Duality-based reformulations in bilevel optimization

Part II

Margarida Carvalho

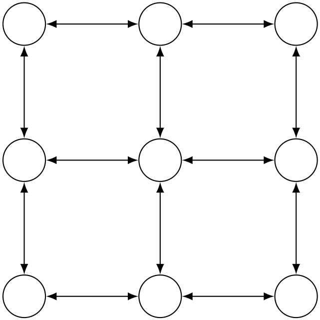
Département d'informatique et de recherche opérationnelle, Université de Montréal

IPCO 2026 Summer School, Padova, Italy
June 15-16, 2026

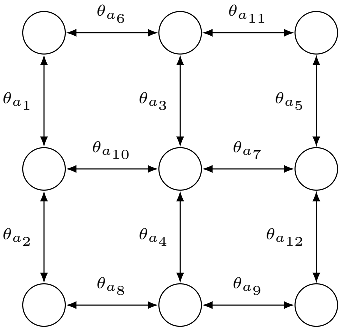
Contents

Contents

- Directed graph (V, A)

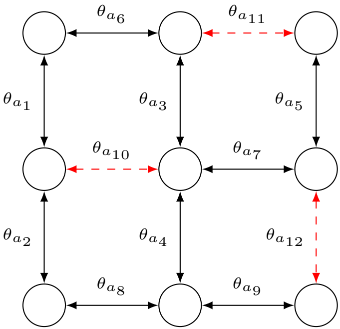


Assume there is always an (o, d)-path over $A \setminus A_1$.



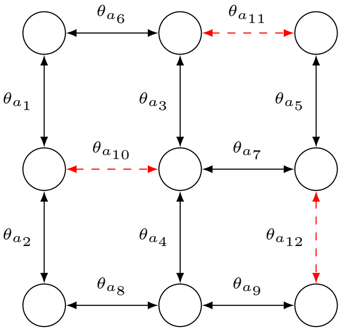
- Directed graph (V, A)
- Travel time $\theta_a(s_a)$ for $a \in A$

Assume there is always an (o, d)-path over $A \setminus A_1$.



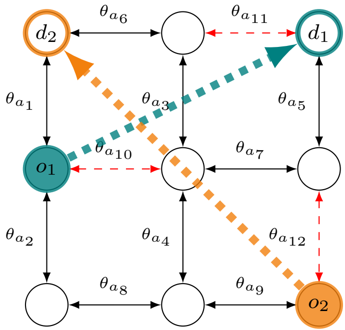
- Directed graph (V, A)
- Travel time $\theta_a(s_a)$ for $a \in A$
- Candidate arcs $A_1 \subseteq A$

Assume there is always an (o, d)-path over $A \setminus A_1$.



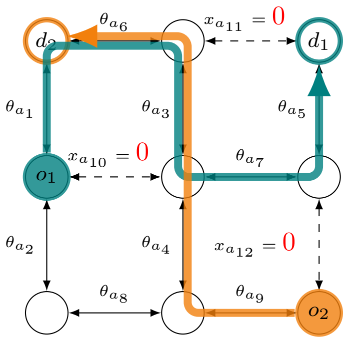
- Directed graph (V, A)
- Travel time $\theta_a(s_a)$ for $a \in A$
- Candidate arcs $A_1 \subseteq A$
- Decisions $x_a = 1$ for $a \in A \setminus A_1$ and $x_a \in \{0,1\}^{A_1}, x \in X$

Assume there is always an (o, d)-path over $A \setminus A_1$.



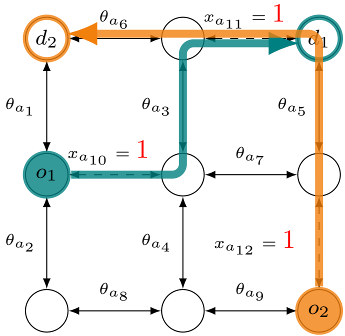
- Directed graph (V, A)
- Travel time $\theta_a(s_a)$ for $a \in A$
- Candidate arcs $A_1 \subseteq A$
- Decisions $x_a = 1$ for $a \in A \setminus A_1$ and $x_a \in \{0,1\}^{A_1}, x \in X$
- Users $k \in K$
  - Origin o_k in V
  - Destination d_k in V
  - Demand e_k

Assume there is always an (o, d)-path over $A \setminus A_1$.



- Directed graph (V, A)
- Travel time $\theta_a(s_a)$ for $a \in A$
- Candidate arcs $A_1 \subseteq A$
- Decisions $x_a = 1$ for $a \in A \setminus A_1$ and $x_a \in \{0,1\}^{A_1}, x \in X$
- Users $k \in K$
  - Origin o_k in V
  - Destination d_k in V
  - Demand e_k
- Flow $s_a = \sum_{k \in K} z_a^k$ for $a \in A$

Assume there is always an (o, d)-path over $A \setminus A_1$.



- Directed graph (V, A)
- Travel time $\theta_a(s_a)$ for $a \in A$
- Candidate arcs $A_1 \subseteq A$
- Decisions $x_a = 1$ for $a \in A \setminus A_1$ and $x_a \in \{0,1\}^{A_1}, x \in X$
- Users $k \in K$
  - Origin o_k in V
  - Destination d_k in V
  - Demand e_k
- Flow $s_a = \sum_{k \in K} z_a^k$ for $a \in A$

Assume there is always an (o, d)-path over $A \setminus A_1$.

Bilevel network design problem

$$\min_{x,z} \sum_{a \in A} \left(\sum_{o \in O} z_{oa} \right) \theta_a \left(\sum_{o \in O} z_{oa} \right) \quad \text{(System Optimum Objective)}$$

$$\text{s.t. } x \in X$$

$$z \in Z(x) \quad \text{(User Equilibrium)}$$

Bilevel network design problem

$$\min_{x,z} \sum_{a\in A} \left(\sum_{o\in O} z_{oa}\right) \theta_a \left(\sum_{o\in O} z_{oa}\right) \quad \text{(System Optimum Objective)}$$

$$\text{s.t. } x \in X$$

$$z \in Z(x) \quad \text{(User Equilibrium)}$$

Travel time function

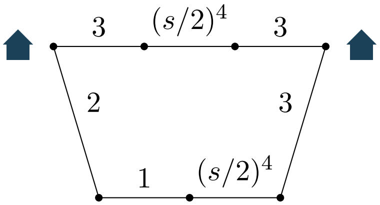
$$\theta_a(s) = f_a + g_a s^p$$

where $f_a \geq 0$ and $g_a \geq 0$ and $p \in \mathbb{Z}_{>0}$.

Contents

Followers

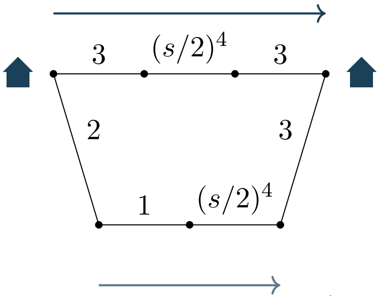
Total flow of 4.



Followers

Total flow of 4. User equilibrium: No incentive to deviate.

flow $= 2$, time $= 3 + (2/2)^4 + 3 = 7$



flow $= 2$, time $= 2 + 1 + (2/2)^4 + 3 = 7$

Followers [Beckmann et al., 1955]

Primal Formulation [Beckmann et al., 1955]

$$\phi(x) = \min_{z} \sum_{a \in A(x)} \int_0^{\sum_{o \in O} z_{oa}} \theta_a(s) ds \tag{1a}$$

$$\text{s.t.} \sum_{a \in A_v^+(x)} z_{oa} - \sum_{a \in A_v^-(x)} z_{oa} = e_{ov}, \quad \forall o \in O, v \in V, \tag{1b}$$

$$z \geq 0. \tag{1c}$$

- $O = \{v \in V : \exists k \in K, o_k = v\}$
- $e_{ov} = \sum_{k \in K: o \in k} e'_{kv}$ where $e'_{ko_k} = -e_k$, $e'_{kd_k} = e_k$, $e'_{kv} = 0$ for $v \notin \{o_k, d_k\}$.
- z_{oa}: Flow of users originating from o traversing arc a.

Followers

Consider for arc a:
Travel time function: $\theta_a(s) = 1 + s^2$
Demand: $o_1 \to 1$ unit, $o_2 \to 1$ unit.
Total flow: $s = 2$.

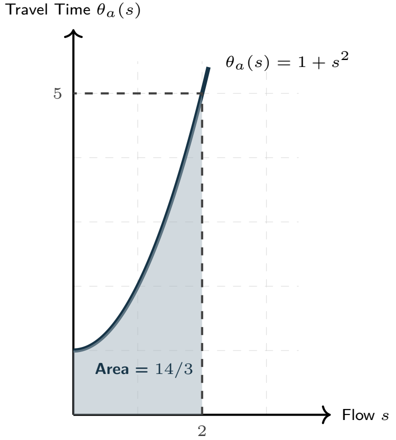
The arc travel time at total flow is:

$$\theta_a(2) = 1 + 4 = 5$$

Contribution in the lower-level objective:

$$\Phi_a(2) = \int_0^2 (1 + s^2)\, ds = \left[s + \frac{s^3}{3} \right]_0^2 = \frac{14}{3}$$

Integrating from 0 to s accumulates the travel time of incremental drivers.



Followers

Consider for arc a:
Travel time function: $\theta_a(s) = 1 + s^2$
Demand: $o_1 \to 1$ unit, $o_2 \to 1$ unit.
Total flow: $s = 2$.

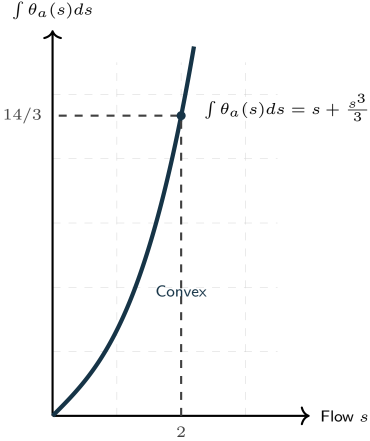
The arc travel time at total flow is:

$$\theta_a(2) = 1 + 4 = 5$$

Contribution in the lower-level objective:

$$\Phi_a(2) = \int_0^2 (1 + s^2)\, ds = \left[s + \frac{s^3}{3} \right]_0^2 = \frac{14}{3}$$

Integrating from 0 to s accumulates the travel time of incremental drivers.



Contents

Followers' dual

To derive KKT conditions, we first introduce dual variables for our constraints:

- π_{ov}: Dual variable for flow conservation.
- $\lambda_{oa} \geq 0$: Dual variable for non-negativity constraints.

The Full Lagrangian $\mathcal{L}(z, \pi, \lambda)$

$$\mathcal{L} = \sum_{a \in A(x)} \int_0^{\sum_{o \in O} z_{oa}} \theta_a(s) ds$$

$$- \sum_{o \in O} \sum_{v \in V} \pi_{ov} \underbrace{\left(\sum_{a \in A_v^+(x)} z_{oa} - \sum_{a \in A_v^-(x)} z_{oa} - e_{ov} \right)}_{\text{Flow Conservation}}$$

$$- \sum_{o \in O} \sum_{a \in A(x)} \lambda_{oa} z_{oa}$$

Optimality conditions

Taking the gradient of $\mathcal{L}$ with respect to flow on arc $a = (t, h)$:

Optimality conditions

Taking the gradient of $\mathcal{L}$ with respect to flow on arc $a = (t, h)$:

$$\frac{\partial \mathcal{L}}{\partial z_{oa}} = \theta_a \left(\sum_{o' \in O} z_{o'a} \right) - \pi_{oh} + \pi_{ot} - \lambda_{oa} = 0$$

Optimality conditions

Taking the gradient of $\mathcal{L}$ with respect to flow on arc $a = (t, h)$:

$$\frac{\partial \mathcal{L}}{\partial z_{oa}} = \theta_a \left(\sum_{o' \in O} z_{o'a} \right) - \pi_{oh} + \pi_{ot} - \lambda_{oa} = 0$$

Because λ_{oa} is non-negativity, $\lambda_{oa} \geq 0$. We substitute:

$$\theta_a \left(\sum_{o' \in O} z_{o'a} \right) - \pi_{oh} + \pi_{ot} = \lambda_{oa} \geq 0$$

Optimality conditions

Taking the gradient of $\mathcal{L}$ with respect to flow on arc $a = (t, h)$:

$$\frac{\partial \mathcal{L}}{\partial z_{oa}} = \theta_a \left(\sum_{o' \in O} z_{o'a} \right) - \pi_{oh} + \pi_{ot} - \lambda_{oa} = 0$$

Because λ_{oa} is non-negativity, $\lambda_{oa} \geq 0$. We substitute:

$$\theta_a \left(\sum_{o' \in O} z_{o'a} \right) - \pi_{oh} + \pi_{ot} = \lambda_{oa} \geq 0$$

Applying **complementary slackness** ($\lambda_{oa} \, z_{oa} = 0$):

$$\left(\theta_a \left(\sum_{o' \in O} z_{o'a} \right) - \pi_{oh} + \pi_{ot} \right) z_{oa} = 0$$

We have fully eliminated λ from the system.

Optimality conditions

KKT Conditions

$$\sum_{a \in A_v^+(x)} z_{oa} - \sum_{a \in A_v^-(x)} z_{oa} = e_{ov}, \qquad \forall o \in O, v \in V$$

(Primal Feas.)

$$\theta_a\left(\sum_{o \in O} z_{oa}\right) - \pi_{oh} + \pi_{ot} \geq 0, \qquad \forall o \in O, a = (t,h) \in A(x)$$

(Dual Feas.)

$$\left(\theta_a\left(\sum_{o \in O} z_{oa}\right) - \pi_{oh} + \pi_{ot}\right) z_{oa} = 0, \qquad \forall o \in O, a = (t,h) \in A(x)$$

(Comp. Slack.)

$$z \geq 0. \qquad \text{(Non-negativity)}$$

Optimality conditions

Recall $\theta_a(s) = f_a + g_a s^p$.

Step 1: Substitute the specific travel time function into the integral.

$$L(z,\pi) = \sum_{a \in A(x)} \int_0^{\sum z_{oa}} \theta_a(s) ds - \sum_{o \in O, v \in V} \pi_{ov} \text{(Flow Cons.)}$$

$$=$$

$$=$$

$$=$$

Optimality conditions

Recall $\theta_a(s) = f_a + g_a s^p$.

Step 2: Expand the integral and distribute the node potential duals (π).

$$L(z,\pi) = \sum_{a \in A(x)} \int_0^{\sum z_{oa}} \theta_a(s)ds - \sum_{o \in O, v \in V} \pi_{ov}(\text{Flow Cons.})$$

$$= \sum_a \left(f_a \left(\sum z_{oa}\right) + \frac{g_a}{p+1} \left(\sum z_{oa}\right)^{p+1} \right) - \sum_{o,a=(t,h)} (\pi_{oh} - \pi_{ot}) z_{oa} + \sum_k e_k (\pi_{o_k d_k} - \pi_{o_k o_k})$$

$$=$$

$$=$$

Optimality conditions

Recall $\theta_a(s) = f_a + g_a s^p$.

Step 3: Regroup per arc. Introduce $(\pi_{oh} - \pi_{ot} - f_a)$ and isolate demand.

$$L(z,\pi) = \sum_{a \in A(x)} \int_0^{\sum z_{oa}} \theta_a(s)ds - \sum_{o \in O, v \in V} \pi_{ov}(\text{Flow Cons.})$$

$$= \sum_a \left(f_a \left(\sum z_{oa}\right) + \frac{g_a}{p+1} \left(\sum z_{oa}\right)^{p+1} \right) - \sum_{o, a=(t,h)} (\pi_{oh} - \pi_{ot}) z_{oa} + \sum_k e_k (\pi_{o_k d_k} - \pi_{o_k o_k})$$

$$= \sum_a \underbrace{\left(\frac{g_a}{p+1} \left(\sum z_{oa}\right)^{p+1} - \sum_o (\pi_{oh} - \pi_{ot} - f_a) z_{oa} \right)}_{L_a(z_a, \pi_a)} + \sum_k e_k (\pi_{o_k d_k} - \pi_{o_k o_k})$$

$$=$$

Optimality conditions

Recall $\theta_a(s) = f_a + g_a s^p$.

Step 4: Isolate the arc-wise component $L_a(z_a, \pi_a)$. The problem now separates!

$$
\begin{aligned}
L(z,\pi) &= \sum_{a \in A(x)} \int_0^{\sum z_{oa}} \theta_a(s)ds - \sum_{o \in O, v \in V} \pi_{ov} (\text{Flow Cons.}) \\
&= \sum_a \left(f_a \left(\sum z_{oa}\right) + \frac{g_a}{p+1} \left(\sum z_{oa}\right)^{p+1} \right) - \sum_{o, a=(t,h)} (\pi_{oh} - \pi_{ot}) z_{oa} + \sum_k e_k (\pi_{o_k d_k} - \pi_{o_k o_k}) \\
&= \sum_a \underbrace{\left(\frac{g_a}{p+1} \left(\sum z_{oa}\right)^{p+1} - \sum_o (\pi_{oh} - \pi_{ot} - f_a) z_{oa} \right)}_{L_a(z_a, \pi_a)} + \sum_k e_k (\pi_{o_k d_k} - \pi_{o_k o_k}) \\
&= \sum_{a \in A(x)} L_a(z_a, \pi_a) + \sum_{k \in K} e_k (\pi_{o_k d_k} - \pi_{o_k o_k})
\end{aligned}
$$

Optimality conditions

Lagrangian dual problem

$$\max_{\pi} \min_{z \geq 0} L(z, \pi) = \max_{\pi} \left\{ \sum_{a \in A(x)} \min_{z \geq 0} L_a(z_a, \pi_a) + \sum_{k \in K} e_k(\pi_{o_k d_k} - \pi_{o_k o_k}) \right\}$$

Solving $\min_{z \geq 0} L_a(z_a, \pi_a)$ analytically yields a piecewise function dependent on g_a [Sugishita et al., 2025]. To linearize it, we introduce η_a:

$$\eta_a = \max_{o \in O} \{\pi_{oh} - \pi_{ot} - f_a\} \Leftrightarrow \eta_a \geq \pi_{oh} - \pi_{ot} - f_a \quad \forall o \in O$$

Dual Formulation

$$\max_{\pi,\eta} \sum_{k \in K} e_k(\pi_{o_k d_k} - \pi_{o_k o_k}) - \sum_{a \in A(x)} \frac{p}{\bar{g}_a^{1/p}(p+1)} \eta_a^{\frac{p+1}{p}}$$

$$\begin{aligned} \text{s.t. } & \eta_a \geq \pi_{oh} - \pi_{ot} - f_a, && \forall o \in O, a = (t,h) \in A(x), \\ & \eta_a = 0, && \forall a \in A(x) : g_a = 0 \\ & \eta \geq 0 \end{aligned}$$

Reformulation

Constraints are linear and the objective is separable and convex (monotropic programming).

Reformulation

Constraints are linear and the objective is separable and convex (monotropic programming).

1. Both follower's primal and dual formulations are feasible $\forall x \in X$.
2. Strong duality holds.
3. Primal/Dual variables are optimal $\iff$ they satisfy the KKT conditions.

Primal Problem

$$\min_{z} \sum_{a \in A(x)} \int_0^{\sum_{o \in O} z_{oa}} \theta_a(s) ds$$

$$\text{s.t.} \sum_{a \in A_v^+(x)} z_{oa} - \sum_{a \in A_v^-(x)} z_{oa} = e_{ov}$$

$$z \geq 0.$$

Dual Problem

$$\max_{\pi,\eta} \sum_k e_k(\pi_{o_d d_d} - \pi_{o_d o_d})$$

$$- \sum_a \frac{p}{\overline{g}_a^{1/p}(p+1)} \eta_a^{\frac{p+1}{p}}$$

$$\text{s.t. } \eta_a \geq \pi_{oh} - \pi_{ot} - f_a$$

$$\eta_a = 0 \; a \in A(x) : g_a = 0$$

$$\eta \geq 0$$

Reformulation

The original Bilevel Network Design Problem:

$$\min_{x,z} \sum_{a\in A} \left(\sum_{o\in O} z_{oa} \right) \theta_a \left(\sum_{o\in O} z_{oa} \right) \quad \text{(System Optimum Objective)}$$

$$\text{s.t. } x \in X,\ z \in Z(x) \quad \text{(Follower's Equilibrium)}$$

Reformulation

The original Bilevel Network Design Problem:

$$\min_{x,z} \sum_{a \in A} \left(\sum_{o \in O} z_{oa} \right) \theta_a \left(\sum_{o \in O} z_{oa} \right) \quad \text{(System Optimum Objective)}$$

$$\text{s.t. } x \in X, \; z \in Z(x) \quad \text{(Follower's Equilibrium)}$$

Dualize-and-combine Formulation

$$\min_{x,z,\pi,\eta} \sum_{a \in A} \left(\sum_{o \in O} z_{oa} \right) \theta_a \left(\sum_{o \in O} z_{oa} \right)$$

$$\text{s.t. } \underbrace{\sum_{a \in A} \int_0^{\sum_{o \in O} z_{oa}} \theta_a(s) ds}_{\text{convex in } z} \leq \underbrace{\sum_{k \in K} e_k (\pi_{o_k d_k} - \pi_{o_k o_k}) - \sum_{a \in A} \frac{p}{g_a^{1/p}(p+1)} \eta_a^{\frac{p+1}{p}}}_{\text{concave in } (\pi,\eta)} \quad \text{(Strong Duality)}$$

$$\sum_{a \in A_v^+} z_{oa} - \sum_{a \in A_v^-} z_{oa} = e_{ov}, \quad z_{oa} \leq \sum_{k \in K_o} e_k x_a \quad \text{(primal constraints)}$$

$$\eta_a \geq \pi_{oh} - \pi_{ot} - f_a - M_{oa}(1 - x_a) \quad \text{(dual constraints)}$$

$$\eta_a = 0 \text{ (if } g_a = 0), z \geq 0, \eta \geq 0, x \in X$$

Contents

The problem

Combinatorial pricing problems (CPP)

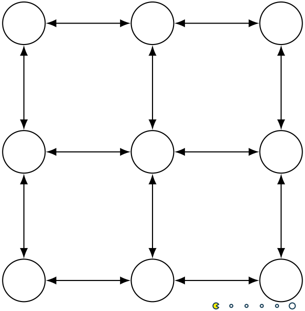
- Leader: sets prices (tolls)
- Follower(s): solves combinatorial problem
  - Shortest path $\rightarrow$ Network pricing problem (NPP)
  - Knapsack $\rightarrow$ Knapsack pricing problem
  - Min set cover pricing, Max stable set pricing, etc.

Natural (non-linear) bilevel formulation

Network pricing problem

$$\max_{t,x} \left\{ \sum_{k \in \mathcal{K}} t^\top x^k \;\middle|\; \begin{array}{ll} t_a \geq 0 & \forall a \in \mathcal{A}_1 \\ t_a = 0 & \forall a \in \mathcal{A}_2 \\ x^k \in \mathbf{R}^k(t) & \forall k \in \mathcal{K} \end{array} \right\}$$

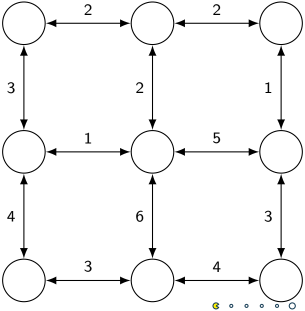
$$\mathbf{R}^k(t) := \operatorname*{argmin}_{x^k} \left\{ (c+t)^\top x^k \;\middle|\; \begin{array}{l} Ax^k = b^k \\ x^k \in \{0,1\}^{|\mathcal{A}|} \end{array} \right\} \qquad \forall k \in \mathcal{K}$$



Network pricing problem

$$\max_{t,x} \left\{ \sum_{k \in \mathcal{K}} t^\top x^k \;\middle|\; \begin{array}{ll} t_a \geq 0 & \forall a \in \mathcal{A}_1 \\ t_a = 0 & \forall a \in \mathcal{A}_2 \\ x^k \in \mathbf{R}^k(t) & \forall k \in \mathcal{K} \end{array} \right\}$$

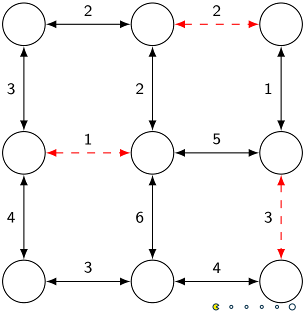
$$\mathbf{R}^k(t) := \operatorname*{argmin}_{x^k} \left\{ (c+t)^\top x^k \;\middle|\; \begin{array}{l} Ax^k = b^k \\ x^k \in \{0,1\}^{|\mathcal{A}|} \end{array} \right\} \qquad \forall k \in \mathcal{K}$$



Network pricing problem

$$\max_{t,x} \left\{ \sum_{k \in \mathcal{K}} t^\top x^k \;\middle|\; \begin{array}{ll} t_a \geq 0 & \forall a \in \mathcal{A}_1 \\ t_a = 0 & \forall a \in \mathcal{A}_2 \\ x^k \in \mathbf{R}^k(t) & \forall k \in \mathcal{K} \end{array} \right\}$$

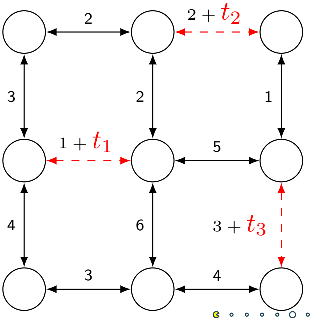
$$\mathbf{R}^k(t) := \operatorname*{argmin}_{x^k} \left\{ (c+t)^\top x^k \;\middle|\; \begin{array}{l} Ax^k = b^k \\ x^k \in \{0,1\}^{|\mathcal{A}|} \end{array} \right\} \qquad \forall k \in \mathcal{K}$$



Network pricing problem

$$\max_{t,x} \left\{ \sum_{k \in \mathcal{K}} t^\top x^k \;\middle|\; \begin{array}{ll} t_a \geq 0 & \forall a \in \mathcal{A}_1 \\ t_a = 0 & \forall a \in \mathcal{A}_2 \\ x^k \in \mathbf{R}^k(t) & \forall k \in \mathcal{K} \end{array} \right\}$$

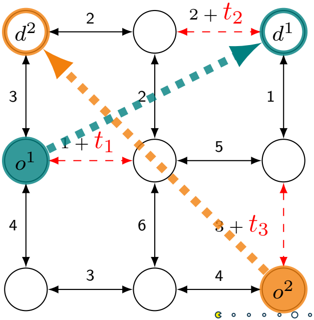
$$\mathbf{R}^k(t) := \operatorname*{argmin}_{x^k} \left\{ (c+t)^\top x^k \;\middle|\; \begin{array}{l} Ax^k = b^k \\ x^k \in \{0,1\}^{|\mathcal{A}|} \end{array} \right\} \qquad \forall k \in \mathcal{K}$$



Network pricing problem

$$\max_{t,x} \left\{ \sum_{k \in \mathcal{K}} t^\top x^k \;\middle|\; \begin{array}{ll} t_a \geq 0 & \forall a \in \mathcal{A}_1 \\ t_a = 0 & \forall a \in \mathcal{A}_2 \\ x^k \in \mathbf{R}^k(t) & \forall k \in \mathcal{K} \end{array} \right\}$$

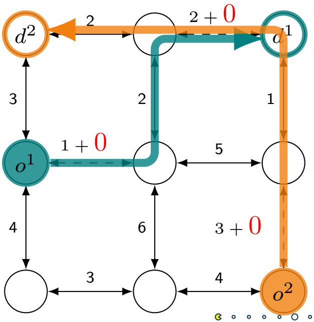
$$\mathbf{R}^k(t) := \operatorname*{argmin}_{x^k} \left\{ (c+t)^\top x^k \;\middle|\; \begin{array}{l} Ax^k = b^k \\ x^k \in \{0,1\}^{|\mathcal{A}|} \end{array} \right\} \qquad \forall k \in \mathcal{K}$$



Network pricing problem

$$\max_{t,x} \left\{ \sum_{k \in \mathcal{K}} t^\top x^k \;\middle|\; \begin{array}{ll} t_a \geq 0 & \forall a \in \mathcal{A}_1 \\ t_a = 0 & \forall a \in \mathcal{A}_2 \\ x^k \in \mathbf{R}^k(t) & \forall k \in \mathcal{K} \end{array} \right\}$$

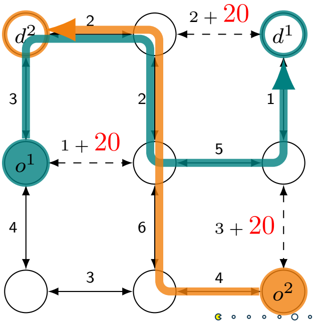
$$\mathbf{R}^k(t) := \operatorname*{argmin}_{x^k} \left\{ (c+t)^\top x^k \;\middle|\; \begin{array}{l} Ax^k = b^k \\ x^k \in \{0,1\}^{|\mathcal{A}|} \end{array} \right\} \qquad \forall k \in \mathcal{K}$$



Network pricing problem

$$\max_{t,x} \left\{ \sum_{k \in \mathcal{K}} t^\top x^k \;\middle|\; \begin{array}{ll} t_a \geq 0 & \forall a \in \mathcal{A}_1 \\ t_a = 0 & \forall a \in \mathcal{A}_2 \\ x^k \in \mathbf{R}^k(t) & \forall k \in \mathcal{K} \end{array} \right\}$$

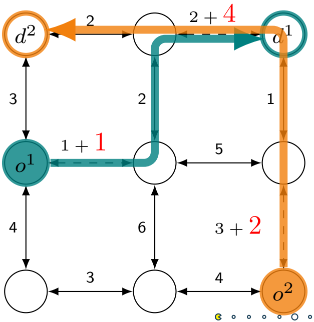
$$\mathbf{R}^k(t) := \operatorname*{argmin}_{x^k} \left\{ (c+t)^\top x^k \;\middle|\; \begin{array}{l} Ax^k = b^k \\ x^k \in \{0,1\}^{|\mathcal{A}|} \end{array} \right\} \qquad \forall k \in \mathcal{K}$$



Network pricing problem

$$\max_{t,x} \left\{ \sum_{k \in \mathcal{K}} t^\top x^k \;\middle|\; \begin{array}{ll} t_a \geq 0 & \forall a \in \mathcal{A}_1 \\ t_a = 0 & \forall a \in \mathcal{A}_2 \\ x^k \in \mathbf{R}^k(t) & \forall k \in \mathcal{K} \end{array} \right\}$$

$$\mathbf{R}^k(t) := \operatorname*{argmin}_{x^k} \left\{ (c+t)^\top x^k \;\middle|\; \begin{array}{l} Ax^k = b^k \\ x^k \in \{0,1\}^{|\mathcal{A}|} \end{array} \right\} \qquad \forall k \in \mathcal{K}$$



rev $= (1+4) + (4+2) = 11$

Network pricing problem

Solving components [Bui et al., 2022]

- Primal representation (of the lower-level)
- Dual representation (of the lower-level)
- Optimality conditions
- Linearization

Network pricing problem

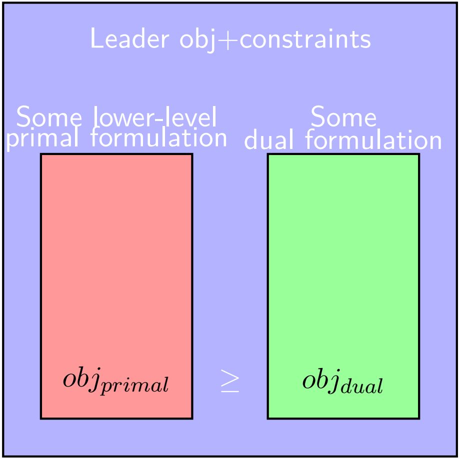
Solving components [Bui et al., 2022]

- Primal representation (of the lower-level)
- Dual representation (of the lower-level)
- Optimality conditions
- Linearization

Primal arc, Dual arc, Strong duality

$$
\begin{aligned}
\max \quad & \sum_{k \in \mathcal{K}} t^\top x^k \\
\text{s.t.} \quad & A x^k = b^k && k \in \mathcal{K}, \\
& x^k \geq 0 && k \in \mathcal{K}, \\
& A^\top y^k \leq c + t && k \in \mathcal{K}, \\
& (c+t)^\top x^k = (b^k)^\top y^k && k \in \mathcal{K}, \\
& t \in \mathcal{T}.
\end{aligned}
$$

Key idea



(if primal is a maximization)

Combinatorial pricing problem

The lower level is a combinatorial optimization problem
expressed as a binary linear program

Combinatorial pricing problem

The lower level is a combinatorial optimization problem
expressed as a binary linear program

What is the lower-level dual? No compact set of optimality conditions

Combinatorial pricing problem

The lower level is a combinatorial optimization problem expressed as a binary linear program

What is the lower-level dual? No compact set of optimality conditions

How to solve the CPP effectively?

Overview

- Literature:
  - Single-commodity NPP:
    well-studied [Labbé et al., 1998, Brotcorne et al., 2001]
  - Multi-commodity NPP: recently studied through the lens of strong bilevel feasibility [Bui et al., 2024]
  - General CPP: only approximation & complexity results
    - Σ_2^p-hard result [Böhnlein et al., 2023, Grüne et al., 2025]
    - NP-hardness: lower-level is a shortest path [Roch et al., 2005] or a minimum spanning tree [Cardinal et al., 2011]
  - Recent technique to formulate the dual of the lower-level [Lozano et al., 2022, Bui et al., 2025, Vásquez et al., 2025]

Overview

- Literature:
  - Single-commodity NPP:
    well-studied [Labbé et al., 1998, Brotcorne et al., 2001]
  - Multi-commodity NPP: recently studied through the lens of strong bilevel feasibility [Bui et al., 2024]
  - General CPP: only approximation & complexity results
    - Σ_2^p-hard result [Böhnlein et al., 2023, Grüne et al., 2025]
    - NP-hardness: lower-level is a shortest path [Roch et al., 2005] or a minimum spanning tree [Cardinal et al., 2011]
  - Recent technique to formulate the dual of the lower-level [Lozano et al., 2022, Bui et al., 2025, Vásquez et al., 2025]
- Goals:
  - Generalize ideas from the NPP to CPPs
  - Obtain tractable reformulations

Contents

Novel ideas [Bui et al., 2025]

Solve the CPP using embedded dynamic programming models

- Formulation of the CPP using dynamic programming
  - Decision diagram
  - Selection diagram
- Dynamic constraint generation

Contents

Knapsack pricing problem

- Items $\mathcal{I}$
  - Weights w_i for $i \in \mathcal{I}$
  - Base profits p_i for $i \in \mathcal{I}$
- Capacity C
- Tolled items $\mathcal{I}_1 \subseteq \mathcal{I}$
  - Tolls t_i for $i \in \mathcal{I}_1$
- Follower's decision x in $\{0,1\}^{\mathcal{I}}$

Knapsack pricing problem

- Items $\mathcal{I}$
  - Weights w_i for $i \in \mathcal{I}$
  - Base profits p_i for $i \in \mathcal{I}$
- Capacity C
- Tolled items $\mathcal{I}_1 \subseteq \mathcal{I}$
  - Tolls t_i for $i \in \mathcal{I}_1$
- Follower's decision x in $\{0,1\}^{\mathcal{I}}$

$$\max_{t,x} \left\{ t^\top x \;\middle|\; \begin{array}{ll} t_i \geq 0 & \forall i \in \mathcal{I}_1 \\ t_i = 0 & \forall i \in \mathcal{I}_2 = \mathcal{I} \setminus \mathcal{I}_1 \\ x \in \mathbf{R}(t) & \end{array} \right\}$$

$$\mathbf{R}(t) := \operatorname*{argmax}_x \left\{ (p-t)^\top x \;\middle|\; \begin{array}{l} w^\top x \leq C \\ x \in \{0,1\}^{\mathcal{I}} \end{array} \right\}$$

Contents

Value function formulation

$$\max_z \left\{ \sum_{\hat{x} \in \mathcal{X}} ((v-t)^\top \hat{x}) z_{\hat{x}} \mid \sum_{\hat{x} \in \mathcal{X}} z_{\hat{x}} = 1,\ z \in \{0,1\}^{|\mathcal{X}|} \right\}$$

$$\min_L \{ L \mid L \geq (v-t)^\top \hat{x},\ \hat{x} \in \mathcal{X} \}.$$

$$\begin{aligned} \max\ & t^\top x \\ \text{s.t. } & w^\top x \leq C, \\ & x \in \{0,1\}^{\mathcal{I}}, \\ & L \geq (p-t)^\top \hat{x}, \qquad \hat{x} \in \mathcal{X}, \\ & (p-t)^\top x = L, \\ & t \in \mathcal{T}. \end{aligned}$$

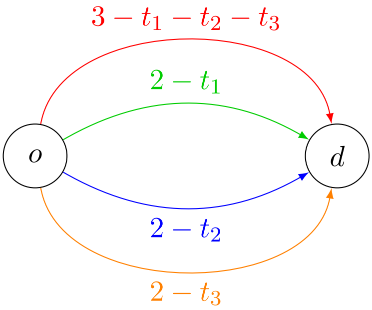
- $L \geq (p-t)^\top \hat{x}$ is generated dynamically
  - Solve master problem for t
  - Solve knapsack problem given t to obtain $\hat{x}$
  - Repeat

Value function example

$\mathcal{I}$	1	2	3	4
w	1	1	1	2
C	3			
p	1	1	1	1
$\hat{x}$	1	1	1	0
	1	0	0	1
	0	1	0	1
	0	0	1	1

$$\begin{aligned} \max\ & t^\top x \\ \text{s.t. } & w^\top x \leq C, \\ & x \in \{0,1\}^{\mathcal{I}}, \\ & L \geq 3 - t_1 - t_2 - t_3, \\ & L \geq 2 - t_1, \\ & L \geq 2 - t_2, \\ & L \geq 2 - t_3, \\ & (p-t)^\top x = L, \\ & t \in \mathcal{T}. \end{aligned}$$

VF as Dynamic programming model



$$L = y_o - y_d$$

$$
\begin{aligned}
\max\ & t^\top x \\
\text{s.t. } & w^\top x \le C, \\
& x \in \{0,1\}^{\mathcal{I}}, \\
& L \ge 3 - t_1 - t_2 - t_3, \\
& L \ge 2 - t_1, \\
& L \ge 2 - t_2, \\
& L \ge 2 - t_3, \\
& (p - t)^\top x = L, \\
& t \in \mathcal{T}.
\end{aligned}
$$

VF as Dynamic programming model

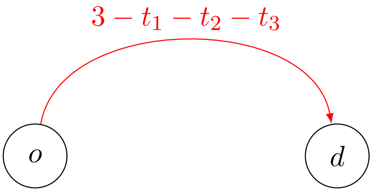
$$\begin{aligned} \max\ & t^\top x \\ \text{s.t. } & w^\top x \le C, \\ & x \in \{0,1\}^{\mathcal{I}}, \end{aligned}$$

$$L = y_o - y_d$$

$$\begin{aligned} & (p-t)^\top x = L, \\ & t \in \mathcal{T}. \end{aligned}$$

Remark: We keep the primal representation and not the path formulation.

VF as Dynamic programming model

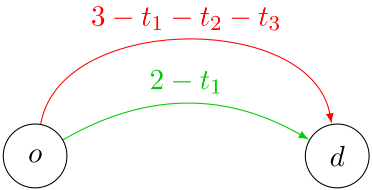


$$\begin{aligned} \max\ & t^\top x \\ \text{s.t. } & w^\top x \le C, \\ & x \in \{0,1\}^{\mathcal{I}}, \\ & L \ge 3 - t_1 - t_2 - t_3, \\ & (p-t)^\top x = L, \\ & t \in \mathcal{T}. \end{aligned}$$

$$L = y_o - y_d$$

Remark: We keep the primal representation and not the path formulation.

VF as Dynamic programming model

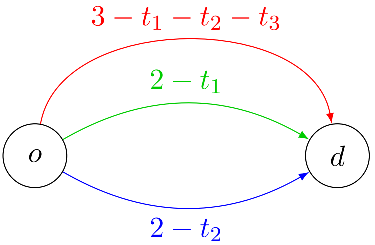


$$
\begin{aligned}
\max\ & t^\top x \\
\text{s.t. } & w^\top x \leq C, \\
& x \in \{0,1\}^{\mathcal{I}}, \\
& L \geq 3 - t_1 - t_2 - t_3, \\
& L \geq 2 - t_1, \\
& (p - t)^\top x = L, \\
& t \in \mathcal{T}.
\end{aligned}
$$

$$L = y_o - y_d$$

Remark: We keep the primal representation and not the path formulation.

VF as Dynamic programming model

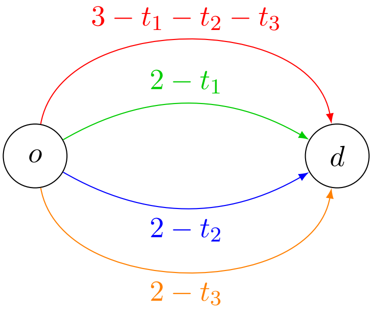


$$
\begin{aligned}
\max\ & t^\top x \\
\text{s.t. } & w^\top x \leq C, \\
& x \in \{0,1\}^{\mathcal{I}}, \\
& L \geq 3 - t_1 - t_2 - t_3, \\
& L \geq 2 - t_1, \\
& L \geq 2 - t_2, \\
& \\
& (p - t)^\top x = L, \\
& t \in \mathcal{T}.
\end{aligned}
$$

$$L = y_o - y_d$$

Remark: We keep the primal representation and not the path formulation.

VF as Dynamic programming model

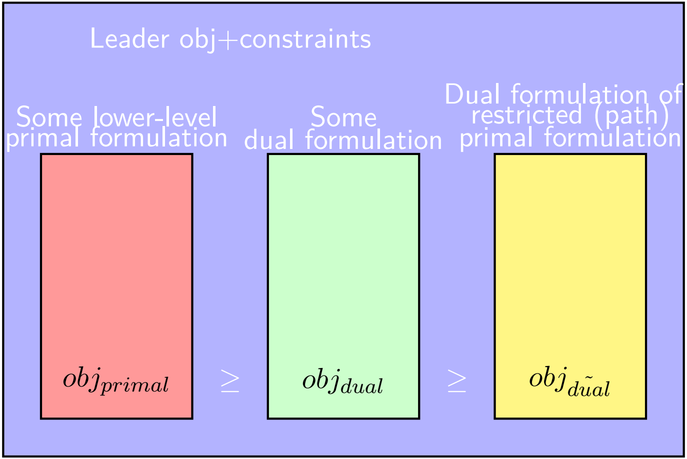


$$L = y_o - y_d$$

$$
\begin{aligned}
\max\ & t^\top x \\
\text{s.t. } & w^\top x \leq C, \\
& x \in \{0,1\}^{\mathcal{I}}, \\
& L \geq 3 - t_1 - t_2 - t_3, \\
& L \geq 2 - t_1, \\
& L \geq 2 - t_2, \\
& L \geq 2 - t_3, \\
& (p - t)^\top x = L, \\
& t \in \mathcal{T}.
\end{aligned}
$$

Remark: We keep the primal representation and not the path formulation.

Key idea

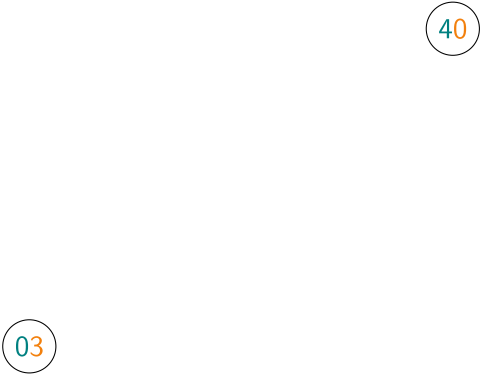


(if primal is a maximization)

Contents

Decision diagram

$$\max\ t^\top x \quad \text{s.t. } w^\top x \le C,\quad x \in \{0,1\}^{\mathcal{I}},$$



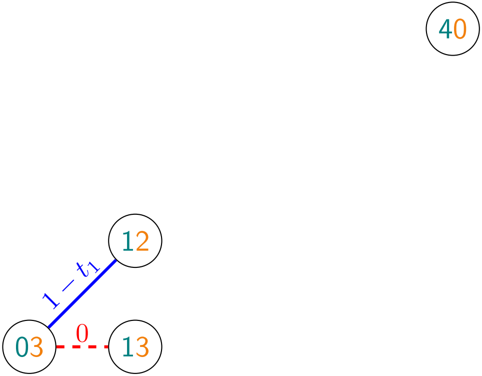
Layer, Remaining capacity

Solid: $x_i = 1$

Dashed: $x_i = 0$

$$(p - t)^\top x = y_{03} - y_{40},$$
$$t \in \mathcal{T}.$$

Decision diagram



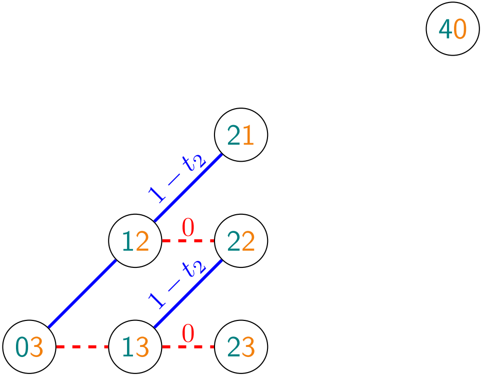
$$
\begin{aligned}
\max\ & t^\top x \\
\text{s.t. } & w^\top x \le C, \\
& x \in \{0,1\}^{\mathcal{I}}, \\
& {\color{red} y_{03} - y_{13} \ge 0}, \\
& {\color{blue} y_{03} - y_{12} \ge 1 - t_1}, \\
& (p - t)^\top x = y_{03} - y_{40}, \\
& t \in \mathcal{T}.
\end{aligned}
$$

Layer, Remaining capacity

Solid: $x_i = 1$

Dashed: $x_i = 0$

Decision diagram



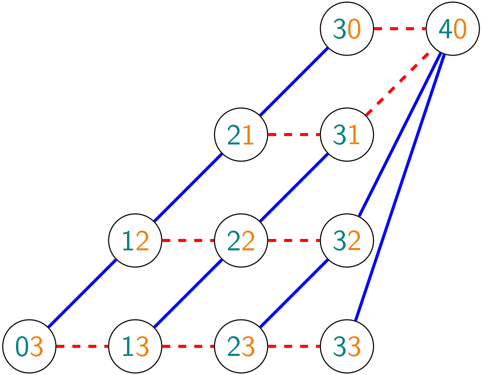
Layer, Remaining capacity

Solid: $x_i = 1$

Dashed: $x_i = 0$

$$
\begin{aligned}
\max\ & t^\top x \\
\text{s.t. } & w^\top x \le C, \\
& x \in \{0,1\}^{\mathcal{I}}, \\
& y_{03} - y_{13} \ge 0, \\
& y_{03} - y_{12} \ge 1 - t_1, \\
& y_{12} - y_{22} \ge 0, \\
& y_{13} - y_{23} \ge 0, \\
& y_{12} - y_{21} \ge 1 - t_2, \\
& y_{13} - y_{22} \ge 1 - t_2, \\
& (p - t)^\top x = y_{03} - y_{40}, \\
& t \in \mathcal{T}.
\end{aligned}
$$

Decision diagram

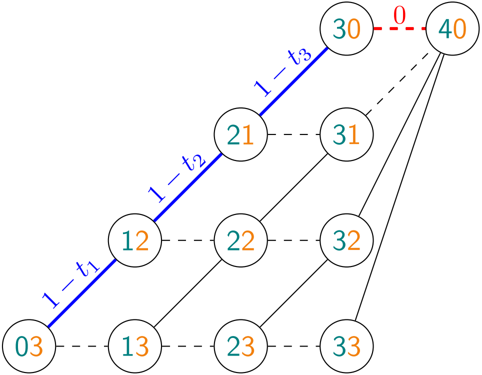


Layer, Remaining capacity
Solid: $x_i = 1$
Dashed: $x_i = 0$

$$
\begin{aligned}
\max\ & t^\top x \\
\text{s.t. } & w^\top x \le C, \\
& x \in \{0,1\}^{\mathcal{I}}, \\
& y_{03} - y_{13} \ge 0, \\
& y_{03} - y_{12} \ge 1 - t_1, \\
& y_{12} - y_{22} \ge 0, \\
& y_{13} - y_{23} \ge 0, \\
& y_{12} - y_{21} \ge 1 - t_2, \\
& y_{13} - y_{22} \ge 1 - t_2, \\
& \vdots \\
& (p-t)^\top x = y_{03} - y_{40}, \\
& t \in \mathcal{T}.
\end{aligned}
$$

Paths in decision diagram

$\hat{x} \in \mathcal{X} \leftrightarrow$ Paths in decision diagram

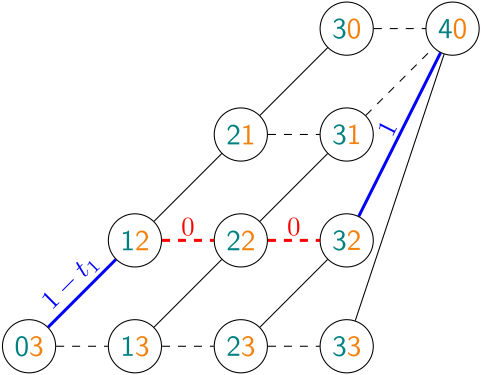


$$\hat{x} = (1, 1, 1, 0)$$

$$(p - t)^\top \hat{x} = 3 - t_1 - t_2 - t_3$$

Paths in decision diagram

$\hat{x} \in \mathcal{X} \leftrightarrow$ Paths in decision diagram

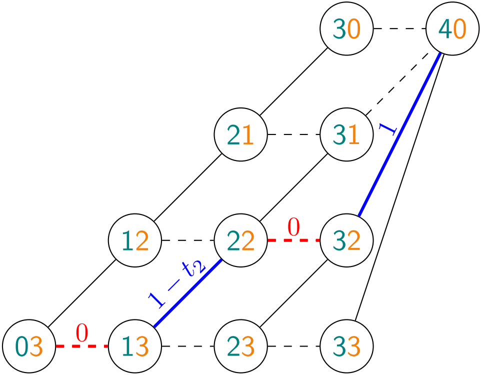


$$\hat{x} = (1, 0, 0, 1)$$

$$(p - t)^{\top} \hat{x} = 2 - t_1$$

Paths in decision diagram

$\hat{x} \in \mathcal{X} \leftrightarrow$ Paths in decision diagram

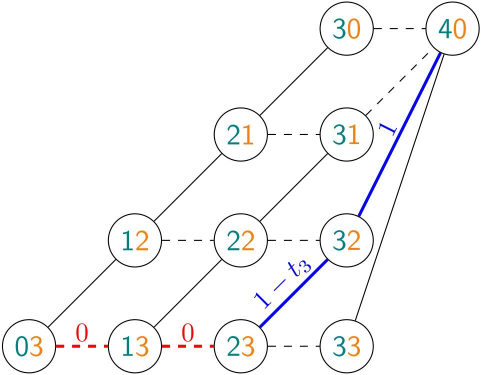


$$\hat{x} = (0, 1, 0, 1)$$

$$(p - t)^\top \hat{x} = 2 - t_2$$

Paths in decision diagram

$\hat{x} \in \mathcal{X} \leftrightarrow$ Paths in decision diagram

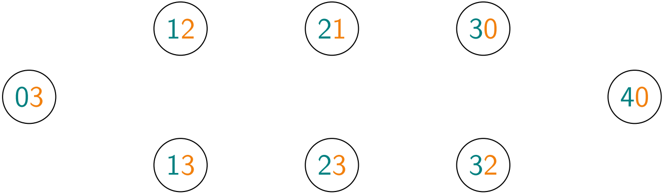


$$\hat{x} = (0, 0, 1, 1)$$

$$(p - t)^\top \hat{x} = 2 - t_3$$

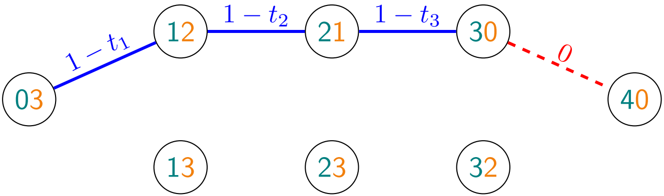
Dynamically-generated constraints in DD

- Start with some nodes
- Given $\hat{x}$, try to fit a path → new constraints
- Worst-case: value function constraint



Dynamically-generated constraints in DD

- Start with some nodes
- Given $\hat{x}$, try to fit a path $\rightarrow$ new constraints
- Worst-case: value function constraint

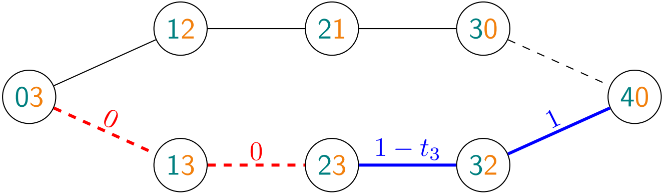


$$\hat{x} = (1,1,1,0)$$

$$(p-t)^\top \hat{x} = 3 - t_1 - t_2 - t_3$$

Dynamically-generated constraints in DD

- Start with some nodes
- Given $\hat{x}$, try to fit a path → new constraints
- Worst-case: value function constraint

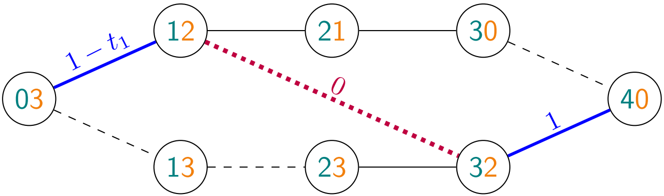


$$\hat{x} = (0, 0, 1, 1)$$

$$(p - t)^\top \hat{x} = 2 - t_3$$

Dynamically-generated constraints in DD

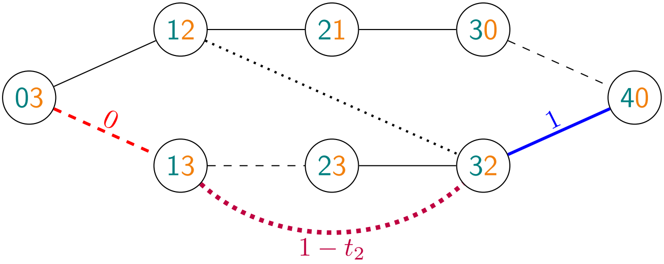
- Start with some nodes
- Given $\hat{x}$, try to fit a path $\rightarrow$ new constraints
- Worst-case: value function constraint



$$\hat{x} = (1,0,0,1)$$
$$(p-t)^\top \hat{x} = 2 - t_1$$

Dynamically-generated constraints in DD

- Start with some nodes
- Given $\hat{x}$, try to fit a path $\rightarrow$ new constraints
- Worst-case: value function constraint



$$\hat{x} = (0, 1, 0, 1)$$

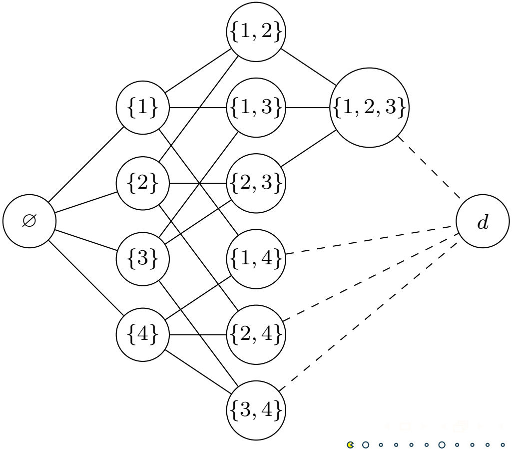
$$(p - t)^\top \hat{x} = 2 - t_2$$

Contents

Selection diagram

At step k:

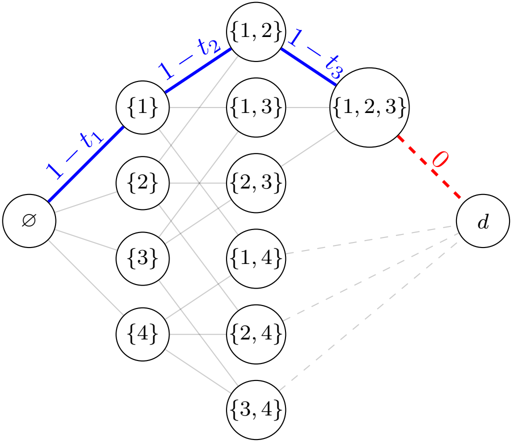
- Decision diagram: Include or exclude the k-th item
- Selection diagram: Include **exactly 1** item



Selection diagram

At step k:

- Decision diagram: Include or exclude the k-th item
- Selection diagram: Include **exactly 1** item



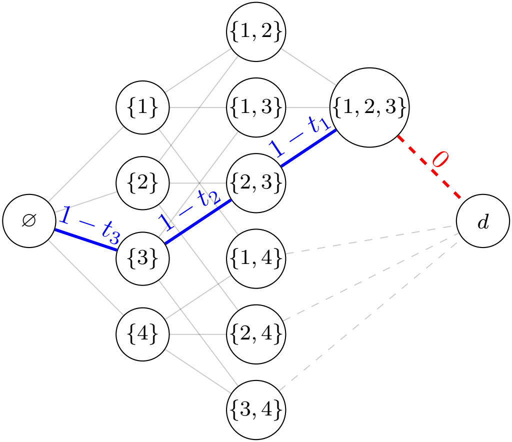
$$\hat{x} = (1,1,1,0)$$

$$(p-t)^\top \hat{x} = 3 - t_1 - t_2 - t_3$$

Selection diagram

At step k:

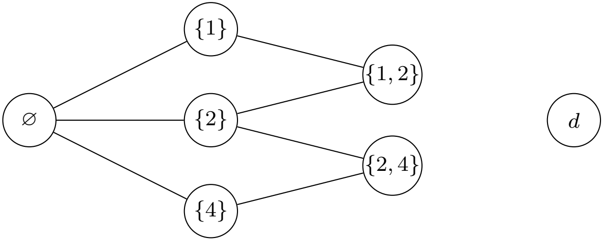
- Decision diagram: Include or exclude the k-th item
- Selection diagram: Include **exactly 1** item



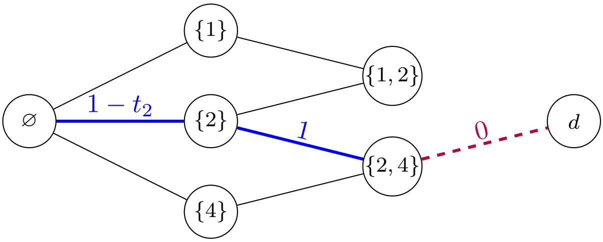
$$\hat{x} = (1, 1, 1, 0)$$

$$(p - t)^\top \hat{x} = 3 - t_1 - t_2 - t_3$$

Dynamically-generated constraints in SD



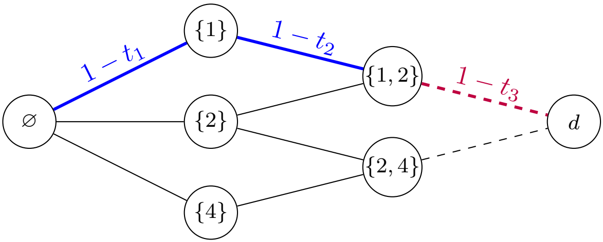
Dynamically-generated constraints in SD



$$\hat{x} = (0, 1, 0, 1)$$

$$(p - t)^\top \hat{x} = 2 - t_2$$

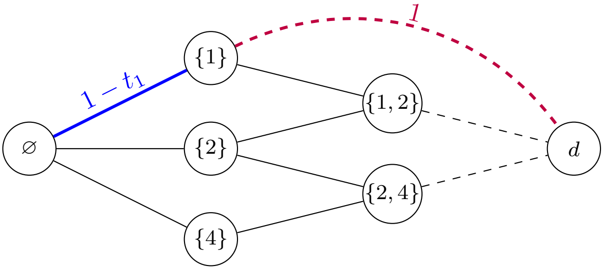
Dynamically-generated constraints in SD



$$\hat{x} = (1,1,1,0)$$

$$(p-t)^\top \hat{x} = 3 - t_1 - t_2 - t_3$$

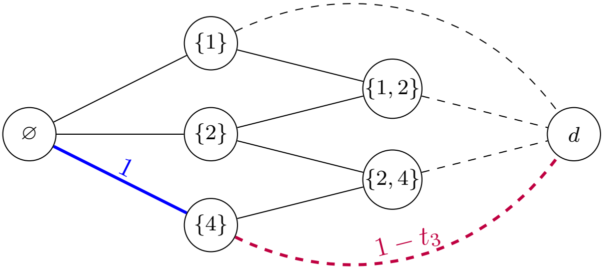
Dynamically-generated constraints in SD



$$\hat{x} = (1, 0, 0, 1)$$

$$(p - t)^\top \hat{x} = 2 - t_1$$

Dynamically-generated constraints in SD



$$\hat{x} = (0, 0, 1, 1)$$

$$(p - t)^\top \hat{x} = 2 - t_3$$

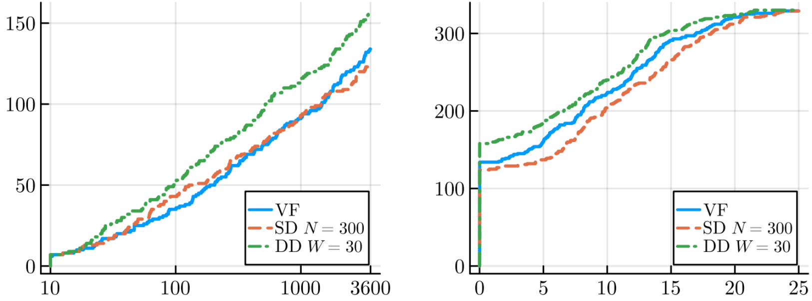
Contents

Experiments

- VF: value function
- SD N: selection diagram with N nodes in layer 3
- DD W: decision diagram with maximum width W and item grouping ($\leq$ 20 layers)
- 1 hour time limit

Experiments: knapsack pricing

- Knapsack pricing (KPP): DD > VF > SD



(a) KPP - Time (s)   (b) KPP - Optimality gap (%)

Cumulative number of instances solved

Maximum stable set pricing problem

$$\mathbf{R}(t) = \arg\max_{x}\{(v-t)^\top x \mid A^\top x \leq \mathbf{1},\ x \in \{0,1\}^{\mathcal{I}}\}$$

where $A \in \{0,1\}^{\mathcal{I}\times\mathcal{E}}$ is the node-edge incidence matrix.

Maximum stable set pricing problem

$$\mathbf{R}(t) = \arg\max_x \{(v-t)^\top x \mid A^\top x \leq \mathbf{1},\ x \in \{0,1\}^{\mathcal{I}}\}$$

where $A \in \{0,1\}^{\mathcal{I}\times\mathcal{E}}$ is the node-edge incidence matrix.

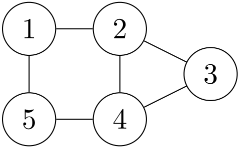
Minimum set cover pricing problem

$$\mathbf{R}(t) = \arg\min_x \{(v+t)^\top x \mid Ax \geq \mathbf{1},\ x \in \{0,1\}^{\mathcal{I}}\}$$

where $A \in \{0,1\}^{\mathcal{E}\times\mathcal{I}}$ is the incidence matrix between $\mathcal{E}$ and $\mathcal{I}$.

MaxSSPP: decision diagram

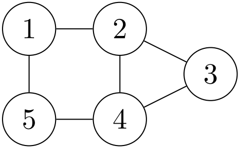
Instance



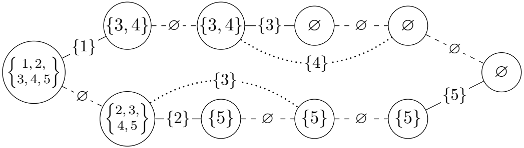
Sampled solutions: $\{1, 3\}, \{2, 5\}$
State: vertices still available

MaxSSPP: decision diagram

Instance



Sampled solutions: $\{1,3\},\{2,5\}$
State: vertices still available



MinSCPP: decision diagram

Instance: $\mathcal{E} = \{a, b, c, d\}$
$\mathcal{I}$ consists of

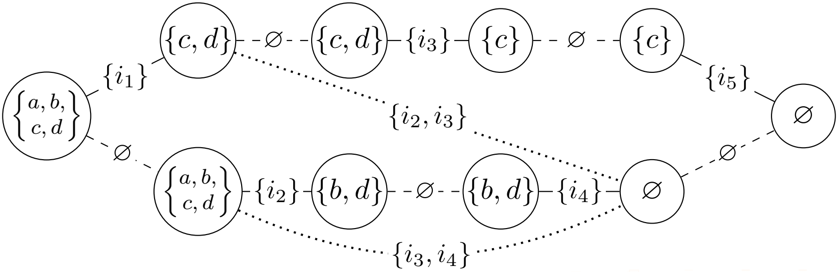
$$i_1 = \{a, b\}, i_2 = \{a, c\}, i_3 = \{a, d\}, i_4 = \{b, c, d\}, i_5 = \{c\}$$

MinSCPP: decision diagram

Instance: $\mathcal{E} = \{a, b, c, d\}$
$\mathcal{I}$ consists of

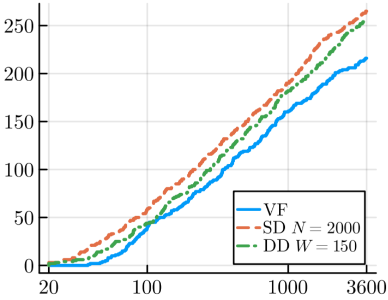
$$i_1 = \{a, b\}, i_2 = \{a, c\}, i_3 = \{a, d\}, i_4 = \{b, c, d\}, i_5 = \{c\}$$

Sampled solutions: $\{i_1, i_3, i_5\}, \{i_2, i_4\}$

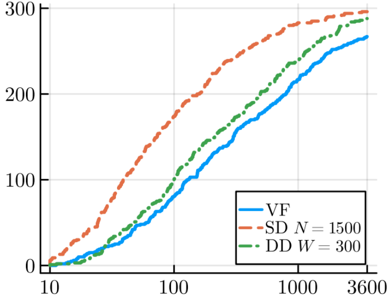


Experiments: Max stable set / Min set cover pricing

- Max stable set pricing (MaxSSPP): SD > DD > VF
- Min set cover pricing (MinSCPP): SD >> DD > VF



(c) MaxSSPP - Time (s)

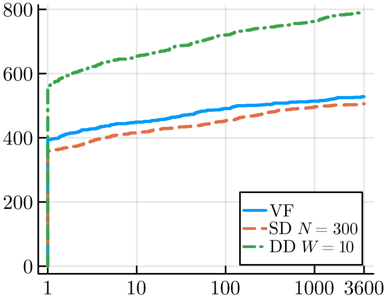


(d) MinSCPP - Time (s)

Cumulative number of instances solved

Experiments: Knapsack interdiction problem

- Knapsack interdiction problem (KIP): DD >> VF > SD

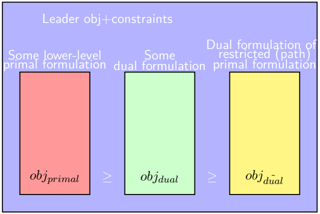


Cumulative number of instances solved

Contents

Summary

- Value function reformulations are oblivious to the follower's primal and dual representations.
- The dual of a follower's primal restriction is a lower bound to the optimal follower's reaction.
- The dual of the restricted lower-level can be dynamically improved to, at the limit, represent the dual of the lower-level.
- Branch-and-cut methodology, essentially improving naive value function representations.



Contents

[Vásquez et al., 2025]: single level reformulations for 0-1 linear bilevel programs.

$$\min_{x,y} \left\{ c_L^\top x + c_F^\top y \;\middle|\; \begin{array}{l} Ax + By \leq a \\ x \in \{0,1\}^{n_L} \\ y \in \mathbf{R}(x) \end{array} \right\}$$

$$\mathbf{R}(x) := \operatorname*{argmin}_{\bar{y}} \left\{ d^\top \bar{y} \;\middle|\; \begin{array}{l} D\bar{y} \leq b - Cx \\ \bar{y} \in \{0,1\}^{n_F} \end{array} \right\}$$

[Vásquez et al., 2025]: Value function reformulation

$$\min_{x,y} \left\{ c_L^\top x + c_F^\top y \;\middle|\; \begin{array}{l} Ax + By \leq a \\ Cx + Dy \leq b \\ (x, y) \in \{0,1\}^{n_L + n_F} \\ d^\top y \leq \phi(x) \end{array} \right\}$$

$$\phi(x) := \min_{\bar{y}} \left\{ d^\top \bar{y} \;\middle|\; \begin{array}{l} D\bar{y} \leq b - Cx \\ \bar{y} \in \{0,1\}^{n_F} \end{array} \right\}$$

[Vásquez et al., 2025]: Value function reformulation

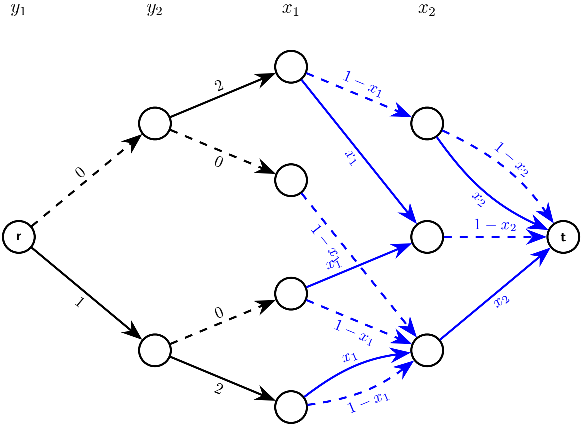
$$\min_{x,y} \left\{ c_L^\top x + c_F^\top y \;\middle|\; \begin{array}{l} Ax + By \leq a \\ Cx + Dy \leq b \\ (x,y) \in \{0,1\}^{n_L+n_F} \\ d^\top y \leq \phi(x) \end{array} \right\}$$

$$\phi(x) := \min_{\bar{y}} \left\{ d^\top \bar{y} \;\middle|\; \begin{array}{l} D\bar{y} \leq b - Cx \\ \bar{y} \in \{0,1\}^{n_F} \end{array} \right\}$$

Transform $\phi(x)$ into a decision diagram.

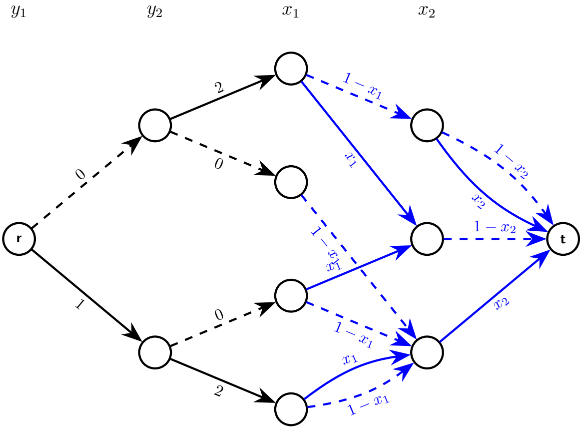
Transform $\phi(x)$ into a decision diagram.

- $n_F + n_L + 1$ layers
- layers for y: from 1 to $n_F + 1$ with value d_i
- layers for x: from $n_F + 2$ to $n_L + 1$ with value 0



[Vásquez et al., 2025]

- Write DD as a capacitated shortest path problem
- Write its dual
- Replace ϕ by the dual



Final thoughts

- Dualize-and-combine is powerful to obtain single-level reformulations or simply to strengthen single-level relaxations.

Final thoughts

- Dualize-and-combine is powerful to obtain single-level reformulations or simply to strengthen single-level relaxations.
- We should focus on "creating more interaction" between primal, dual and upper-level formulations.

References I

Beckmann, M. J., McGuire, C. B., and Winsten, C. B. (1955).
Studies in the Economics of Transportation.
RAND Corporation, Santa Monica, CA.

Brotcorne, L., Labbé, M., Marcotte, P., and Savard, G. (2001).
A bilevel model for toll optimization on a multicommodity transportation network.
Transportation Science, 35(4):345–358.

Bui, Q. M., Carvalho, M., and Neto, J. (2024).
Asymmetry in the complexity of the multi-commodity network pricing problem.
Mathematical Programming, pages 1–37.

Bui, Q. M., Carvalho, M., and Neto, J. (2025).
Solving combinatorial pricing problems using embedded dynamic programming models.
INFORMS Journal on Computing, 38(2):548–567.

Bui, Q. M., Gendron, B., and Carvalho, M. (2022).
A catalog of formulations for the network pricing problem.
INFORMS Journal on Computing, 34(5):2658–2674.

Böhnlein, T., Schaudt, O., and Schauer, J. (2023).
Stackelberg packing games.
Theoretical Computer Science, 943:16–35.

References II

Cardinal, J., Demaine, E. D., Fiorini, S., Joret, G., Langerman, S., Newman, I., and Weimann, O. (2011).
The Stackelberg minimum spanning tree game.
Algorithmica, 59:129–144.

Grüne, C., Henke, D., Rotenberg, E., and Wulf, L. (2025).
The complexity of stackelberg pricing games.
arXiv preprint arXiv:2511.05700.

Labbé, M., Marcotte, P., and Savard, G. (1998).
A bilevel model of taxation and its application to optimal highway pricing.
Management Science, 44(12-part-1):1608–1622.

Lozano, L., Bergman, D., and Cire, A. A. (2022).
Constrained shortest-path reformulations for discrete bilevel and robust optimization.
arXiv preprint arXiv:2206.12962.

Roch, S., Savard, G., and Marcotte, P. (2005).
An approximation algorithm for stackelberg network pricing.
Networks, 46(1):57–67.

Sugishita, N., Sevim, I., Carvalho, M., Dems, A., and Atallah, R. (2025).
Fair network design problem: an application to EV charging station capacity expansion.
Optimization Online.

References III

Vásquez, S., Lozano, L., and van Hoeve, W.-J. (2025).
A single-level reformulation of binary bilevel programs using decision diagrams: S. vásquez et al.
Mathematical Programming, pages 1–54.