The Shortest Path Interdiction Problem with Improvement

J. Cole Smith Tim Holzmann

Department of Industrial Engineering
Clemson University

October 1, 2018

Contents

1. Introduction
2. Problem Description
3. Algorithm 1: Reformulation
4. Algorithm 2: Recursion
5. Algorithm 3: Best of Both
6. Conclusions

Motivation

- Network interdiction: leader-follower max-min game
  - Maximize shortest path
  - Minimize maximum flow
- Network interdiction problem (NIP) applications
  - Original: Attacks on public infrastructure (Durbin, 1966; Wollmer, 1964)
  - Modern:
    - Nuclear smuggling (Morton et al., 2007)
    - Border protection (Sullivan et al., 2014)
    - Human trafficking (Konrad et al., 2017)
- Assumptions valid?
  - Information symmetry
    - Leader lacks info (Borrero et al., 2015; Lunday and Sherali, 2010)
    - Follower lacks info (Salmerón, 2012; Sullivan et al., 2014)

Multiobjective (MO) Fundamentals

Let $X \subseteq \mathbb{Z}^n$ and $\mathbf{z} : X \to \mathbb{R}^{p+1}$. Define $Z = \{\mathbf{z} \in \mathbb{R}^{p+1} : \mathbf{z} = \mathbf{z}(\mathbf{x}) \text{ for some } x \in X\}$. We adopt the following definitions for Pareto-optimal solutions to max problems.

Objective Space (Z)	**Decision Space (X)**
Dominance: Let $\mathbf{z}^a, \mathbf{z}^b \in Z$. Then $\mathbf{z}^a \geqq \mathbf{z}^b \Leftrightarrow z_k^a \geq z_k^b,\ \forall k \in P.$ $\mathbf{z}^a \geq \mathbf{z}^b \Leftrightarrow \mathbf{z}^a \geqq \mathbf{z}^b \text{ and } \mathbf{z}^a \neq \mathbf{z}^b.$ $\mathbf{z}^a > \mathbf{z}^b \Leftrightarrow z_k^a > z_k^b,\ \forall k \in P.$	
Non-Dominated (ND): We say $\mathbf{z} \in Z$ is $ND \Leftrightarrow \nexists \mathbf{z}' \in Z : \mathbf{z}' \geq \mathbf{z}$	**Efficient**: We say $\mathbf{x} \in X$ is $efficient \Leftrightarrow \mathbf{z}(\mathbf{x}) \text{ is ND in } Z.$
ND Set: $Z_N = \{z \in Z : z \text{ is ND}\}$.	**Eff. Set**: $X_E = \{\mathbf{x} \in X : \mathbf{z}(\mathbf{x}) \in Z_N\}$. **Minimal Complete Eff. Set** (X_{CE}): 1. $\bigcup_{\mathbf{x} \in X_{CE}} \mathbf{z}(\mathbf{x}) = Z_N$ 2. $\mathbf{x}^a, \mathbf{x}^b \in X_{CE} \Rightarrow \mathbf{z}(\mathbf{x}^a) \neq \mathbf{z}(\mathbf{x}^b)$

Contents

1. Introduction
2. Problem Description
3. Algorithm 1: Reformulation
4. Algorithm 2: Recursion
5. Algorithm 3: Best of Both
6. Conclusions

Shortest Path Interdiction Problem (SPIP) with Improvement (SPIP-I)

- Instance is defined by:
  - Digraph: $G = (N, A)$
  - Cost functions: $\mathbf{c}_{ij} : \mathbb{Z}_+^2 \to \mathbb{R}_+, \ \forall (i,j) \in A$
  - Source/End nodes: $s, t \in N$
  - Interdiction/Improvement budgets: $k^L, k^F \in \mathbb{Z}_+$
  - Parameter: $p \geq k^F$

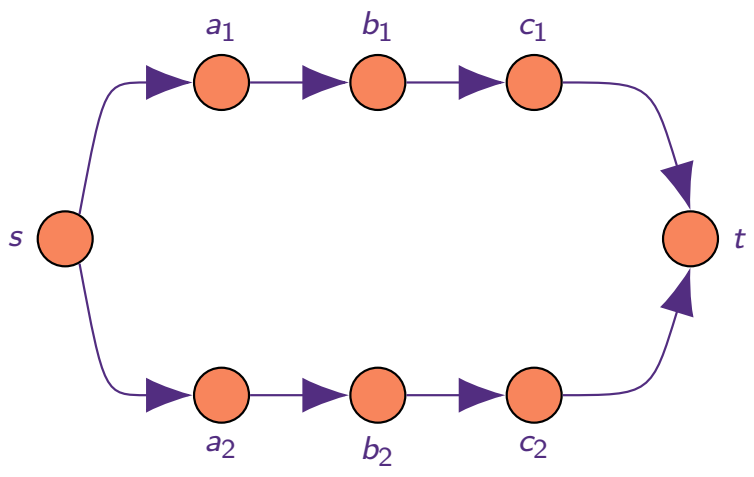
Shortest Path Interdiction Problem (SPIP) with Improvement (SPIP-I)

- Instance is defined by:
  - Digraph: $G = (N, A)$
  - Cost functions: $\mathbf{c}_{ij} : \mathbb{Z}_+^2 \to \mathbb{R}_+, \ \forall (i,j) \in A$
  - Source/End nodes: $s, t \in N$
  - Interdiction/Improvement budgets: $k^L, k^F \in \mathbb{Z}_+$
  - Parameter: $p \geq k^F$
- Game Flow:
  1. Leader selects interdiction strategy, $\mathbf{x}^L \in \mathbb{Z}_+^{|A|}$
  2. Follower selects improvement strategy, $\mathbf{x}^F \in \mathbb{Z}_+^{|A|}$
  3. Follower selects shortest $s \to t$ path

Shortest Path Interdiction Problem (SPIP) with Improvement (SPIP-I)

- Instance is defined by:
  - Digraph: $G = (N, A)$
  - Cost functions: $\mathbf{c}_{ij} : \mathbb{Z}_+^2 \to \mathbb{R}_+, \ \forall (i,j) \in A$
  - Source/End nodes: $s, t \in N$
  - Interdiction/Improvement budgets: $k^L, k^F \in \mathbb{Z}_+$
  - Parameter: $p \geq k^F$
- Game Flow:
  1. Leader selects interdiction strategy, $\mathbf{x}^L \in \mathbb{Z}_+^{|A|}$
  2. Follower selects improvement strategy, $\mathbf{x}^F \in \mathbb{Z}_+^{|A|}$
  3. Follower selects shortest $s \to t$ path
- Leader knows everything <u>except k^F</u>; only that $k^F \in \{0, \ldots, p\}$

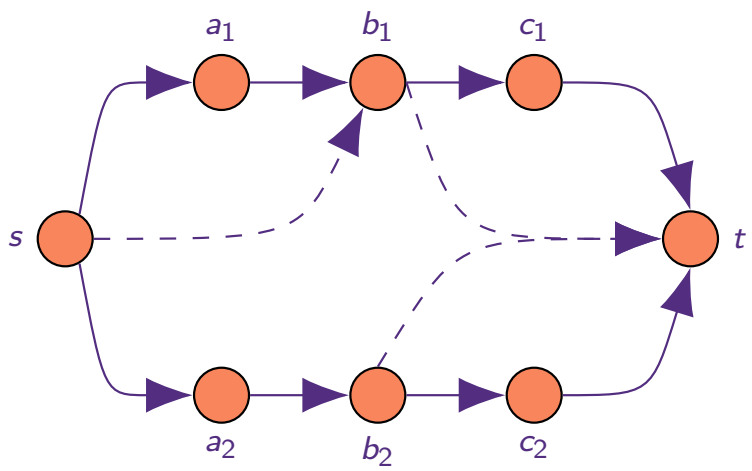
SPIP-I Example



SPIP-I Example.

- Solid arcs: Interdiction-only

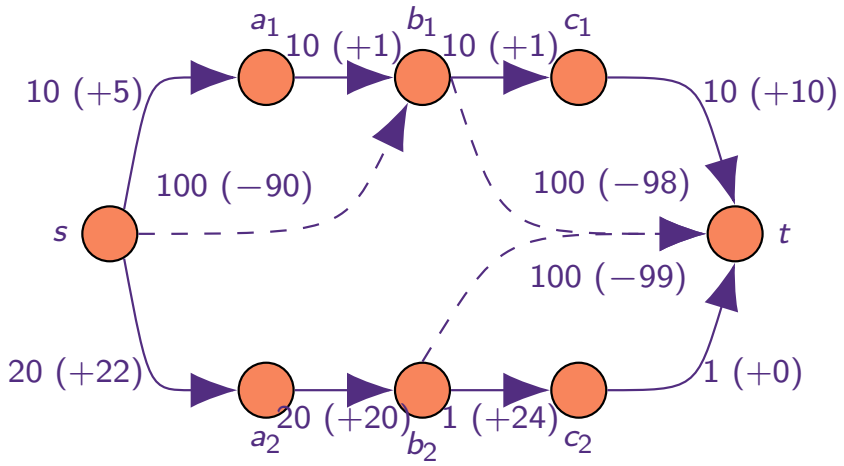
SPIP-I Example



SPIP-I Example.

- Solid arcs: Interdiction-only
- Dotted arcs: Improvement-only

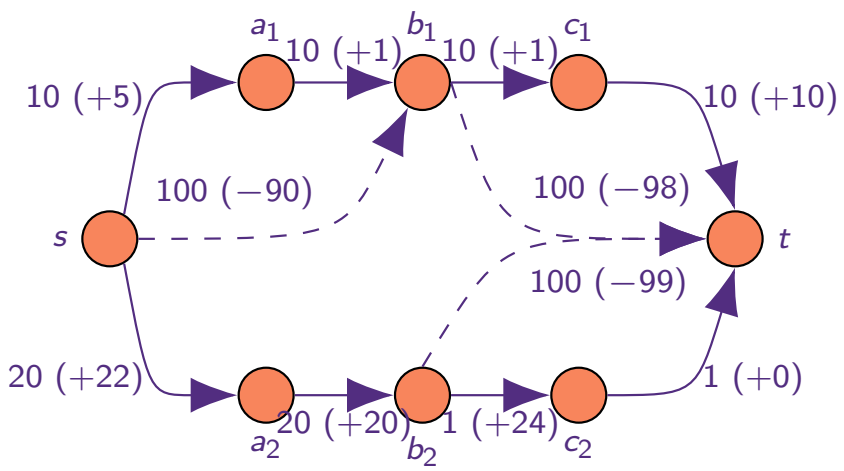
SPIP-I Example



SPIP-I Example.

- Solid arcs: Interdiction-only
- Dotted arcs: Improvement-only
- Arc costs: B_{ij} $(\pm\delta_{ij})$

SPIP-I Example

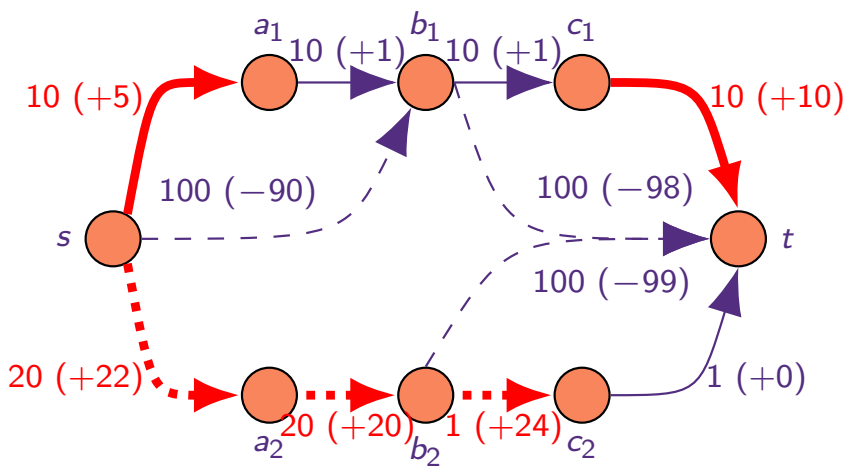


SPIP-I Example.

- Solid arcs: Interdiction-only
- Dotted arcs: Improvement-only
- Arc costs: B_{ij} $(\pm\delta_{ij})$
- $k^L = 3$, $p = 1$
- $X_E = \{\ldots\}$
- $Z_N = \{\ldots\}$

$\mathbf{x}^L$	$\mathbf{y}$ for $k^F = 0$ $(z_0(\mathbf{x}))$	$\mathbf{y}$ for $k^F = 1$ $(z_1(\mathbf{x}))$	**Domination**

SPIP-I Example

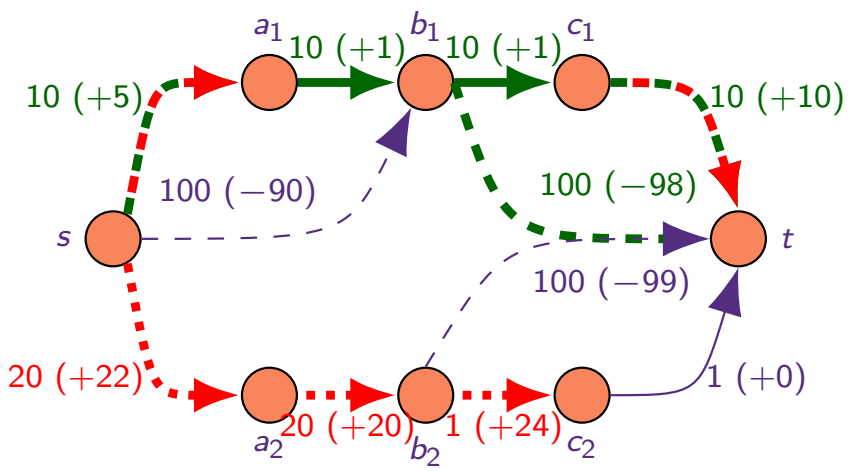


SPIP-I Example.

- Solid arcs: Interdiction-only
- Dotted arcs: Improvement-only
- Arc costs: B_{ij} $(\pm\delta_{ij})$
- $k^L = 3$, $p = 1$
- $X_E = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \ldots\}$
- $Z_N = \{\ldots\}$

$\mathbf{x}^L$	$\mathbf{y}$ for $k^F = 0$ $(z_0(\mathbf{x}))$	$\mathbf{y}$ for $k^F = 1$ $(z_1(\mathbf{x}))$	**Domination**
$\mathbf{x}^1 \equiv \{(s, a_1), (s, a_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^2 \equiv \{(s, a_1), (a_2, b_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^3 \equiv \{(s, a_1), (b_2, c_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated

SPIP-I Example

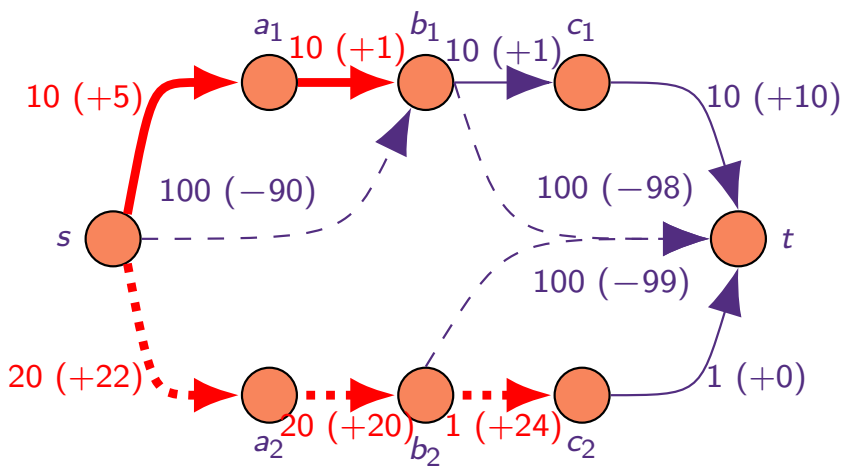


SPIP-I Example.

- Solid arcs: Interdiction-only
- Dotted arcs: Improvement-only
- Arc costs: B_{ij} $(\pm\delta_{ij})$
- $k^L = 3$, $p = 1$
- $X_E = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \ldots\}$
- $Z_N = \{(55, 27), \ldots\}$

$\mathbf{x}^L$	$\mathbf{y}$ for $k^F = 0$ $(z_0(\mathbf{x}))$	$\mathbf{y}$ for $k^F = 1$ $(z_1(\mathbf{x}))$	**Domination**
$\mathbf{x}^1 \equiv \{(s, a_1), (s, a_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^2 \equiv \{(s, a_1), (a_2, b_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^3 \equiv \{(s, a_1), (b_2, c_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated

SPIP-I Example

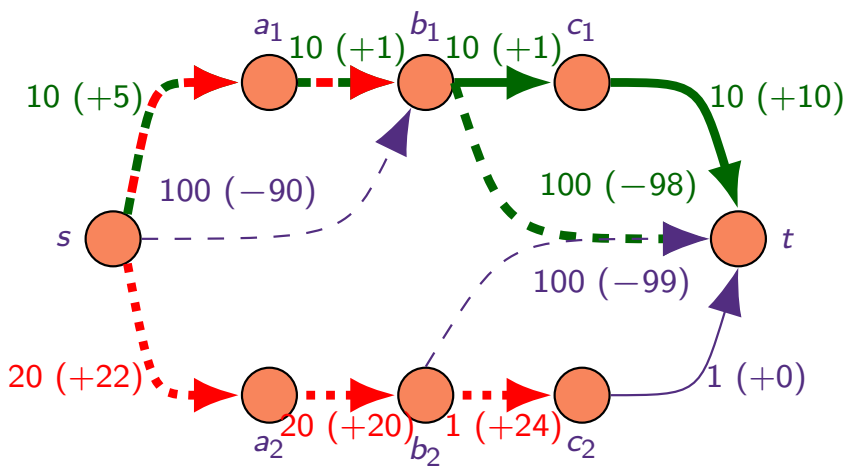


SPIP-I Example.

- Solid arcs: Interdiction-only
- Dotted arcs: Improvement-only
- Arc costs: B_{ij} $(\pm\delta_{ij})$
- $k^L = 3$, $p = 1$
- $X_E = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4, \mathbf{x}^5, \mathbf{x}^6\}$
- $Z_N = \{(55, 27), \ldots\}$

$\mathbf{x}^L$	$\mathbf{y}$ for $k^F = 0$ $(z_0(\mathbf{x}))$	$\mathbf{y}$ for $k^F = 1$ $(z_1(\mathbf{x}))$	**Domination**
$\mathbf{x}^1 \equiv \{(s, a_1), (s, a_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^2 \equiv \{(s, a_1), (a_2, b_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^3 \equiv \{(s, a_1), (b_2, c_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^4 \equiv \{(s, a_1), (s, a_2), (a_1, b_1)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated
$\mathbf{x}^5 \equiv \{(s, a_1), (a_1, b_1), (a_2, b_2)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated
$\mathbf{x}^6 \equiv \{(s, a_1), (a_1, b_1), (b_2, c_2)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated

SPIP-I Example

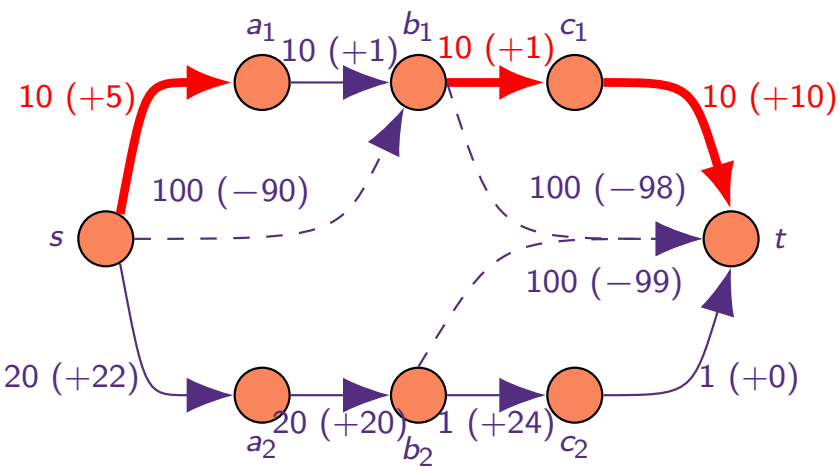


SPIP-I Example.

- Solid arcs: Interdiction-only
- Dotted arcs: Improvement-only
- Arc costs: B_{ij} $(\pm\delta_{ij})$
- $k^L = 3$, $p = 1$
- $X_E = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4, \mathbf{x}^5, \mathbf{x}^6\}$
- $Z_N = \{(55, 27), (46, 28)\}$

$\mathbf{x}^L$	$\mathbf{y}$ for $k^F = 0$ $(z_0(\mathbf{x}))$	$\mathbf{y}$ for $k^F = 1$ $(z_1(\mathbf{x}))$	**Domination**
$\mathbf{x}^1 \equiv \{(s, a_1), (s, a_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^2 \equiv \{(s, a_1), (a_2, b_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^3 \equiv \{(s, a_1), (b_2, c_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^4 \equiv \{(s, a_1), (s, a_2), (a_1, b_1)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated
$\mathbf{x}^5 \equiv \{(s, a_1), (a_1, b_1), (a_2, b_2)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated
$\mathbf{x}^6 \equiv \{(s, a_1), (a_1, b_1), (b_2, c_2)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated

SPIP-I Example

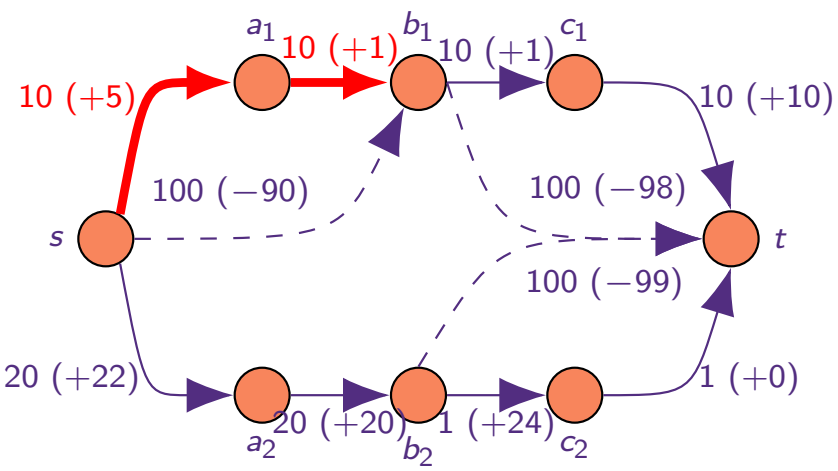


SPIP-I Example.

- Solid arcs: Interdiction-only
- Dotted arcs: Improvement-only
- Arc costs: B_{ij} $(\pm\delta_{ij})$
- $k^L = 3$, $p = 1$
- $X_E = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4, \mathbf{x}^5, \mathbf{x}^6\}$
- $Z_N = \{(55, 27), (46, 28)\}$

$\mathbf{x}^L$	$\mathbf{y}$ for $k^F = 0$ $(z_0(\mathbf{x}))$	$\mathbf{y}$ for $k^F = 1$ $(z_1(\mathbf{x}))$	**Domination**
$\mathbf{x}^1 \equiv \{(s, a_1), (s, a_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^2 \equiv \{(s, a_1), (a_2, b_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^3 \equiv \{(s, a_1), (b_2, c_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^4 \equiv \{(s, a_1), (s, a_2), (a_1, b_1)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated
$\mathbf{x}^5 \equiv \{(s, a_1), (a_1, b_1), (a_2, b_2)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated
$\mathbf{x}^6 \equiv \{(s, a_1), (a_1, b_1), (b_2, c_2)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated
$\mathbf{x}^7 \equiv \{(s, a_1), (b_1, c_1), (c_1, t)\}$	$s \to a_2 \to b_2 \to c_2 \to t$ (42)	$s \to a_1 \to b_1 \to t$ (27)	$\mathbf{z}(\mathbf{x}^4) > \mathbf{z}(\mathbf{x}^7)$

SPIP-I Example



SPIP-I Example.

- Solid arcs: Interdiction-only
- Dotted arcs: Improvement-only
- Arc costs: B_{ij} $(\pm\delta_{ij})$
- $k^L = 3$, $p = 1$
- $X_E = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4, \mathbf{x}^5, \mathbf{x}^6\}$
- $Z_N = \{(55, 27), (46, 28)\}$

$\mathbf{x}^L$	$\mathbf{y}$ for $k^F = 0$ $(z_0(\mathbf{x}))$	$\mathbf{y}$ for $k^F = 1$ $(z_1(\mathbf{x}))$	**Domination**
$\mathbf{x}^1 \equiv \{(s, a_1), (s, a_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^2 \equiv \{(s, a_1), (a_2, b_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^3 \equiv \{(s, a_1), (b_2, c_2), (c_1, t)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (55)	$s \to a_1 \to b_1 \to t$ (27)	Non-dominated
$\mathbf{x}^4 \equiv \{(s, a_1), (s, a_2), (a_1, b_1)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated
$\mathbf{x}^5 \equiv \{(s, a_1), (a_1, b_1), (a_2, b_2)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated
$\mathbf{x}^6 \equiv \{(s, a_1), (a_1, b_1), (b_2, c_2)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (46)	$s \to a_1 \to b_1 \to t$ (28)	Non-dominated
$\mathbf{x}^7 \equiv \{(s, a_1), (b_1, c_1), (c_1, t)\}$	$s \to a_2 \to b_2 \to c_2 \to t$ (42)	$s \to a_1 \to b_1 \to t$ (27)	$\mathbf{z}(\mathbf{x}^4) > \mathbf{z}(\mathbf{x}^7)$
$\mathbf{x}^8 \equiv \{(s, a_1), (a_1, b_1)\}$	$s \to a_1 \to b_1 \to c_1 \to t$ (42)	$s \to a_1 \to b_1 \to t$ (28)	$\mathbf{z}(\mathbf{x}^4) \geq \mathbf{z}(\mathbf{x}^8)$

SPIPI Formulation

$$\max_{\mathbf{x}^L} \quad z_{kF}(\mathbf{x}^L) \qquad \left(\rightarrow \textbf{Multi-obj: } \max_{\mathbf{x}^L} \quad [z_0(\mathbf{x}^L), \ldots, z_p(\mathbf{x}^L)]\right) \tag{L's Obj.}$$

$$\text{subject to:} \quad \sum_{(i,j)\in A} x_{ij}^L \leq k^L \tag{Int. Budget}$$

$$x_{ij}^L \in \mathbb{Z}_+ \quad \forall (i,j) \in A \tag{Int. Dec. Var.}$$

SPIPI Formulation

$$\max_{\mathbf{x}^L} \quad z_{k^F}(\mathbf{x}^L) \qquad \left(\rightarrow \textbf{Multi-obj:} \max_{\mathbf{x}^L} \quad [z_0(\mathbf{x}^L), \ldots, z_p(\mathbf{x}^L)]\right) \qquad \text{(L's Obj.)}$$

$$\text{subject to:} \quad \sum_{(i,j)\in A} x_{ij}^L \leq k^L \qquad \text{(Int. Budget)}$$

$$x_{ij}^L \in \mathbb{Z}_+ \quad \forall (i,j) \in A \qquad \text{(Int. Dec. Var.)}$$

$$\text{where } z_{k^F}(\mathbf{x}^L) = \min_{(\mathbf{x}^F, \mathbf{y})} \quad \sum_{(i,j)\in A} c_{ij}(x_{ij}^L, x_{ij}^F) y_{ij} \qquad \text{(F's Obj.)}$$

$$\text{subject to:} \quad \sum_{\substack{j\in N \\ (i,j)\in A}} y_{ij} - \sum_{\substack{j\in N \\ (j,i)\in A}} y_{ji} = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N \qquad \text{(Flow Constr.)}$$

$$\sum_{(i,j)\in A} x_{ij}^F \leq k^F \qquad \text{(Imp. Budget)}$$

$$y_{ij} \geq 0 \quad \forall (i,j) \in A \qquad \text{(Path Dec. Var.)}$$

$$x_{ij}^F \in \mathbb{Z}_+ \quad \forall (i,j) \in A \qquad \text{(Imp. Dec. Var.)}$$

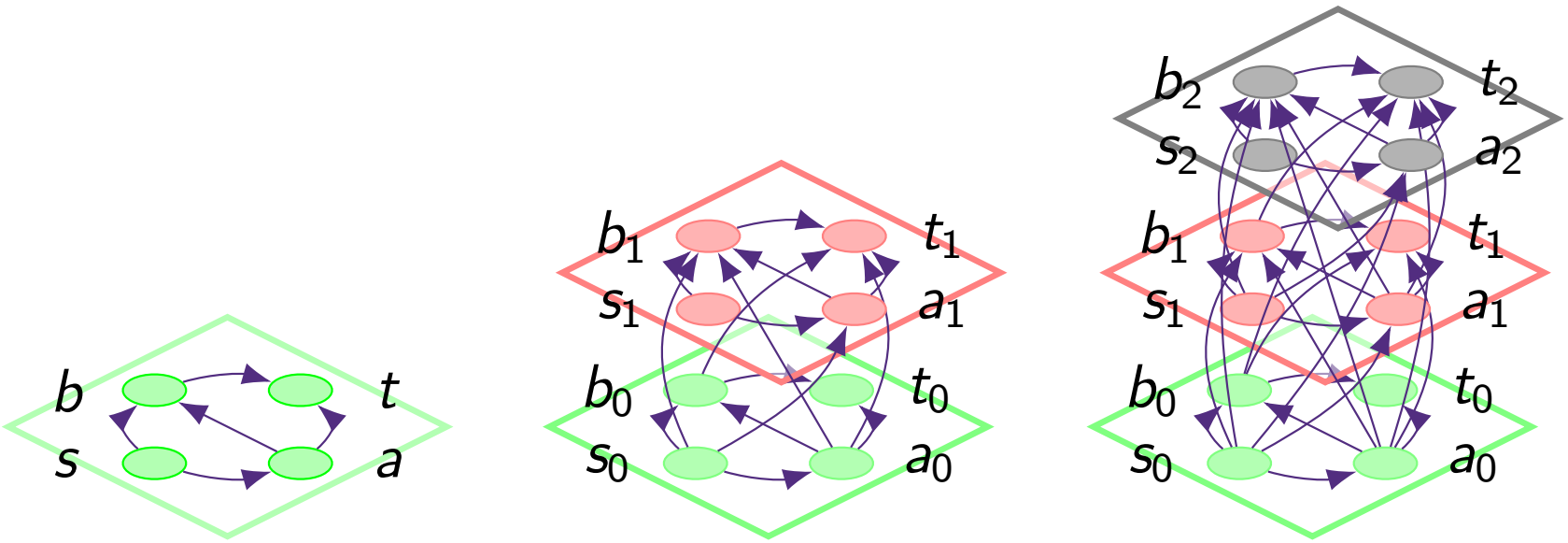
Contents

1. Introduction
2. Problem Description
3. Algorithm 1: Reformulation
4. Algorithm 2: Recursion
5. Algorithm 3: Best of Both
6. Conclusions

Reformulation ($\bar{G}$)

Define $\bar{G} = (\bar{N}, \bar{A})$ with

- $\bar{N} = \{n_h : n \in N,\ h \in \{0, \ldots, p\}\}$.
- $\bar{A} = \{(i_h, j_{h+\ell}) : (i,j) \in A,\ h \in \{0, \ldots, p\},\ \ell \in \{0, \ldots, p-h\}\}$.



(a) $G = (N, A)$ (b) $\bar{G} = (\bar{N}, \bar{A})$ for $p = 1$ (c) $\bar{G} = (\bar{N}, \bar{A})$ for $p = 2$

Reformulation (Bi-level MILP)

$$\max_{\mathbf{x}^L} \quad z_{k^F}(\mathbf{x}^L) \qquad \left(\rightarrow \textbf{MO:} \max_{\mathbf{x}^L} \quad [z_0(\mathbf{x}^L), \ldots, z_p(\mathbf{x}^L)]\right) \qquad \text{(L's Obj.)}$$

$$\text{subject to:} \quad \sum_{(i,j)\in A} x_{ij}^L \leq k^L \qquad \text{(Int. Budget)}$$

$$x_{ij}^L \in \mathbb{Z}_+ \quad \forall (i,j) \in A \qquad \text{(Int. Dec. Var.)}$$

Reformulation (Bi-level MILP)

$$\max_{\mathbf{x}^L} \quad z_{k^F}(\mathbf{x}^L) \qquad \left(\rightarrow \textbf{MO:} \max_{\mathbf{x}^L} \quad [z_0(\mathbf{x}^L), \ldots, z_p(\mathbf{x}^L)]\right) \qquad \text{(L's Obj.)}$$

$$\text{subject to:} \quad \sum_{(i,j)\in A} x_{ij}^L \leq k^L \qquad \text{(Int. Budget)}$$

$$x_{ij}^L \in \mathbb{Z}_+ \quad \forall (i,j) \in A \qquad \text{(Int. Dec. Var.)}$$

$$\text{where } z_{k^F}(\mathbf{x}^L) = \min_{\bar{\mathbf{y}}} \quad \sum_{(i_h, j_{h+\ell})\in \bar{A}} c_{ij}(x_{ij}^L, \ell)\bar{y}_{i_h j_{h+\ell}} \qquad \text{(F's Obj.)}$$

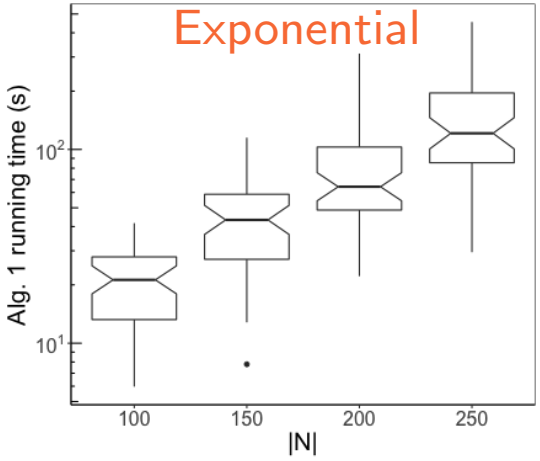
$$\text{subject to:} \quad \sum_{\substack{j\in\bar{N} \\ (i,j)\in\bar{A}}} \bar{y}_{ij} - \sum_{\substack{j\in\bar{N} \\ (j,i)\in\bar{A}}} \bar{y}_{ji} = \begin{cases} 1, & i = s_0 \\ -1, & i = t_{k^F}, \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in \bar{N} \qquad \text{(Flow Constr.)}$$

$$\bar{y}_{ij} \geq 0, \quad \forall (i,j) \in \bar{A} \qquad \text{(Path Dec. Var.)}$$

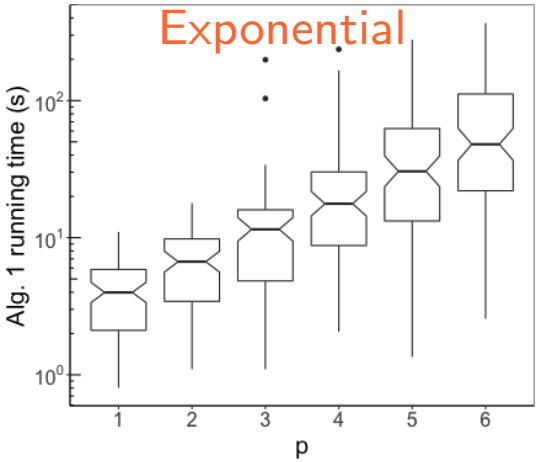
Algorithm 1

- **Reformulate**: Multi-Obj. SPIP-I (G) $\rightarrow$ Multi-Obj. SPIP ($\bar{G}$)
- **Use generating algorithm** to find efficient strategies
  - ϵ-constraint (Kirlik and Sayin, 2014; Özlen et al., 2014)
  - 2-phase (Przybylski et al., 2010; Ulungu and Teghem, 1994),
  - B-&-B (Mavrotas and Diakoulaki, 2005; Stidsen et al., 2014; Vincent et al., 2013)
  - *MAWT Norm* (Holzmann and Smith, 2018)
- **Use SPIP Algorithms** for sub-problems:
  - MILP (Smith, 2010)
  - *Row generation* (Israeli and Wood, 2002; Lozano and Smith, 2017)

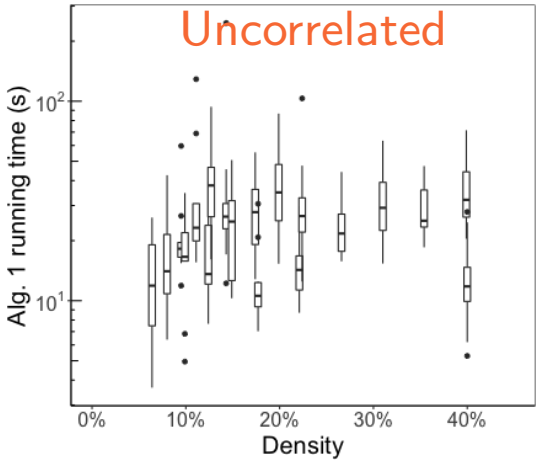
Computation Times for Algorithm 1



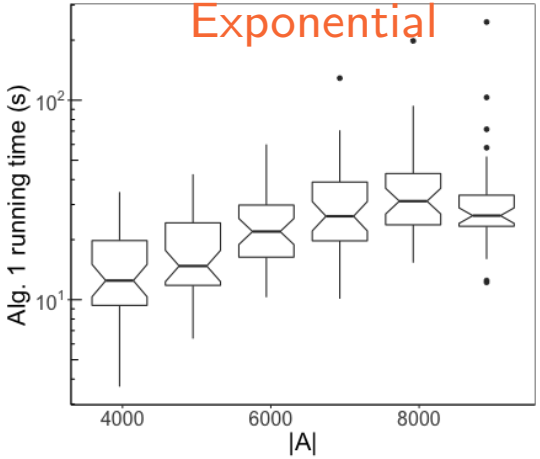
Running Time by $|N|$



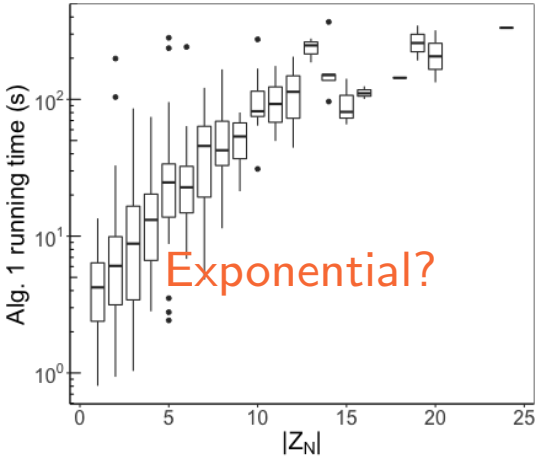
Running Time by p



Running Time by $|A|/(|N|^2)$



Running Time by $|A|$



Running Time by $|Z_N|$

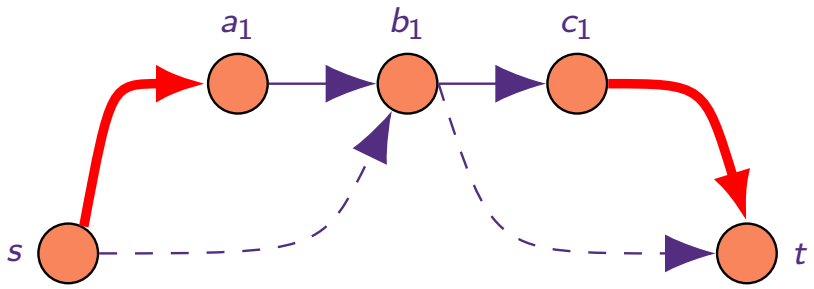
Contents

1. Introduction
2. Problem Description
3. Algorithm 1: Reformulation
4. Algorithm 2: Recursion
5. Algorithm 3: Best of Both
6. Conclusions

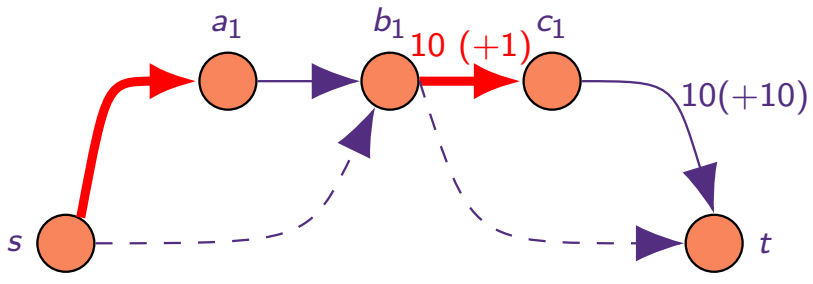
Key Lemmas

Lemma (Series Configurations)

If $\mathbf{x} \in X_E(G^a \otimes G^b, \kappa^L)$, *then* $\mathbf{x} = \mathbf{x}^a + \mathbf{x}^b$ *where* $\mathbf{x}^a \in X_E(G^a, \kappa^L_a)$ *and* $\mathbf{x}^b \in X_E(G^b, \kappa^L - \kappa^L_a)$ *with* $\kappa^L_a = \sum_{(i,j)\in A} x^a_{ij}$.



(a) Efficient



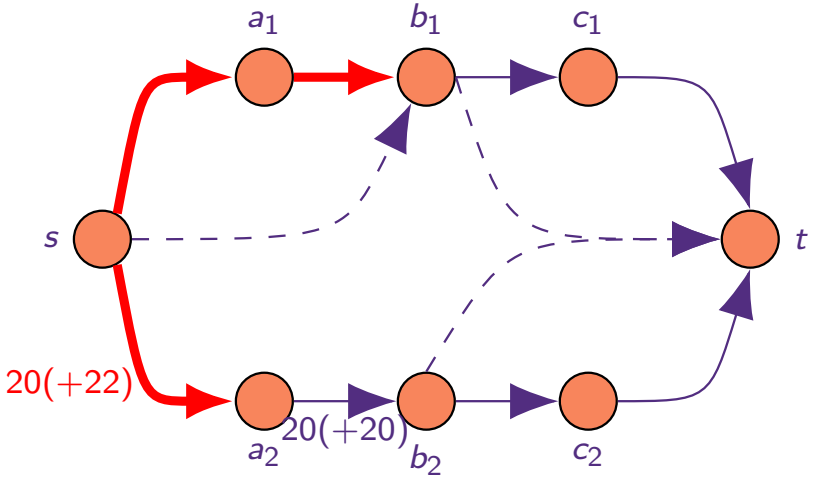
(b) Not Efficient

Key Lemmas

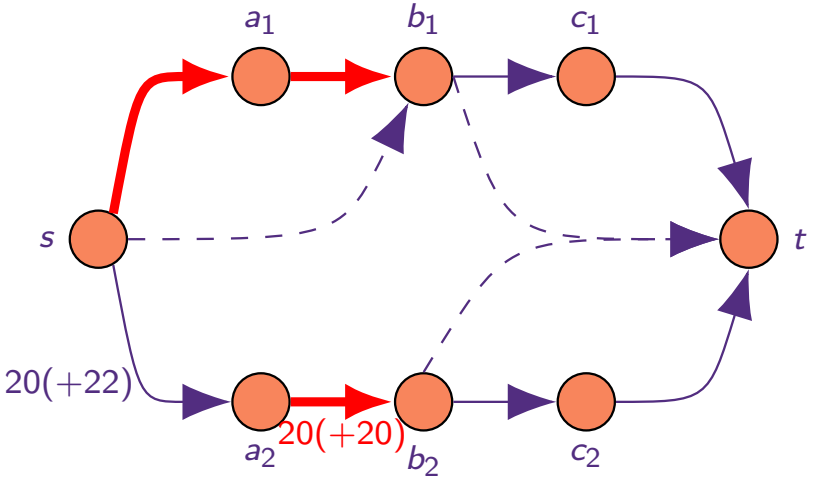
Lemma (Series Configurations)

Lemma (Parallel Configurations)

Let $\mathbf{x} \in X_E(G^a \oplus G^b, \kappa^L)$. *Then there exist* $\bar{\mathbf{x}}^a \in X_E(G^a, \kappa^L_a)$ *and* $\bar{\mathbf{x}}^b \in X_E(G^b, \kappa^L - \kappa^L_a)$, *such that* $\mathbf{z}^{G^a \oplus G^b}(\mathbf{x}) = \mathbf{z}^{G^a \oplus G^b}(\bar{\mathbf{x}}^a + \bar{\mathbf{x}}^b)$.



(a) Efficient



(b) Efficient

Key Lemmas

Lemma (Series Configurations)

If $\mathbf{x} \in X_E(G^a \otimes G^b, \kappa^L)$*, then* $\mathbf{x} = \mathbf{x}^a + \mathbf{x}^b$ *where* $\mathbf{x}^a \in X_E(G^a, \kappa_a^L)$ *and* $\mathbf{x}^b \in X_E(G^b, \kappa^L - \kappa_a^L)$ *with* $\kappa_a^L = \sum_{(i,j)\in A} x_{ij}^a$.

Lemma (Parallel Configurations)

Let $\mathbf{x} \in X_E(G^a \oplus G^b, \kappa^L)$*. Then there exist* $\bar{\mathbf{x}}^a \in X_E(G^a, \kappa_a^L)$ *and* $\bar{\mathbf{x}}^b \in X_E(G^b, \kappa^L - \kappa_a^L)$*, such that* $\mathbf{z}^{G^a \oplus G^b}(\mathbf{x}) = \mathbf{z}^{G^a \oplus G^b}(\bar{\mathbf{x}}^a + \bar{\mathbf{x}}^b)$.

Corollary

If G *is series-parallel, then there exists a minimal complete efficient set composed solely of efficient sub-graph strategies.*

Algorithm 2

Algorithm 1: Recursion

Generate a binary decomposition tree, $\mathcal{T}$
Call $\texttt{SP}(n)$ on the root, r, of $\mathcal{T}$ ($G^r = G$)
Return $X_{CE}(G^r, k^L)$

SP(n)
if $\texttt{Type}(n) =$ "L" **then** // Base Case
  return $X_{CE}(G^n, \kappa^L) = [0, \ldots, 0, \kappa^L, 0, \ldots, 0]$ for $\kappa^L = 0, \ldots, k^L$
else // Recursion Case
  $(c^a, c^b) \leftarrow \texttt{Children}(n)$
  $X_{CE}(G^a, \kappa^L) \leftarrow \texttt{SP}(c^a)$; $X_{CE}(G^b, \kappa^L) \leftarrow \texttt{SP}(c^b)$
  for $(\mathbf{x}^a, \mathbf{x}^b) \in X_E(G^a, \kappa^a) \times X_E(G^b, \kappa^b)$ **do**
    Add $(\mathbf{x}^a + \mathbf{x}^b)$ to $\widehat{X}_{CE}(G^n, \kappa^a + \kappa^b)$
    if $\texttt{Type}(n) =$ "P" **then**
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\min\left\{z_0^{G^a}(\mathbf{x}^a), z_0^{G^b}(\mathbf{x}^b)\right\}, \ldots, \min\left\{z_p^{G^a}(\mathbf{x}^a), z_p^{G^b}(\mathbf{x}^b)\right\}\right]$
    else // $\texttt{Type}(n) =$ "S"
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\left(z_0^{G^a}(\mathbf{x}^a) + z_0^{G^b}(\mathbf{x}^b)\right), \ldots, \min_{\kappa \in \{0, \ldots, p\}}\left\{z_\kappa^{G^a}(\mathbf{x}^a) + z_{p-\kappa}^{G^b}(\mathbf{x}^b)\right\}\right]$
  for $\kappa^L = 0, \ldots, k^L$ **do**
    for $\mathbf{x}^a, \mathbf{x}^b \in \widehat{X}_{CE}(G^n, \kappa^L)$ **do**
      if $\mathbf{z}^{G^n}(\mathbf{x}^a) \geqq \mathbf{z}^{G^n}(\mathbf{x}^b)$ **then** Remove $\mathbf{x}^b$ from $\widehat{X}_{CE}(G^n, \kappa^L)$
  return $X_{CE}(G^n, \kappa^L)$ for $\kappa^L = 0, \ldots, k^L$

Algorithm 2

Algorithm 2: Recursion

Generate a binary decomposition tree, $\mathcal{T}$
Call SP(n) on the root, r, of $\mathcal{T}$ ($G^r = G$)
Return $X_{CE}(G^r, k^L)$

SP(n)
if Type(n) = "L" **then** // Base Case
  return $X_{CE}(G^n, \kappa^L) = [0, \ldots, 0, \kappa^L, 0, \ldots, 0]$ for $\kappa^L = 0, \ldots, k^L$
else // Recursion Case
  $(c^a, c^b) \leftarrow$ Children(n)
  $X_{CE}(G^a, \kappa^L) \leftarrow$ SP(c^a); $X_{CE}(G^b, \kappa^L) \leftarrow$ SP(c^b)
  for $(\mathbf{x}^a, \mathbf{x}^b) \in X_E(G^a, \kappa^a) \times X_E(G^b, \kappa^b)$ **do**
    Add $(\mathbf{x}^a + \mathbf{x}^b)$ to $\widehat{X}_{CE}(G^n, \kappa^a + \kappa^b)$
    if Type(n) = "P" **then**
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\min\left\{z_0^{G^a}(\mathbf{x}^a), z_0^{G^b}(\mathbf{x}^b)\right\}, \ldots, \min\left\{z_p^{G^a}(\mathbf{x}^a), z_p^{G^b}(\mathbf{x}^b)\right\}\right]$
    else // Type(n) = "S"
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\left(z_0^{G^a}(\mathbf{x}^a) + z_0^{G^b}(\mathbf{x}^b)\right), \ldots, \min_{\kappa \in \{0, \ldots, p\}}\left\{z_\kappa^{G^a}(\mathbf{x}^a) + z_{p-\kappa}^{G^b}(\mathbf{x}^b)\right\}\right]$
  for $\kappa^L = 0, \ldots, k^L$ **do**
    for $\mathbf{x}^a, \mathbf{x}^b \in \widehat{X}_{CE}(G^n, \kappa^L)$ **do**
      if $\mathbf{z}^{G^n}(\mathbf{x}^a) \geqq \mathbf{z}^{G^n}(\mathbf{x}^b)$ **then** Remove $\mathbf{x}^b$ from $\widehat{X}_{CE}(G^n, \kappa^L)$
  return $X_{CE}(G^n, \kappa^L)$ for $\kappa^L = 0, \ldots, k^L$

Core Alg: 3 Steps

Algorithm 2

Algorithm 2: Recursion

Generate a binary decomposition tree, $\mathcal{T}$
Call SP(n) on the root, r, of $\mathcal{T}$ ($G^r = G$)
Return $X_{CE}(G^r, k^L)$

SP(n)

Base Case: Interdict one arc

if Type(n) = "L" **then** // Base Case
  return $X_{CE}(G^n, \kappa^L) = [0, \ldots, 0, \kappa^L, 0, \ldots, 0]$ for $\kappa^L = 0, \ldots, k^L$
else // Recursion Case
  $(c^a, c^b) \leftarrow$ Children(n)
  $X_{CE}(G^a, \kappa^L) \leftarrow$ SP(c^a); $X_{CE}(G^b, \kappa^L) \leftarrow$ SP(c^b)
  for $(\mathbf{x}^a, \mathbf{x}^b) \in X_E(G^a, \kappa^a) \times X_E(G^b, \kappa^b)$ **do**
    Add $(\mathbf{x}^a + \mathbf{x}^b)$ to $\widehat{X}_{CE}(G^n, \kappa^a + \kappa^b)$
    if Type(n) = "P" **then**
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\min\left\{z_0^{G^a}(\mathbf{x}^a), z_0^{G^b}(\mathbf{x}^b)\right\}, \ldots, \min\left\{z_p^{G^a}(\mathbf{x}^a), z_p^{G^b}(\mathbf{x}^b)\right\}\right]$
    else // Type(n) = "S"
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\left(z_0^{G^a}(\mathbf{x}^a) + z_0^{G^b}(\mathbf{x}^b)\right), \ldots, \min_{\kappa \in \{0,\ldots,p\}}\left\{z_\kappa^{G^a}(\mathbf{x}^a) + z_{p-\kappa}^{G^b}(\mathbf{x}^b)\right\}\right]$
  for $\kappa^L = 0, \ldots, k^L$ **do**
    for $\mathbf{x}^a, \mathbf{x}^b \in \widehat{X}_{CE}(G^n, \kappa^L)$ **do**
      if $\mathbf{z}^{G^n}(\mathbf{x}^a) \geqq \mathbf{z}^{G^n}(\mathbf{x}^b)$ **then** Remove $\mathbf{x}^b$ from $\widehat{X}_{CE}(G^n, \kappa^L)$
  return $X_{CE}(G^n, \kappa^L)$ for $\kappa^L = 0, \ldots, k^L$

Algorithm 2

Algorithm 2: Recursion

Generate a binary decomposition tree, $\mathcal{T}$
Call SP(n) on the root, r, of $\mathcal{T}$ ($G^r = G$)
Return $X_{CE}(G^r, k^L)$

SP(n)
if Type(n) = "L" **then** // Base Case
  return $X_{CE}(G^n, \kappa^L) = [0, \ldots, 0, \kappa^L, 0, \ldots, 0]$ for $\kappa^L = 0, \ldots, k^L$
else // Recursion Case
  $(c^a, c^b) \leftarrow$ Children(n)
  $X_{CE}(G^a, \kappa^L) \leftarrow$ SP(c^a); $X_{CE}(G^b, \kappa^L) \leftarrow$ SP(c^b)
  for $(\mathbf{x}^a, \mathbf{x}^b) \in X_E(G^a, \kappa^a) \times X_E(G^b, \kappa^b)$ **do**
    Add $(\mathbf{x}^a + \mathbf{x}^b)$ to $\widehat{X}_{CE}(G^n, \kappa^a + \kappa^b)$
    if Type(n) = "P" **then**
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\min\left\{z_0^{G^a}(\mathbf{x}^a), z_0^{G^b}(\mathbf{x}^b)\right\}, \ldots, \min\left\{z_p^{G^a}(\mathbf{x}^a), z_p^{G^b}(\mathbf{x}^b)\right\}\right]$
    else // Type(n) = "S"
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\left(z_0^{G^a}(\mathbf{x}^a) + z_0^{G^b}(\mathbf{x}^b)\right), \ldots, \min_{\kappa \in \{0,\ldots,p\}}\left\{z_\kappa^{G^a}(\mathbf{x}^a) + z_{p-\kappa}^{G^b}(\mathbf{x}^b)\right\}\right]$
  for $\kappa^L = 0, \ldots, k^L$ **do**
    for $\mathbf{x}^a, \mathbf{x}^b \in \widehat{X}_{CE}(G^n, \kappa^L)$ **do**
      if $\mathbf{z}^{G^n}(\mathbf{x}^a) \geqq \mathbf{z}^{G^n}(\mathbf{x}^b)$ **then** Remove $\mathbf{x}^b$ from $\widehat{X}_{CE}(G^n, \kappa^L)$
  return $X_{CE}(G^n, \kappa^L)$ for $\kappa^L = 0, \ldots, k^L$

Recourse through children

Algorithm 2

Algorithm 2: Recursion

Generate a binary decomposition tree, $\mathcal{T}$
Call $\mathtt{SP}(n)$ on the root, r, of $\mathcal{T}$ ($G^r = G$)
Return $X_{CE}(G^r, k^L)$

SP(n)
if $\mathtt{Type}(n) =$ "L" **then** // Base Case
  return $X_{CE}(G^n, \kappa^L) = [0, \ldots, 0, \kappa^L, 0, \ldots, 0]$ for $\kappa^L = 0, \ldots, k^L$
else // Recursion Case
  $(c^a, c^b) \leftarrow \mathtt{Children}(n)$
  $X_{CE}(G^a, \kappa^L) \leftarrow \mathtt{SP}(c^a)$; $X_{CE}(G^b, \kappa^L) \leftarrow \mathtt{SP}(c^b)$
  for $(\mathbf{x}^a, \mathbf{x}^b) \in X_E(G^a, \kappa^a) \times X_E(G^b, \kappa^b)$ **do**
    Add $(\mathbf{x}^a + \mathbf{x}^b)$ to $\widehat{X}_{CE}(G^n, \kappa^a + \kappa^b)$
    if $\mathtt{Type}(n) =$ "P" **then**
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\min\left\{z_0^{G^a}(\mathbf{x}^a), z_0^{G^b}(\mathbf{x}^b)\right\}, \ldots, \min\left\{z_p^{G^a}(\mathbf{x}^a), z_p^{G^b}(\mathbf{x}^b)\right\}\right]$
    else // $\mathtt{Type}(n) =$ "S"
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\left(z_0^{G^a}(\mathbf{x}^a) + z_0^{G^b}(\mathbf{x}^b)\right), \ldots, \min_{\kappa \in \{0,\ldots,p\}}\left\{z_\kappa^{G^a}(\mathbf{x}^a) + z_{p-\kappa}^{G^b}(\mathbf{x}^b)\right\}\right]$
  for $\kappa^L = 0, \ldots, k^L$ **do**
    for $\mathbf{x}^a, \mathbf{x}^b \in \widehat{X}_{CE}(G^n, \kappa^L)$ **do**
      if $\mathbf{z}^{G^n}(\mathbf{x}^a) \geq \mathbf{z}^{G^n}(\mathbf{x}^b)$ **then** Remove $\mathbf{x}^b$ from $\widehat{X}_{CE}(G^n, \kappa^L)$
  return $X_{CE}(G^n, \kappa^L)$ for $\kappa^L = 0, \ldots, k^L$

Combine children solutions

Algorithm 2

Algorithm 2: Recursion

Generate a binary decomposition tree, $\mathcal{T}$
Call $\mathtt{SP}(n)$ on the root, r, of $\mathcal{T}$ ($G^r = G$)
Return $X_{CE}(G^r, k^L)$

SP(n)
if $\mathtt{Type}(n) =$ "L" **then** // Base Case
  return $X_{CE}(G^n, \kappa^L) = [0, \ldots, 0, \kappa^L, 0, \ldots, 0]$ for $\kappa^L = 0, \ldots, k^L$
else // Recursion Case
  $(c^a, c^b) \leftarrow \mathtt{Children}(n)$
  $X_{CE}(G^a, \kappa^L) \leftarrow \mathtt{SP}(c^a)$; $X_{CE}(G^b, \kappa^L) \leftarrow \mathtt{SP}(c^b)$
  for $(\mathbf{x}^a, \mathbf{x}^b) \in X_E(G^a, \kappa^a) \times X_E(G^b, \kappa^b)$ **do**
    Add $(\mathbf{x}^a + \mathbf{x}^b)$ to $\widehat{X}_{CE}(G^n, \kappa^a + \kappa^b)$
    if $\mathtt{Type}(n) =$ "P" **then**
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\min\left\{z_0^{G^a}(\mathbf{x}^a), z_0^{G^b}(\mathbf{x}^b)\right\}, \ldots, \min\left\{z_p^{G^a}(\mathbf{x}^a), z_p^{G^b}(\mathbf{x}^b)\right\}\right]$
    else // $\mathtt{Type}(n) =$ "S"
      $\mathbf{z}^{G^n}(\mathbf{x}^a + \mathbf{x}^b) \leftarrow \left[\left(z_0^{G^a}(\mathbf{x}^a) + z_0^{G^b}(\mathbf{x}^b)\right), \ldots, \min_{\kappa \in \{0, \ldots, p\}}\left\{z_\kappa^{G^a}(\mathbf{x}^a) + z_{p-\kappa}^{G^b}(\mathbf{x}^b)\right\}\right]$

  for $\kappa^L = 0, \ldots, k^L$ **do**
    for $\mathbf{x}^a, \mathbf{x}^b \in \widehat{X}_{CE}(G^n, \kappa^L)$ **do**
      if $\mathbf{z}^{G^n}(\mathbf{x}^a) \geqq \mathbf{z}^{G^n}(\mathbf{x}^b)$ **then** Remove $\mathbf{x}^b$ from $\widehat{X}_{CE}(G^n, \kappa^L)$

  return $X_{CE}(G^n, \kappa^L)$ for $\kappa^L = 0, \ldots, k^L$

Throw out dominated sol'ns

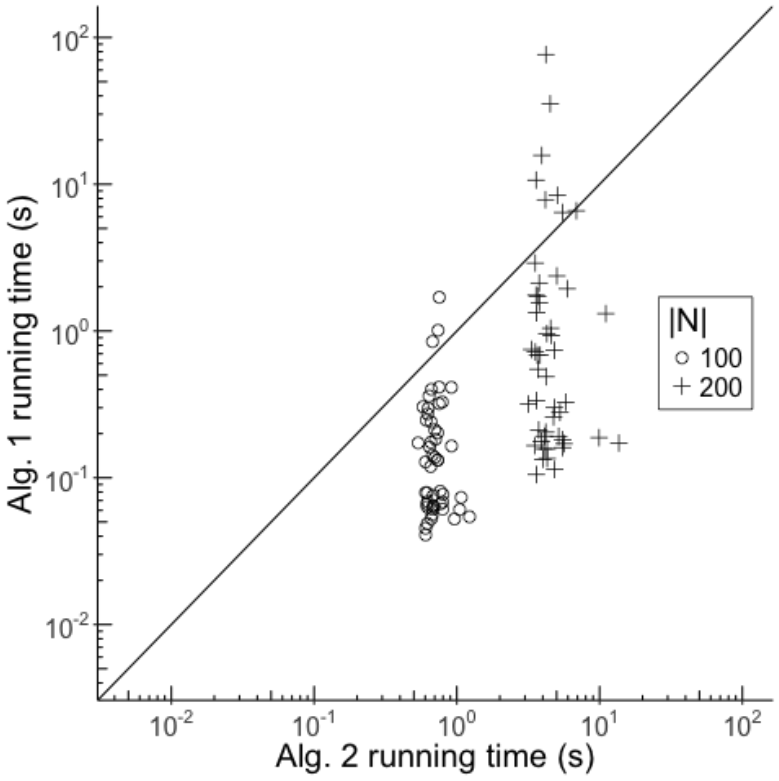
Comparison

- Which one is faster on Series-Parallel Graphs?
  - Reformulation?
  - Recursion?

Comparison

- Which one is faster on Series-Parallel Graphs?
  - Reformulation?
  - Recursion?

It depends.



Comparison

- Which one is faster on Series-Parallel Graphs?
  - Reformulation
  - Recursion

	Reformulation Faster	**Recursion Faster**
$\|N\|$	100 nodes	100 nodes
$\|Z_N\|$	1 solution	5 solutions
Time (Alg 1/Alg 2)	0.070/0.218 seconds	0.662/0.317 seconds

Comparison

- Which one is faster on Series-Parallel Graphs?
  - Reformulation: Avoid "rabbit trails" via row generation
  - Recursion

	Reformulation Faster	**Recursion Faster**
$\lvert N \rvert$	100 nodes	100 nodes
$\lvert Z_N \rvert$	1 solution	5 solutions
Time (Alg 1/Alg 2)	0.070/0.218 seconds	0.662/0.317 seconds

Comparison

- Which one is faster on Series-Parallel Graphs?
  - Reformulation: Avoid "rabbit trails" via row generation
  - Recursion: Computation time bounded by $|A|$, and not $|Z_N|$

	Reformulation Faster	**Recursion Faster**
$\|N\|$	100 nodes	100 nodes
$\|Z_N\|$	1 solution	5 solutions
Time (Alg 1/Alg 2)	0.070/0.218 seconds	0.662/0.317 seconds

Contents

1. Introduction
2. Problem Description
3. Algorithm 1: Reformulation
4. Algorithm 2: Recursion
5. Algorithm 3: Best of Both
6. Conclusions

Algorithm 3: Best of Both

$(\mathcal{T}, r) \leftarrow \texttt{DecompTree}(G)$; $\bar{G} \leftarrow \texttt{Layerize}(G, p)$
for $n \in \mathcal{T}$ **do**
  $\widehat{X}_{CE}(G^n, \kappa^L) \leftarrow \begin{cases} \{\mathbf{0}\} & \kappa^L = 0 \\ \emptyset & \kappa^L = 1, \ldots, k^L \end{cases}$
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{False}$
$\texttt{NeedsUpdate}(r) \leftarrow \texttt{True}$
while $\texttt{NeedsUpdate}(r)$ **do**
  $\widehat{X}_{CE}(G, k^L) \leftarrow \texttt{SP}^*(r)$
  for $\mathbf{x} \in \widehat{X}_{CE}(G, k^L)$ **do**
    Update $\bar{\mathbf{c}}$ over $\bar{A}$ with interdictions in $\mathbf{x}$
    $\mathbf{y}_h \leftarrow \texttt{Dijkstra}(\bar{G}, s_0, t_h, \bar{\mathbf{c}})$ for $h = 0, \ldots, p$
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{True}$ for all leaves corresponding to new arcs in $\mathbf{y}$ and their ancestors in $\mathcal{T}$
return $\widehat{X}_{CE}(G, k^L)$

$\texttt{SP}^*(n)$
if $\texttt{NeedsUpdate}(n)$ **then**
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{False}$
  return $\texttt{SP}(n)$
else
  return $\widehat{X}_{CE}(G^n, \kappa^L)$

Algorithm 3: Best of Both

```
(𝒯, r) ← DecompTree(G); Ḡ ← Layerize(G, p)
for n ∈ 𝒯 do
    X̂_CE(G^n, κ^L) ← { {0}  κ^L = 0
                       ∅    κ^L = 1, ..., k^L
    NeedsUpdate(n) ← False
NeedsUpdate(r) ← True
while NeedsUpdate(r) do
    X̂_CE(G, k^L) ← SP*(r)
    for x ∈ X̂_CE(G, k^L) do
        Update c̄ over Ā with interdictions in x
        y_h ← Dijkstra(Ḡ, s_0, t_h, c̄) for h = 0, ..., p
    NeedsUpdate(n) ← True for all leaves corresponding to new arcs in y and their ancestors in 𝒯
return X̂_CE(G, k^L)

SP*(n)
if NeedsUpdate(n) then
    NeedsUpdate(n) ← False
    return SP(n)
else
    return X̂_CE(G^n, κ^L)
```

Use both structures.
Only parse recursion tree for known leaves.

Algorithm 3: Best of Both

$(\mathcal{T}, r) \leftarrow \texttt{DecompTree}(G)$; $\bar{G} \leftarrow \texttt{Layerize}(G, p)$
for $n \in \mathcal{T}$ **do**
  $\widehat{X}_{CE}(G^n, \kappa^L) \leftarrow \begin{cases} \{\mathbf{0}\} & \kappa^L = 0 \\ \emptyset & \kappa^L = 1, \ldots, k^L \end{cases}$
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{False}$
$\texttt{NeedsUpdate}(r) \leftarrow \texttt{True}$
while $\texttt{NeedsUpdate}(r)$ **do**
  $\widehat{X}_{CE}(G, k^L) \leftarrow \texttt{SP}^*(r)$
  for $\mathbf{x} \in \widehat{X}_{CE}(G, k^L)$ **do**
    Update $\bar{\mathbf{c}}$ over $\bar{A}$ with interdictions in $\mathbf{x}$
    $\mathbf{y}_h \leftarrow \texttt{Dijkstra}(\bar{G}, s_0, t_h, \bar{\mathbf{c}})$ for $h = 0, \ldots, p$
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{True}$ for all leaves corresponding to new arcs in $\mathbf{y}$ and their ancestors in $\mathcal{T}$
return $\widehat{X}_{CE}(G, k^L)$

$\texttt{SP}^*(n)$
if $\texttt{NeedsUpdate}(n)$ **then**
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{False}$
  return $\texttt{SP}(n)$
else
  return $\widehat{X}_{CE}(G^n, \kappa^L)$

Set incumbent strategy

Algorithm 3: Best of Both

$(\mathcal{T}, r) \leftarrow \texttt{DecompTree}(G)$; $\bar{G} \leftarrow \texttt{Layerize}(G, p)$
for $n \in \mathcal{T}$ **do**
  $\widehat{X}_{CE}(G^n, \kappa^L) \leftarrow \begin{cases} \{\mathbf{0}\} & \kappa^L = 0 \\ \emptyset & \kappa^L = 1, \ldots, k^L \end{cases}$
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{False}$
$\texttt{NeedsUpdate}(r) \leftarrow \texttt{True}$
while $\texttt{NeedsUpdate}(r)$ **do**
  $\widehat{X}_{CE}(G, k^L) \leftarrow \texttt{SP}^*(r)$
  for $\mathbf{x} \in \widehat{X}_{CE}(G, k^L)$ **do**
    Update $\bar{\mathbf{c}}$ over $\bar{A}$ with interdictions in $\mathbf{x}$
    $\mathbf{y}_h \leftarrow \texttt{Dijkstra}(\bar{G}, s_0, t_h, \bar{\mathbf{c}})$ for $h = 0, \ldots, p$
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{True}$ for all leaves corresponding to new arcs in $\mathbf{y}$ and their ancestors in $\mathcal{T}$
return $\widehat{X}_{CE}(G, k^L)$

Get a new strategy

$\texttt{SP}^*(n)$
if $\texttt{NeedsUpdate}(n)$ **then**
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{False}$
  return $\texttt{SP}(n)$
else
  return $\widehat{X}_{CE}(G^n, \kappa^L)$

If node needs updating use SP from Alg 2; otherwise take incumbent

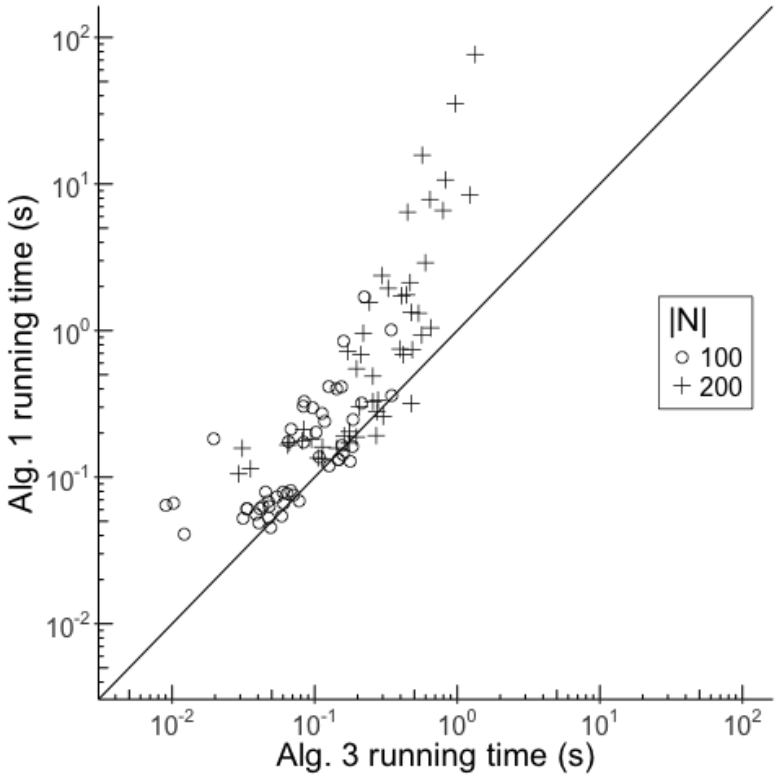
Algorithm 3: Best of Both

$(\mathcal{T}, r) \leftarrow \texttt{DecompTree}(G)$; $\bar{G} \leftarrow \texttt{Layerize}(G, p)$
for $n \in \mathcal{T}$ **do**
  $\widehat{X}_{CE}(G^n, \kappa^L) \leftarrow \begin{cases} \{\mathbf{0}\} & \kappa^L = 0 \\ \emptyset & \kappa^L = 1, \ldots, k^L \end{cases}$
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{False}$
$\texttt{NeedsUpdate}(r) \leftarrow \texttt{True}$
while $\texttt{NeedsUpdate}(r)$ **do**
  $\widehat{X}_{CE}(G, k^L) \leftarrow \texttt{SP}^*(r)$
  for $\mathbf{x} \in \widehat{X}_{CE}(G, k^L)$ **do**
    Update $\bar{\mathbf{c}}$ over $\bar{A}$ with interdictions in $\mathbf{x}$
    $\mathbf{y}_h \leftarrow \texttt{Dijkstra}(\bar{G}, s_0, t_h, \bar{\mathbf{c}})$ for $h = 0, \ldots, p$
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{True}$ for all leaves corresponding to new arcs in $\mathbf{y}$ and their ancestors in $\mathcal{T}$
return $\widehat{X}_{CE}(G, k^L)$

$\texttt{SP}^*(n)$
if $\texttt{NeedsUpdate}(n)$ **then**
  $\texttt{NeedsUpdate}(n) \leftarrow \texttt{False}$
  return $\texttt{SP}(n)$
else
  return $\widehat{X}_{CE}(G^n, \kappa^L)$

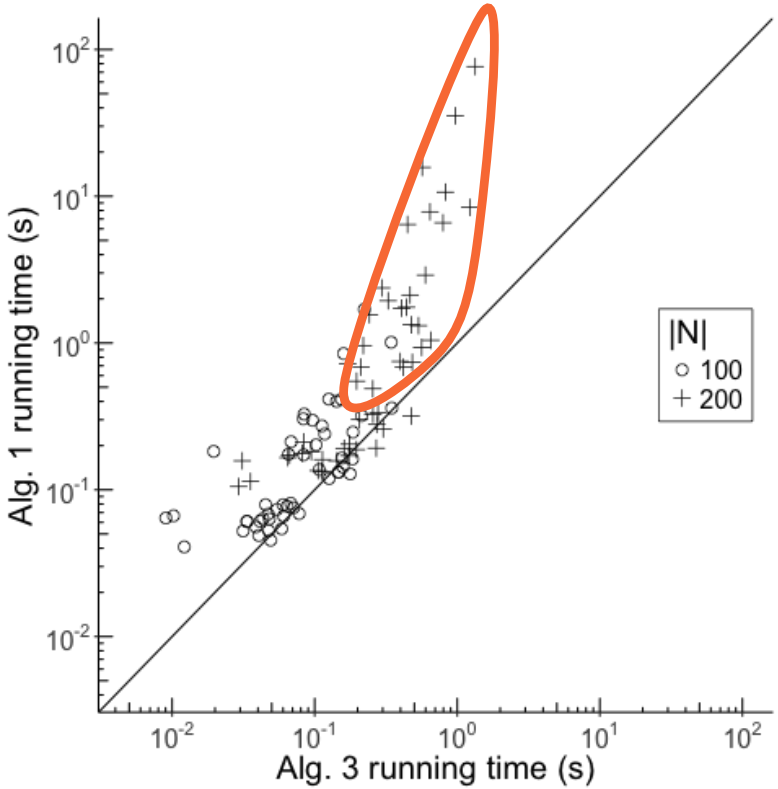
Find shortest paths.
Update leaves & ancestors.

Alg 3 Running Times



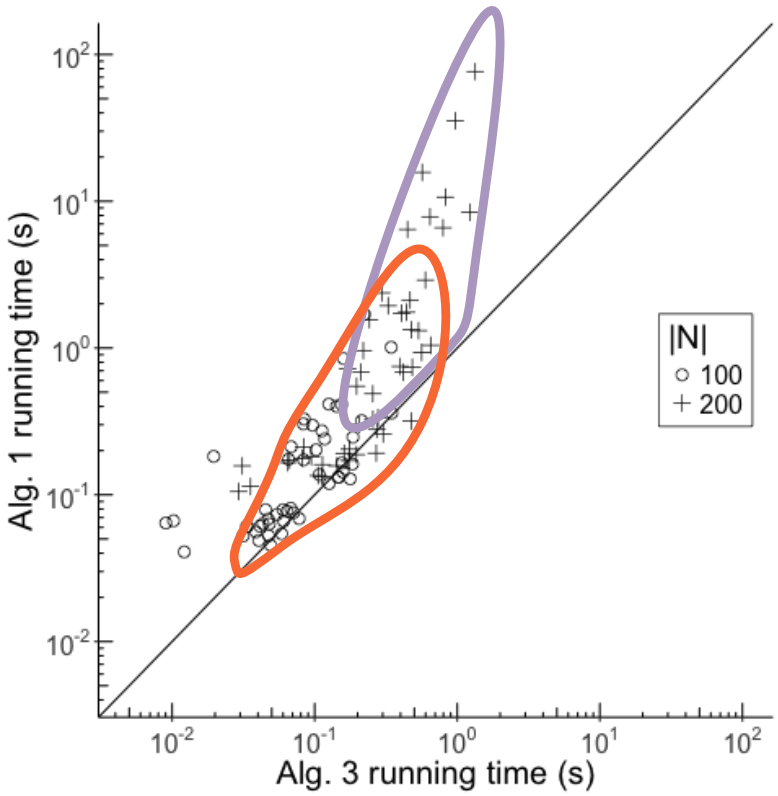
- Strengths of Both

Alg 3 Running Times



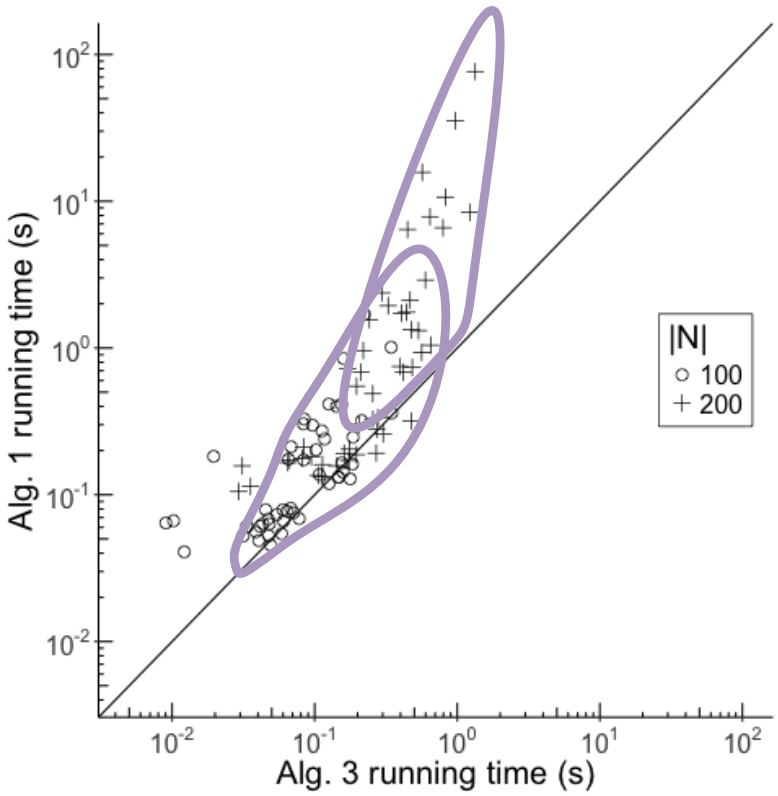
- Strengths of Both
  - $|Z_N|$ large $\Rightarrow$ Time $\sim$ Alg 2 (bounded by $|A|$)

Alg 3 Running Times



- Strengths of Both
  - $|Z_N|$ large $\Rightarrow$ Time $\sim$ Alg 2 (bounded by $|A|$)
  - $|Z_N|$ small $\Rightarrow$ Time $\sim$ Alg 1 (ignores "rabbit trails")

Alg 3 Running Times



- Strengths of Both
  - $|Z_N|$ large $\Rightarrow$ Time $\sim$ Alg 2 (bounded by $|A|$)
  - $|Z_N|$ small $\Rightarrow$ Time $\sim$ Alg 1 (ignores "rabbit trails")

Contents

1. Introduction
2. Problem Description
3. Algorithm 1: Reformulation
4. Algorithm 2: Recursion
5. Algorithm 3: Best of Both
6. Conclusions

Conclusion

- **Recap**
  - SPIP-I: SPIP with uncertain follower recourse
  - Algorithm 1: Reformulation
  - Algorithm 2: Recursion
  - Algorithm 3: Best of Both
- **Future Work**
  - Algorithm 3 on other graph structures
  - Fortification-by-obfuscation

References I

Juan S. Borrero, Oleg A. Prokopyev, and Denis Sauré. Sequential shortest path interdiction with incomplete information. *Decision Analysis*, 13(1):68–98, 2015.

Eugene P. Durbin. *An Interdiction Model of Highway Transportation*. Rand Corporation, 1966.

Timothy Holzmann and J. C. Smith. Solving discrete multi-objective optimization problems using modified augmented weighted Tchebychev scalarizations. *European Journal of Operational Research*, 271(2):436–449, 2018.

E. Israeli and R. K. Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002.

Gokhan Kirlik and Serpil Sayin. A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, 232(3):479–488, 2014.

Renata A. Konrad, Andrew C. Trapp, Timothy M. Palmbach, and Jeffrey S. Blom. Overcoming human trafficking via operations research and analytics: Opportunities for methods, models, and applications. *European Journal of Operational Research*, 259(1):733–745, 2017.

Leonardo Lozano and J Cole Smith. A backward sampling framework for interdiction problems with fortification. *INFORMS Journal on Computing*, 29(1):123–139, 2017. URL `http://dx.doi.org/10.1287/ijoc.2016.0721`.

Brian J. Lunday and Hanif D. Sherali. A dynamic network interdiction problem. *INFORMATICA*, 21(4):553–574, 2010.

George Mavrotas and Danae Diakoulaki. Multi-criteria branch and bound: A vector maximization algorithm for mixed 0-1 multiple objective linear programming. *Applied Mathematics and Computation*, 171(1):53–71, 2005.

D. P. Morton, F. Pan, and K. J. Saeger. Models for nuclear smuggling interdiction. *IIE Transactions*, 39(1):3–14, 2007.

Melih Özlen, Benjamin A. Burton, and Cameron A. G. MacRae. Multi-objective integer programming: An improved recursive algorithm. *Journal of Optimization Theory and Applications*, 160(2):470–482, 2014.

A. Przybylski, X. Gandibleux, and Matthias Ehrgott. A two-phase method for multi-objective integer programming and its application to the assignment problem with three objectives. *Discrete Optimization*, 7(3):149–165, 2010.

References II

Javier Salmerón. Deception tactics for network interdiction: A multiobjective approach. *Networks*, 60(1):45–58, 2012. doi: 10.1002/net.20458. URL http://dx.doi.org/10.1002/net.20458.

J. C. Smith. Basic interdiction models. In J. Cochran, editor, *Wiley Encyclopedia of Operations Research and Management Science*, pages 323–330. Wiley, Hoboken, NJ, 2010.

Thomas Stidsen, Kim Allan Andersen, and Bernd Damman. A branch and bound algorithm for a class of biobjective mixed integer programs. *Management Science*, 60(4):1009–1032, 2014.

K. M. Sullivan, D. P. Morton, F. Pan, and J. C. Smith. Securing a border under asymmetric information. *Naval Research Logistics*, 61(2):91–100, 2014.

E.L. Ulungu and J. Teghem. Multi-objective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*, 3(2):83–104, 1994.

Thomas Vincent, Seipp Florian, S. Ruzika, and A. Przybylski. Multiple objective branch and bound for mixed 0-1 linear programming: Corrections and improvements for the biobjective case. *Computers and Operations Research*, 40(1):498–509, 2013.

R. D. Wollmer. Removing arcs from a network. *Operations Research*, 12(6):934–940, 1964.