Decomposition for Network Design

Bernard Gendron*

March 16-17, 2016

EPFL, Lausanne, Switzerland

* CIRRELT and Département d'informatique et de recherche opérationnelle, Université de Montréal, Canada

Outline of lesson 5: Multicommodity capacitated network design

Fixed-charge problem
- Modelling alternatives
- Cutting-plane method
- Column-and-row generation
- Lagrangian relaxation
- Benders decomposition

Capacity installation case
- General integer formulation
- Binary formulation
- Polyhedral results
- Column-and-row generation

Multicommodity capacitated fixed-charge network design

- Directed network $G = (N, A)$, with node set N and arc set A
- Commodity set K: known demand d^k between origin $O(k)$ and destination $D(k)$ for each $k \in K$
- Unit transportation cost c_{ij} on each arc (i, j)
- Capacity u_{ij} on each arc (i, j)
- Fixed charge f_{ij} incurred whenever arc (i, j) is used to transport some commodity units

Problem formulation (MCNDF)

$$Z = \min \sum_{(i,j)\in A} \sum_{k\in K} c_{ij} x_{ij}^k + \sum_{(i,j)\in A} f_{ij} y_{ij}$$

$$\sum_{j\in N_i^+} x_{ij}^k - \sum_{j\in N_i^-} x_{ji}^k = \begin{cases} d^k, & i = O(k) \\ -\,d^k, & i = D(k) \\ 0, & i \neq O(k), D(k) \end{cases} \qquad i \in N,\ k \in K$$

$$\sum_{k\in K} x_{ij}^k \leq u_{ij} y_{ij} \quad (i,j) \in A$$

$$x_{ij}^k \geq 0 \quad (i,j) \in A,\ k \in K$$

$$y_{ij} \in \{0,1\} \quad (i,j) \in A$$

- How would you solve the LP relaxation? What do you think of the lower bound?

Commodity representation

- Each commodity is identified with one origin and one destination: **disaggregated representation**
- Give an estimate of the maximum number of commodities

Commodity representation

- Each commodity is identified with one origin and one destination: **disaggregated representation**
- Give an estimate of the maximum number of commodities
- In this model, costs and capacities are independent of the commodities
- Use this observation to suggest a different way of representing the commodities

Commodity representation

- Each commodity is identified with one origin and one destination: **disaggregated representation**
- Give an estimate of the maximum number of commodities
- In this model, costs and capacities are independent of the commodities
- Use this observation to suggest a different way of representing the commodities
- All commodities with the same origin can be identified as a single commodity with multiple destinations: **aggregated representation**
- What is the maximum number of commodities in this representation?

Commodity representation

- Each commodity is identified with one origin and one destination: **disaggregated representation**
- Give an estimate of the maximum number of commodities
- In this model, costs and capacities are independent of the commodities
- Use this observation to suggest a different way of representing the commodities
- All commodities with the same origin can be identified as a single commodity with multiple destinations: **aggregated representation**
- What is the maximum number of commodities in this representation?
- $|N|$ instead of $|N|^2$, so much less variables!
- Which representation is better and why?

Basic inequalities

- Suggest simple valid inequalities to improve the LP relaxation

Basic inequalities

- Suggest simple valid inequalities to improve the LP relaxation
- Strong inequality (SI)

$$x_{ij}^k \leq d^k y_{ij} \quad (i,j) \in A, k \in K$$

- $S \subset N$ is a **cutset** if at least one commodity has its origin in S and its destination in $\bar{S} = N \setminus S$
- $(S, \bar{S})$: set of arcs (i,j) that cross cutset S ($i \in S$ and $j \in \bar{S}$)
- $d_{(S,\bar{S})}$: demand of all commodities with origin in S - destination in $\bar{S}$
- Suggest a simple valid inequality based on a cutset $(S, \bar{S})$

Basic inequalities

- Suggest simple valid inequalities to improve the LP relaxation
- Strong inequality (SI)

$$x_{ij}^k \leq d^k y_{ij} \quad (i,j) \in A, k \in K$$

- $S \subset N$ is a **cutset** if at least one commodity has its origin in S and its destination in $\bar{S} = N \setminus S$
- $(S, \bar{S})$: set of arcs (i,j) that cross cutset S ($i \in S$ and $j \in \bar{S}$)
- $d_{(S,\bar{S})}$: demand of all commodities with origin in S - destination in $\bar{S}$
- Suggest a simple valid inequality based on a cutset $(S, \bar{S})$
- Cutset inequality

$$\sum_{(i,j)\in(S,\bar{S})} u_{ij} y_{ij} \geq d_{(S,\bar{S})}$$

- Does this inequality improve the LP relaxation lower bound?

Knapsack inequalities

- **Cover** $C \subseteq (S, \bar{S})$: set of arcs such that $\sum_{(i,j)\in(S,\bar{S})\setminus C} u_{ij} < d_{(S,\bar{S})}$
- Suggest a simple valid inequality based on a cover

Knapsack inequalities

- **Cover** $C \subseteq (S, \bar{S})$: set of arcs such that $\sum_{(i,j)\in(S,\bar{S})\setminus C} u_{ij} < d_{(S,\bar{S})}$
- Suggest a simple valid inequality based on a cover
- Cover inequality (CI)

$$\sum_{(i,j)\in C} y_{ij} \geq 1$$

- l_S: minimum number of arcs in $(S, \bar{S})$ needed to satisfy $d(S, \bar{S})$
- Give an algorithm to compute l_S and a valid inequality based on l_S

Knapsack inequalities

- **Cover** $C \subseteq (S, \bar{S})$: set of arcs such that $\sum_{(i,j) \in (S,\bar{S}) \setminus C} u_{ij} < d_{(S,\bar{S})}$
- Suggest a simple valid inequality based on a cover
- Cover inequality (CI)

$$\sum_{(i,j) \in C} y_{ij} \geq 1$$

- l_S: minimum number of arcs in $(S, \bar{S})$ needed to satisfy $d(S, \bar{S})$
- Give an algorithm to compute l_S and a valid inequality based on l_S
- Minimum cardinality inequality (MCI)

$$\sum_{(i,j) \in (S,\bar{S})} y_{ij} \geq l_S$$

Flow cover/pack inequalities

- Flow cover inequality (FCI)

$$\sum_{(i,j)\in C_1} (x_{ij}^L + (b_{ij}^L - \mu)^+ (1 - y_{ij})) \quad \leq \quad \sum_{(j,i)\in D_2} \min\{b_{ji}^L, \mu\} y_{ji} + \sum_{(j,i)\in C_2} b_{ji}^L + d_{(S,\bar{S})}^L + \sum_{(j,i)\in(\bar{S},S)\setminus C_2\cup D_2} x_{ji}^L$$

- Flow pack inequality (FPI)

$$\sum_{(i,j)\in C_1} x_{ij}^L + \sum_{(i,j)\in D_1} (x_{ij}^L - \min\{b_{ij}^L, -\mu\} y_{ij})) \quad \leq \quad - \sum_{(j,i)\in C_2} (b_{ji}^L + \mu)^+ (1 - y_{ji}) + \sum_{(j,i)\in(\bar{S},S)\setminus C_2} x_{ji}^L + \sum_{(i,j)\in C_1} b_{ij}^L$$

Cutting-plane method

- Starting with the weak LP relaxation, iteratively add violated valid inequalities:
  - To be more efficient: keep the problem size as small as possible
  - To be more effective: improve the lower bound
- But the black-box solver already does that, so why not simply use it?
- True, but we can be more efficient and more effective by exploiting the structure of MCNDF
- Five classes of valid inequalities:
  - Strong inequalities (SI)
  - Cover inequalities (CI)
  - Minimum cardinality inequalities (MCI)
  - Flow cover inequalities (FCI)
  - Flow pack inequalities (FPI)

Cutting-plane method: computational results

- The disaggregated commodity representation outperforms the aggregated one, even when all inequalities are generated
- Single-node cutset inequalities are almost as effective as multi-node cutset inequalities, but much faster to generate
- On instances with few commodities O(10) and many nodes O(100), FCI/FPI are the most effective (but costly)
- On instances with many commodities O(100) and few nodes O(10), SI are the most effective and fast to generate
- Cut-and-branch with CPLEX: our cutting-plane method is competitive with CPLEX own cutting-plane
  - Slower on instances with few commodities O(10) and many nodes O(100)
  - Faster on instances with many commodities O(100) and few nodes O(10)
- A branch-and-cut algorithm has been developed based on the same cutting-plane method

Column-and-row generation

- Extension of the cutting-plane method
- At each iteration, not all the flow variables are generated
- Flow variables are gradually added to the LP relaxation by pricing the variables:
  - Solve the restricted LP relaxation
  - Compute the reduced costs of non-generated flow variables
  - Add (some of) the variables with negative reduced costs
- The restricted LP relaxation is solved by the cutting-plane method (only SI are added)

Restricted LP relaxation

$$\min \sum_{(i,j)\in A_+} \sum_{k\in\widetilde{K}_{ij}} c_{ij} x_{ij}^k + \sum_{(i,j)\in A_+} f_{ij} y_{ij}$$

$$\sum_{j\in\widetilde{N}_i^+(k)} x_{ij}^k - \sum_{j\in\widetilde{N}_i^-(k)} x_{ji}^k = \begin{cases} d^k, & \text{if } i = O(k) \\ -d^k, & \text{if } i = D(k) \\ 0, & \text{otherwise} \end{cases} \quad i \in N,\ k \in K\ (\pi_i^k)$$

$$\sum_{k\in\widetilde{K}_{ij}} x_{ij}^k \leq u_{ij} y_{ij} \quad (i,j) \in A_+ \quad (\alpha_{ij})$$

$$x_{ij}^k \leq d^k y_{ij}, \quad (i,j) \in A_+,\ k \in \overline{K}_{ij} \subseteq \widetilde{K}_{ij} \quad (\beta_{ij}^k)$$

$$y_{ij} \leq 1, \quad (i,j) \in A_+ \quad (\gamma_{ij})$$

$$x_{ij}^k \geq 0, \quad (i,j) \in A_+,\ k \in \widetilde{K}_{ij}$$

$$y_{ij} \geq 0, \quad (i,j) \in A_+$$

LP dual

LP dual

$$\max \sum_{k \in K} d^k \left(\pi^k_{D(k)} - \pi^k_{O(k)} \right) - \sum_{(i,j) \in A} \gamma_{ij}$$

LP dual

$$\max \sum_{k \in K} d^k \left(\pi^k_{D(k)} - \pi^k_{O(k)} \right) - \sum_{(i,j) \in A} \gamma_{ij}$$

$$\pi^k_j - \pi^k_i - \alpha_{ij} - \beta^k_{ij} \leq c_{ij}, \quad (i,j) \in A,\ k \in K \quad (x^k_{ij})$$

LP dual

$$\max \sum_{k \in K} d^k \left(\pi^k_{D(k)} - \pi^k_{O(k)} \right) - \sum_{(i,j) \in A} \gamma_{ij}$$

$$\pi^k_j - \pi^k_i - \alpha_{ij} - \beta^k_{ij} \leq c_{ij}, \quad (i,j) \in A,\ k \in K \quad (x^k_{ij})$$

$$u_{ij}\alpha_{ij} + \sum_{k \in K} d^k \beta^k_{ij} - \gamma_{ij} \leq f_{ij}, \quad (i,j) \in A \quad (y_{ij})$$

LP dual

$$\max \sum_{k \in K} d^k \left(\pi^k_{D(k)} - \pi^k_{O(k)} \right) - \sum_{(i,j) \in A} \gamma_{ij}$$

$$\pi^k_j - \pi^k_i - \alpha_{ij} - \beta^k_{ij} \leq c_{ij}, \quad (i,j) \in A,\ k \in K \quad (x^k_{ij})$$

$$u_{ij}\alpha_{ij} + \sum_{k \in K} d^k \beta^k_{ij} - \gamma_{ij} \leq f_{ij}, \quad (i,j) \in A \quad (y_{ij})$$

$$\alpha_{ij} \geq 0, \quad (i,j) \in A$$

$$\beta^k_{ij} \geq 0, \quad (i,j) \in A,\ k \in K$$

$$\gamma_{ij} \geq 0, \quad (i,j) \in A$$

Complementary slackness conditions

Complementary slackness conditions

$$x_{ij}^k \left(c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij} + \beta_{ij}^k \right) = 0, \quad (i,j) \in A,\ k \in K$$

Complementary slackness conditions

$$x_{ij}^k \left(c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij} + \beta_{ij}^k \right) = 0, \quad (i,j) \in A,\ k \in K$$

$$y_{ij} \left(f_{ij} + \gamma_{ij} - u_{ij}\alpha_{ij} - \sum_{k \in K} d^k \beta_{ij}^k \right) = 0, \quad (i,j) \in A,$$

Complementary slackness conditions

$$x_{ij}^k \left(c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij} + \beta_{ij}^k \right) = 0, \quad (i,j) \in A,\ k \in K$$

$$y_{ij} \left(f_{ij} + \gamma_{ij} - u_{ij}\alpha_{ij} - \sum_{k \in K} d^k \beta_{ij}^k \right) = 0, \quad (i,j) \in A,$$

$$\alpha_{ij} \left(u_{ij} y_{ij} - \sum_{k \in K} x_{ij}^k \right) = 0, \quad (i,j) \in A,$$

Complementary slackness conditions

$$x_{ij}^k \left(c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij} + \beta_{ij}^k \right) = 0, \quad (i,j) \in A,\ k \in K$$

$$y_{ij} \left(f_{ij} + \gamma_{ij} - u_{ij}\alpha_{ij} - \sum_{k \in K} d^k \beta_{ij}^k \right) = 0, \quad (i,j) \in A,$$

$$\alpha_{ij} \left(u_{ij} y_{ij} - \sum_{k \in K} x_{ij}^k \right) = 0, \quad (i,j) \in A,$$

$$\beta_{ij}^k \left(d^k y_{ij} - x_{ij}^k \right) = 0, \quad (i,j) \in A,\ k \in K,$$

Complementary slackness conditions

$$x_{ij}^k \left(c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij} + \beta_{ij}^k \right) = 0, \quad (i,j) \in A,\ k \in K$$

$$y_{ij} \left(f_{ij} + \gamma_{ij} - u_{ij}\alpha_{ij} - \sum_{k \in K} d^k \beta_{ij}^k \right) = 0, \quad (i,j) \in A,$$

$$\alpha_{ij} \left(u_{ij} y_{ij} - \sum_{k \in K} x_{ij}^k \right) = 0, \quad (i,j) \in A,$$

$$\beta_{ij}^k \left(d^k y_{ij} - x_{ij}^k \right) = 0, \quad (i,j) \in A,\ k \in K,$$

$$\gamma_{ij} \left(1 - y_{ij} \right) = 0, \quad (i,j) \in A.$$

Reduced cost optimality conditions for $(i, j), k$

- $k \in \widetilde{K}_{ij}$: conditions are automatically satisfied
- $k \notin \widetilde{K}_{ij}$: add flow variables that do not satisfy the conditions
- $c_{ij}^k(\pi, \alpha) \equiv c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij}$, $k \in K$, $f_{ij}(\alpha) \equiv f_{ij} - u_{ij}\alpha_{ij}$
- Case 1: $\overline{y}_{ij} > 0$

Reduced cost optimality conditions for $(i, j), k$

- $k \in \widetilde{K}_{ij}$: conditions are automatically satisfied
- $k \notin \widetilde{K}_{ij}$: add flow variables that do not satisfy the conditions
- $c_{ij}^k(\pi, \alpha) \equiv c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij}$, $k \in K$, $f_{ij}(\alpha) \equiv f_{ij} - u_{ij}\alpha_{ij}$
- Case 1: $\overline{y}_{ij} > 0$
  - $\overline{\beta}_{ij}^k(\underbrace{d^k\overline{y}_{ij}}_{>0} - \underbrace{\overline{x}_{ij}^k}_{=0}) = 0 \quad \Longrightarrow \quad \overline{\beta}_{ij}^k = 0, k \notin \widetilde{K}_{ij}$

Reduced cost optimality conditions for $(i, j), k$

- $k \in \widetilde{K}_{ij}$: conditions are automatically satisfied
- $k \notin \widetilde{K}_{ij}$: add flow variables that do not satisfy the conditions
- $c_{ij}^k(\pi, \alpha) \equiv c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij}$, $k \in K$, $f_{ij}(\alpha) \equiv f_{ij} - u_{ij}\alpha_{ij}$
- Case 1: $\overline{y}_{ij} > 0$
  - $\overline{\beta}_{ij}^k(\underbrace{d^k \overline{y}_{ij}}_{>0} - \underbrace{\overline{x}_{ij}^k}_{=0}) = 0 \quad \Longrightarrow \quad \overline{\beta}_{ij}^k = 0, k \notin \widetilde{K}_{ij}$
  - Reduced cost optimality condition: $c_{ij}^k(\overline{\pi}, \overline{\alpha}) \geq 0$
- Case 2: $\overline{y}_{ij} = 0$

Reduced cost optimality conditions for $(i, j), k$

- $k \in \widetilde{K}_{ij}$: conditions are automatically satisfied
- $k \notin \widetilde{K}_{ij}$: add flow variables that do not satisfy the conditions
- $c_{ij}^k(\pi, \alpha) \equiv c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij}$, $k \in K$, $f_{ij}(\alpha) \equiv f_{ij} - u_{ij}\alpha_{ij}$
- Case 1: $\overline{y}_{ij} > 0$
  - $\overline{\beta}_{ij}^k(\underbrace{d^k\overline{y}_{ij}}_{>0} - \underbrace{\overline{x}_{ij}^k}_{=0}) = 0 \quad \Longrightarrow \quad \overline{\beta}_{ij}^k = 0, k \notin \widetilde{K}_{ij}$
  - Reduced cost optimality condition: $c_{ij}^k(\overline{\pi}, \overline{\alpha}) \geq 0$
- Case 2: $\overline{y}_{ij} = 0$
  - $\overline{\gamma}_{ij}(\underbrace{1 - \overline{y}_{ij}}_{\neq 0}) = 0 \Longrightarrow \overline{\gamma}_{ij} = 0 \Longrightarrow f_{ij}(\overline{\alpha}) \geq \sum_{k \in K} d^k \overline{\beta}_{ij}^k$

Reduced cost optimality conditions for $(i, j), k$

- $k \in \widetilde{K}_{ij}$: conditions are automatically satisfied
- $k \notin \widetilde{K}_{ij}$: add flow variables that do not satisfy the conditions
- $c_{ij}^k(\pi, \alpha) \equiv c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij}$, $k \in K$, $f_{ij}(\alpha) \equiv f_{ij} - u_{ij}\alpha_{ij}$
- Case 1: $\overline{y}_{ij} > 0$
  - $\overline{\beta}_{ij}^k(\underbrace{d^k\overline{y}_{ij}}_{>0} - \underbrace{\overline{x}_{ij}^k}_{=0}) = 0 \implies \overline{\beta}_{ij}^k = 0, k \notin \widetilde{K}_{ij}$
  - Reduced cost optimality condition: $c_{ij}^k(\overline{\pi}, \overline{\alpha}) \geq 0$
- Case 2: $\overline{y}_{ij} = 0$
  - $\overline{\gamma}_{ij}(\underbrace{1 - \overline{y}_{ij}}_{\neq 0}) = 0 \implies \overline{\gamma}_{ij} = 0 \implies f_{ij}(\overline{\alpha}) \geq \sum_{k \in K} d^k \overline{\beta}_{ij}^k$
  - But, we have $\overline{\beta}_{ij}^k \geq \max\{0, -c_{ij}^k(\overline{\pi}, \overline{\alpha})\}$

Reduced cost optimality conditions for $(i, j), k$

- $k \in \widetilde{K}_{ij}$: conditions are automatically satisfied
- $k \notin \widetilde{K}_{ij}$: add flow variables that do not satisfy the conditions
- $c_{ij}^k(\pi, \alpha) \equiv c_{ij} + \pi_i^k - \pi_j^k + \alpha_{ij},\ k \in K,\ f_{ij}(\alpha) \equiv f_{ij} - u_{ij}\alpha_{ij}$
- Case 1: $\overline{y}_{ij} > 0$
  - $\overline{\beta}_{ij}^k(\underbrace{d^k \overline{y}_{ij}}_{>0} - \underbrace{\overline{x}_{ij}^k}_{=0}) = 0 \quad \Longrightarrow \quad \overline{\beta}_{ij}^k = 0, k \notin \widetilde{K}_{ij}$
  - Reduced cost optimality condition: $c_{ij}^k(\overline{\pi}, \overline{\alpha}) \geq 0$
- Case 2: $\overline{y}_{ij} = 0$
  - $\overline{\gamma}_{ij}(\underbrace{1 - \overline{y}_{ij}}_{\neq 0}) = 0 \Longrightarrow \overline{\gamma}_{ij} = 0 \Longrightarrow f_{ij}(\overline{\alpha}) \geq \sum_{k \in K} d^k \overline{\beta}_{ij}^k$
  - But, we have $\overline{\beta}_{ij}^k \geq \max\{0, -c_{ij}^k(\overline{\pi}, \overline{\alpha})\}$
  - Reduced cost optimality condition: $c_{ij}^k(\overline{\pi}, \overline{\alpha}) \geq 0$ AND $f_{ij}(\overline{\alpha}) \geq \sum_{k \in K} d^k \max\{0, -c_{ij}^k(\overline{\pi}, \overline{\alpha})\}$

Pricing problem

- Decomposes by arc (i,j)
- $\overline{y}_{ij} > 0$: for any $k \notin \widetilde{K}_{ij}$ such that $c_{ij}^k(\overline{\pi},\overline{\alpha}) < 0$, add flow variables x_{ij}^k
- $\overline{y}_{ij} = 0$ and $f_{ij}(\overline{\alpha}) < \sum_{k \in K} d^k \max\{0, -c_{ij}^k(\overline{\pi},\overline{\alpha})\}$: for any $k \notin \widetilde{K}_{ij}$ such that $c_{ij}^k(\overline{\pi},\overline{\alpha}) < 0$, add flow variables x_{ij}^k
- Make a connection between this pricing problem and a Lagrangian relaxation

Column-and-row generation: computational results

- Column-and-row generation is embedded into branch-and-bound: branch-and-price-and-cut
- How is branching performed?

Column-and-row generation: computational results

- Column-and-row generation is embedded into branch-and-bound: branch-and-price-and-cut
- How is branching performed?
- Simply branch on y_{ij} variables: they appear in both the restricted LP and the pricing problem
- B&P&C is faster and more effective than B&C (B&P&C without column generation)
- B&P&C is faster and more effective than CPLEX on on instances with many commodities O(100) and few nodes O(10)

Lagrangian relaxation

$$Z = \min \sum_{(i,j)\in A} \sum_{k\in K} c_{ij} x_{ij}^k + \sum_{(i,j)\in A} f_{ij} y_{ij}$$

$$\sum_{j\in N_i^+} x_{ij}^k - \sum_{j\in N_i^-} x_{ji}^k = \begin{cases} d^k, & i = O(k) \\ -d^k, & i = D(k) \\ 0, & i \neq O(k), D(k) \end{cases} \quad i \in N,\ k \in K \quad (\pi_i^k)$$

$$\sum_{k\in K} x_{ij}^k \leq u_{ij} y_{ij} \quad (i,j) \in A \quad (\alpha_{ij})$$

$$x_{ij}^k \leq b_{ij}^k y_{ij} \quad (i,j) \in A,\ k \in K \quad (\beta_{ij}^k)$$

$$x_{ij}^k \geq 0 \quad (i,j) \in A,\ k \in K$$

$$y_{ij} \in \{0,1\} \quad (i,j) \in A$$

- Suggest several Lagrangian relaxations

Shortest path relaxation

$$Z(\alpha, \beta) = \min \sum_{(i,j)\in A} \sum_{k\in K} (c_{ij} + \alpha_{ij} + \beta_{ij}^k) x_{ij}^k$$

$$+ \sum_{(i,j)\in A} (f_{ij} - u_{ij}\alpha_{ij} - \sum_{k\in K} b_{ij}^k \beta_{ij}^k) y_{ij}$$

$$\sum_{j\in N_i^+} x_{ij}^k - \sum_{j\in N_i^-} x_{ji}^k = \begin{cases} d^k, & i = O(k) \\ -d^k, & i = D(k) \\ 0, & i \neq O(k), D(k) \end{cases} \qquad i \in N,\ k \in K$$

$$y_{ij} \in \{0,1\} \quad (i,j) \in A$$

- How would you solve this problem?

Shortest path relaxation

$$Z(\alpha, \beta) = \min \sum_{(i,j)\in A} \sum_{k\in K} (c_{ij} + \alpha_{ij} + \beta_{ij}^k) x_{ij}^k$$

$$+ \sum_{(i,j)\in A} (f_{ij} - u_{ij}\alpha_{ij} - \sum_{k\in K} b_{ij}^k \beta_{ij}^k) y_{ij}$$

$$\sum_{j\in N_i^+} x_{ij}^k - \sum_{j\in N_i^-} x_{ji}^k = \begin{cases} d^k, & i = O(k) \\ -d^k, & i = D(k) \\ 0, & i \neq O(k), D(k) \end{cases} \qquad i \in N,\; k \in K$$

$$y_{ij} \in \{0, 1\} \quad (i, j) \in A$$

- How would you solve this problem?
- Does it have the integrality property?

Knapsack relaxation

$$Z(\pi) = \min \sum_{(i,j)\in A} \sum_{k\in K} (c_{ij} + \pi_i^k - \pi_j^k) x_{ij}^k + \sum_{(i,j)\in A} f_{ij} y_{ij} + \sum_{k\in K} d^k (\pi_{D(k)}^k - \pi_{O(k)}^k)$$

$$\sum_{k\in K} x_{ij}^k \leq u_{ij} y_{ij} \quad (i,j) \in A$$

$$x_{ij}^k \leq b_{ij}^k y_{ij} \quad (i,j) \in A,\ k \in K$$

$$x_{ij}^k \geq 0 \quad (i,j) \in A,\ k \in K$$

$$y_{ij} \in \{0,1\} \quad (i,j) \in A$$

- How would you solve this problem?

Knapsack relaxation

$$Z(\pi) = \min \sum_{(i,j)\in A} \sum_{k\in K} (c_{ij} + \pi_i^k - \pi_j^k) x_{ij}^k + \sum_{(i,j)\in A} f_{ij} y_{ij} + \sum_{k\in K} d^k (\pi_{D(k)}^k - \pi_{O(k)}^k)$$

$$\sum_{k\in K} x_{ij}^k \leq u_{ij} y_{ij} \quad (i,j) \in A$$

$$x_{ij}^k \leq b_{ij}^k y_{ij} \quad (i,j) \in A, \; k \in K$$

$$x_{ij}^k \geq 0 \quad (i,j) \in A, \; k \in K$$

$$y_{ij} \in \{0,1\} \quad (i,j) \in A$$

- How would you solve this problem?
- Does it have the integrality property?

Lagrangian relaxation: computational results

- What can you say about the Lagrangian dual lower bounds?

Lagrangian relaxation: computational results

- What can you say about the Lagrangian dual lower bounds?
- Both Lagrangian relaxations provide the same lower bound as the strong LP relaxation
- To find (near-)optimal Lagrangian multipliers, two classes of methods have been traditionally used:
  - Column generation (CG) methods
  - Subgradient methods
- Our computational results show that:
  - CG methods are much more robust
  - CG methods converge faster
  - Any of these two methods converge much faster than solving the strong LP relaxation with the simplex method (without cutting-plane or column-and-row generation)

Benders subproblem

- Fix the design variables to $\bar{y}$ by solving a (MIP) master problem
- Write down the Benders subproblem: what is the structure of this problem?

Benders subproblem

- Fix the design variables to $\overline{y}$ by solving a (MIP) master problem
- Write down the Benders subproblem: what is the structure of this problem?
- Solve the multicommodity flow subproblem restricted to $\overline{y}$:

$$Z_x(\overline{y}) = \min \sum_{(i,j)\in A} \sum_{k\in K} c_{ij} x_{ij}^k$$

$$\sum_{j\in N_i^+} x_{ij}^k - \sum_{j\in N_i^-} x_{ji}^k = \begin{cases} d^k, & i = O(k) \\ -d^k, & i = D(k) \\ 0, & i \neq O(k), D(k) \end{cases} \qquad i \in N,\ k \in K$$

$$\sum_{k\in K} x_{ij}^k \leq u_{ij}\overline{y}_{ij} \quad (i,j) \in A$$

$$x_{ij}^k \geq 0 \quad (i,j) \in A,\ k \in K$$

Dual of the Benders subproblem

- Write down the dual of the Benders subproblem

Dual of the Benders subproblem

- Write down the dual of the Benders subproblem
- The dual of the Benders subproblem has the following structure:

$$Z_x(\bar{y}) = \max \sum_{k \in K} d^k \pi^k_{D(k)} - \sum_{(i,j) \in A} u_{ij} \bar{y}_{ij} \alpha_{ij}$$

$$\pi^k_j - \pi^k_i - \alpha_{ij} \leq c_{ij} \quad (i,j) \in A,\ k \in K$$

$$\alpha_{ij} \geq 0 \quad (i,j) \in A$$

- Note: we can eliminate one of the π multipliers for each commodity ($\pi^k_{O(k)} = 0$)

Benders cuts and master problem

- If the LP dual is bounded, then generate an optimality cut
- Write down the Benders optimality cut

Benders cuts and master problem

- If the LP dual is bounded, then generate an optimality cut
- Write down the Benders optimality cut

$$\sum_{k \in K} d^k \pi^k_{D(k)} - \sum_{(i,j) \in A} u_{ij} y_{ij} \alpha_{ij} \leq z$$

- If the LP dual is unbounded, then generate a feasibility cut
- Write down the Benders feasibility cut

Benders cuts and master problem

- If the LP dual is bounded, then generate an optimality cut
- Write down the Benders optimality cut

$$\sum_{k \in K} d^k \pi^k_{D(k)} - \sum_{(i,j) \in A} u_{ij} y_{ij} \alpha_{ij} \leq z$$

- If the LP dual is unbounded, then generate a feasibility cut
- Write down the Benders feasibility cut

$$\sum_{k \in K} d^k \pi^k_{D(k)} - \sum_{(i,j) \in A} u_{ij} y_{ij} \alpha_{ij} \leq 0$$

- Add the (optimality or feasibility) cut to the master problem defined by the objective:

$$\min \sum_{(i,j) \in A} f_{ij} y_{ij} + z$$

- Find a new $\overline{y}$ and perform another iteration
- This process converges to an optimal solution of MCNDF

Benders decomposition: implementation and results

- Several techniques are implemented to accelerate the convergence of the algorithm:
  - Add cutset inequalities to the master problem
  - Solve the (strong) LP relaxation by Benders decomposition (master problem $\equiv$ LP)
  - Use (slope scaling) heuristic to generate several tentative $\overline{y}$
  - When solving MIP master problem by branch-and-bound, collect all feasible solutions to generate several tentative $\overline{y}$
- Even with these improvements (and many others), the method is not competitive with simplex-based branch-and-cut approach
- But, Benders feasibility cuts can be used in any branch-and-cut approach!

Capacity installation multicommodity network design

- Directed network $G = (N, A)$, with node set N and arc set A
- Commodity set K: known demand d^k between origin $O(k)$ and destination $D(k)$ for each $k \in K$
- Unit transportation cost c_{ij} on each arc (i, j)
- Capacity u_{ij} on each arc (i, j)
- Cost f_{ij} for each capacity unit installed on arc (i, j)

General integer formulation (I)

$$\min \sum_{k \in K} \sum_{(i,j) \in A} d^k c_{ij} x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij}$$

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{ji}^k = \begin{cases} 1, & \text{if } i = O(k) \\ -1, & \text{if } i = D(k) \\ 0, & \text{if } i \neq O(k), D(k) \end{cases} \qquad i \in N,\ k \in K$$

$$\sum_{k \in K} d^k x_{ij}^k \leq u_{ij} y_{ij} \qquad (i,j) \in A$$

$$0 \leq x_{ij}^k \leq 1 \qquad (i,j) \in A,\ k \in K$$

$$y_{ij} \geq 0 \qquad (i,j) \in A$$

$$y_{ij} \ \textit{integer} \qquad (i,j) \in A$$

Lagrangian relaxation of flow conservation

$$\min \sum_{k \in K} \sum_{(i,j) \in A} (d^k c_{ij} + \pi_i^k - \pi_j^k) x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} + \sum_{k \in K} \left(\pi_{D(k)}^k - \pi_{O(k)}^k \right)$$

$$\sum_{k \in K} d^k x_{ij}^k \leq u_{ij} y_{ij} \quad (i,j) \in A$$

$$0 \leq x_{ij}^k \leq 1 \quad (i,j) \in A, \ k \in K$$

$$y_{ij} \geq 0 \quad (i,j) \in A$$

$$y_{ij} \ \textit{integer} \quad (i,j) \in A$$

- Lagrangian subproblem *decomposes by arc*
- Easy ($\approx$ 2 continuous knapsack) but *no* integrality property

Residual capacity inequalities

- For any $P \subseteq K$, define $d^P = \sum_{k \in P} d^k$
- Then, for any $(i,j) \in A$, define

$$a_{ij}^P = \frac{d^P}{u_{ij}}$$

$$q_{ij}^P = \lceil a_{ij}^P \rceil$$

$$r_{ij}^P = a_{ij}^P - \lfloor a_{ij}^P \rfloor$$

- Residual capacity inequalities

$$\sum_{k \in P} \{a_{ij}^k (1 - x_{ij}^k)\} \geq r_{ij}^P (q_{ij}^P - y_{ij}) \quad (i,j) \in A,\ P \subseteq K$$

- Characterize the convex hull of solutions to the Lagrangian subproblem (Magnanti, Mirchandani, Vachani 1993)
- Separation can be performed in $O(|A||K|)$ (Atamtürk, Rajan 2002)

Multiple choice model

$$y_{ij} \leq \left\lceil \frac{\sum_{k \in K} d^k}{u_{ij}} \right\rceil = T_{ij}$$

$$S_{ij} = \{1, \dots, T_{ij}\}$$

$$y_{ij}^s = \begin{cases} 1, & \mathit{if}\ \ y_{ij} = s \\ 0, & \mathit{otherwise} \end{cases} \qquad s \in S_{ij}$$

$$x_{ij}^s = \begin{cases} \sum_{k \in K} d^k x_{ij}^k, & \mathit{if}\ \ y_{ij} = s \\ 0, & \mathit{otherwise} \end{cases} \qquad s \in S_{ij}$$

Binary formulation (B)

$$y_{ij} = \sum_{s \in S_{ij}} s y_{ij}^s \quad (i,j) \in A$$

$$\sum_{k \in K} d^k x_{ij}^k = \sum_{s \in S_{ij}} x_{ij}^s \quad (i,j) \in A$$

$$(s-1) u_{ij} y_{ij}^s \leq x_{ij}^s \leq s u_{ij} y_{ij}^s \quad (i,j) \in A, s \in S_{ij}$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \quad (i,j) \in A$$

$$y_{ij}^s \geq 0 \quad (i,j) \in A, s \in S_{ij}$$

$$y_{ij}^s \ integer \quad (i,j) \in A, s \in S_{ij}$$

Variable disaggregation and extended formulation (B^+)

- Extended auxiliary variables

$$x_{ij}^{ks} = \begin{cases} x_{ij}^{k}, & \textit{if } y_{ij} = s \\ 0, & \textit{otherwise} \end{cases} \qquad s \in S_{ij}$$

$$x_{ij}^{k} = \sum_{s \in S_{ij}} x_{ij}^{ks} \quad (i,j) \in A, k \in K$$

$$x_{ij}^{s} = \sum_{k \in K} d^{k} x_{ij}^{ks} \quad (i,j) \in A, s \in S_{ij}$$

- Extended linking inequalities

$$x_{ij}^{ks} \leq y_{ij}^{s} \quad (i,j) \in A, k \in K, s \in S_{ij}$$

Polyhedral results: notation

- $Z(M)$: optimal value for model M
- $F(M)$: feasible set for model M
- $conv(F(M))$: convex hull of $F(M)$
- $LP(M)$: LP relaxation for model M
- $LS(M)$: Lagrangian subproblem
  (relaxation of flow conservation constraints)
- $LD(M)$: Lagrangian dual for $LS(M)$

Polyhedral results

- **Result 1**:
  - $I^+ = I$ + residual capacity inequalities
  - $F(LP(LS(I^+))) = conv(F(LS(I^+)))$
    (Magnanti, Mirchandani, Vachani 1993)
  - $Z(LP(I^+)) = Z(LD(I))$: primal interpretation of Lagrangian dual!

Polyhedral results

- **Result 1**:
  - $I^+ = I +$ residual capacity inequalities
  - $F(LP(LS(I^+))) = conv(F(LS(I^+)))$
    (Magnanti, Mirchandani, Vachani 1993)
  - $Z(LP(I^+)) = Z(LD(I))$: primal interpretation of Lagrangian dual!
- **Result 2**:
  - $Z(LS(I)) = Z(LS(B^+))$ for the same values of the Lagrange multipliers: apply reformulation to Lagrangian subproblem!
  - $Z(LD(I)) = Z(LD(B^+))$: same constraints relaxed in I and B^+!

Polyhedral results

- **Result 1**:
  - $I^+ = I +$ residual capacity inequalities
  - $F(LP(LS(I^+))) = conv(F(LS(I^+)))$ (Magnanti, Mirchandani, Vachani 1993)
  - $Z(LP(I^+)) = Z(LD(I))$: primal interpretation of Lagrangian dual!
- **Result 2**:
  - $Z(LS(I)) = Z(LS(B^+))$ for the same values of the Lagrange multipliers: apply reformulation to Lagrangian subproblem!
  - $Z(LD(I)) = Z(LD(B^+))$: same constraints relaxed in I and B^+!
- **Result 3**:
  - $F(LP(LS(B^+))) = conv(F(LS(B^+)))$ (Croxton, Gendron, Magnanti 2007)
  - $Z(LD(B^+)) = Z(LP(B^+))$: primal interpretation of Lagrangian dual!

Polyhedral results

- **Result 1**:
  - $I^+ = I +$ residual capacity inequalities
  - $F(LP(LS(I^+))) = conv(F(LS(I^+)))$ (Magnanti, Mirchandani, Vachani 1993)
  - $Z(LP(I^+)) = Z(LD(I))$: primal interpretation of Lagrangian dual!
- **Result 2**:
  - $Z(LS(I)) = Z(LS(B^+))$ for the same values of the Lagrange multipliers: apply reformulation to Lagrangian subproblem!
  - $Z(LD(I)) = Z(LD(B^+))$: same constraints relaxed in I and B^+!
- **Result 3**:
  - $F(LP(LS(B^+))) = conv(F(LS(B^+)))$ (Croxton, Gendron, Magnanti 2007)
  - $Z(LD(B^+)) = Z(LP(B^+))$: primal interpretation of Lagrangian dual!
- $Z(LP(I^+)) = Z(LD(I)) = Z(LD(B^+)) = Z(LP(B^+))$ (Frangioni, Gendron 2009)

Column-and-row generation for $LP(B^+)$

- Only a small subset of the x_{ij}^{ks} *and* y_{ij}^{s} variables are generated
- Variables are gradually added to the LP relaxation by pricing them:
  - Solve the restricted LP relaxation
  - Compute the reduced costs of non-generated flow variables $\equiv$ solve the Lagrangian subproblem
  - Add variables with negative reduced costs
- The restricted LP relaxation is solved by the cutting-plane method: constraints $x_{ij}^{ks} \leq y_{ij}^{s}$ are gradually added

Column-and-row generation: computational results

- Implementation: solving the LP relaxation, then freezing the formulation + `CPLEX` heuristics for one hour
- Comparison of three model/method combinations:
  - B^+: Binary model (pseudo-polynomial number of variables and constraints)/ column-and-row generation (easy pricing)
  - I^+: General integer model (exponential number of constraints)/cutting-plane (easy separation)
  - DW: Dantzig-Wolfe Lagrangian dual (exponential number of variables)/column generation (easy pricing)

Column-and-row generation: computational results

- Implementation: solving the LP relaxation, then freezing the formulation + `CPLEX` heuristics for one hour
- Comparison of three model/method combinations:
  - B^+: Binary model (pseudo-polynomial number of variables and constraints)/ column-and-row generation (easy pricing)
  - I^+: General integer model (exponential number of constraints)/cutting-plane (easy separation)
  - DW: Dantzig-Wolfe Lagrangian dual (exponential number of variables)/column generation (easy pricing)
- B^+ is much faster than DW
- B^+ is generally faster AND more effective (better upper bounds) than I^+
- As $|K|$ increases, the advantage of B^+ over I^+ increases
- Additional features (subgradient warmstart, stabilization) improve efficiency (time) AND effectiveness (upper bounds)