	$w_j \sim \chi^2(3)$	$w_j \sim \chi^2(4)$

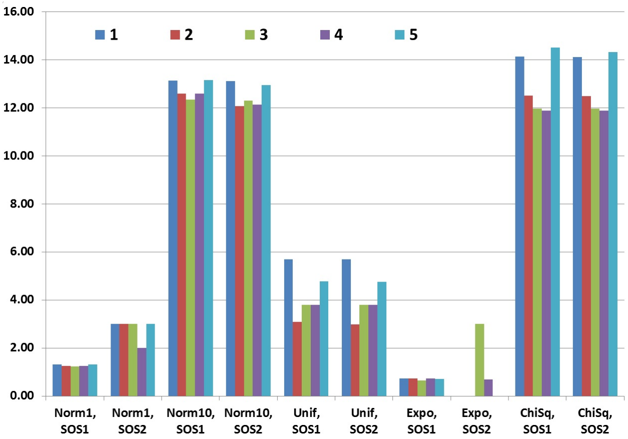
Comparisons of SOS1, SOS2, and SAA for the SBSTP



CPU time of solving the SBSTP with various distributions

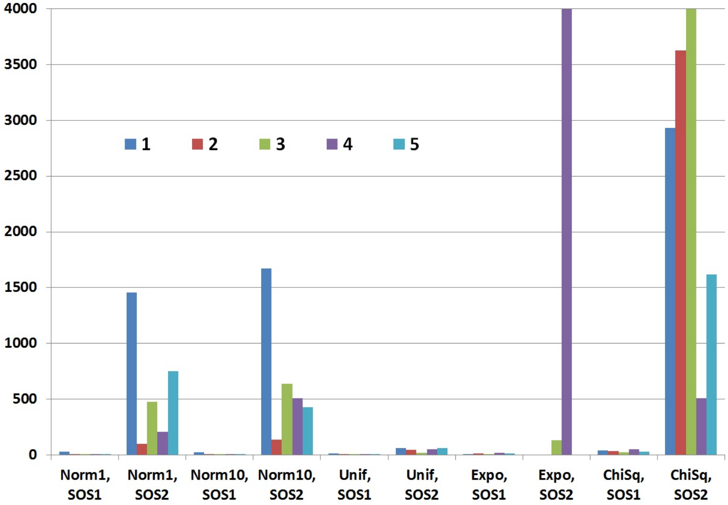
Comparisons of SOS1 and SOS2 for the SBSTP

Type	1	2	3	4	5
Distribution	Chi-Squared	Exponential	Normal1	Normal2	Uniform
Setting	$w_j \sim \chi^2(k_j)$	$w_j \sim \text{Exp}(\lambda_j)$	$w_j \sim \mathcal{N}(10, (0.35\sigma_j)^2)$	$w_j \sim \mathcal{N}(1, (0.035\sigma_j)^2)$	$w_j \sim \text{U}(0, b_j)$



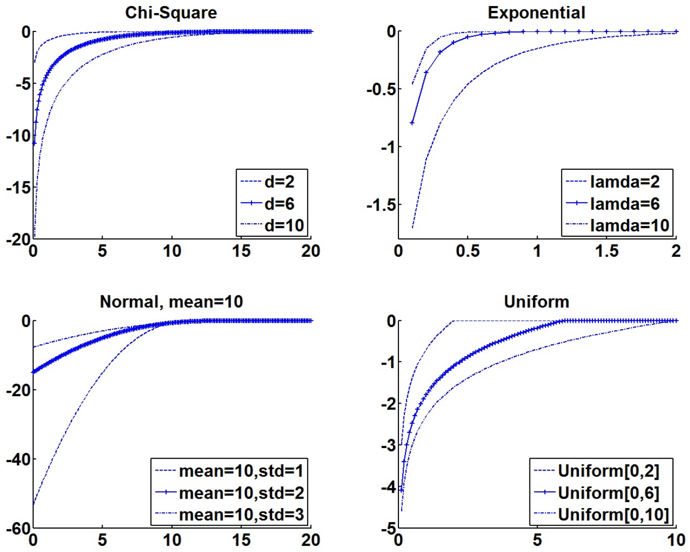
$\{|V|, P\}$=$\{20,50\%\}$, $n = 6$, Same Distribution, Objective Value

Comparisons of SOS1 and SOS2 for the SBSTP



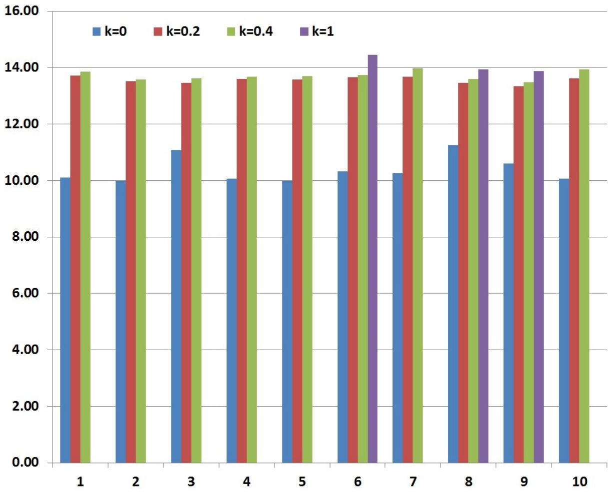
$\{|V|, P\}$=$\{$20,50%$\}$, $n = 6$, Same Distribution, CPU Time (seconds)

Explanation



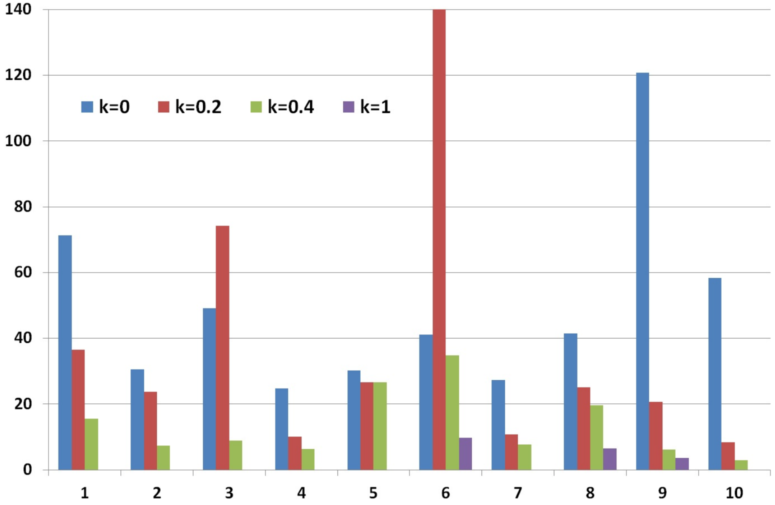
Log Cumulative Distribution Function

Using SOS1 to solve the BCSBSTP (12Types)



Objective value of solving the BCSBSTP, 12Types, $\{|V|, P\} = \{20, 50\%\}$

Using SOS1 to solve the BCSBSTP (12Types)



CPU time of solving the BCSBSTP, 12Types, $\{|V|, P\} = \{20, 50\%\}$

Conclusion & Future Research

Conclusion

- SOS1 is significantly better than the other two approaches in terms of CPU times (without losing too much solution accuracy in all instances we tested).
- Probability distribution types influence computational performances of all three approximations.
- The increase of κ may increase the CPU time of the SOS1 approximation for the BCSBSTP at first but eventually decrease the solution time.

Future Research

Increasing effectiveness of approximation algorithms; seeking tight bounds; incorporating cost and restrictions on spanning tree solutions for special applications; node uncertainty and edge dependencies.