Decomposition for Network Design

Bernard Gendron*

March 2-10, 2016

EPFL, Lausanne, Switzerland

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

Outline of lesson 4: Classic examples of decomposition

Branch-and-cut for traveling salesman

Branch-and-price for integer multicommodity flow

Lagrangian relaxation for budget network design

Benders decomposition for uncapacitated facility location

Asymmetric traveling salesman problem (ATSP)

- Directed network $G = (N, A)$, with node set $N = \{1, 2, \ldots, n\}$ and arc set A
- Routing cost c_{ij} on each arc (i, j) (can be different than c_{ji}, routing cost on arc (j, i))
- Problem description: find the minimum cost Hamiltonian circuit, i.e., a circuit that visits each node in N exactly one time (we assume such a circuit exists in G)
- Useful notation: for any $S, T \subseteq N$, let $A(S, T) = \{(i, j) \in A | i \in S, j \in T\}$

Problem formulation

- y_{ij}: 1, if arc (i, j) is chosen in the Hamiltonian circuit, 0, otherwise

Problem formulation

- y_{ij}: 1, if arc (i, j) is chosen in the Hamiltonian circuit, 0, otherwise

$$Z = \min \sum_{(i,j) \in A} c_{ij} y_{ij}$$

Problem formulation

- y_{ij}: 1, if arc (i,j) is chosen in the Hamiltonian circuit, 0, otherwise

$$Z = \min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

$$\sum_{j\in N_i^+} y_{ij} = 1 \quad i \in N$$

$$\sum_{j\in N_i^-} y_{ji} = 1 \quad i \in N$$

Problem formulation

- y_{ij}: 1, if arc (i,j) is chosen in the Hamiltonian circuit, 0, otherwise

$$Z = \min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

$$\sum_{j\in N_i^+} y_{ij} = 1 \quad i \in N$$

$$\sum_{j\in N_i^-} y_{ji} = 1 \quad i \in N$$

$$\sum_{(i,j)\in A(S,N\setminus S)} y_{ij} \geq 1, \quad S \subset N, S \neq \emptyset$$

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

Alternative formulations

- If routing costs are symmetric ($c_{ij} = c_{ji}$ for each (i, j)), how would you formulate the problem?

Alternative formulations

- If routing costs are symmetric ($c_{ij} = c_{ji}$ for each (i, j)), how would you formulate the problem?
- Think of another way of expressing **subtour elimination constraints (SEC)** (using the same variables)

Alternative formulations

- If routing costs are symmetric ($c_{ij} = c_{ji}$ for each (i, j)), how would you formulate the problem?
- Think of another way of expressing **subtour elimination constraints (SEC)** (using the same variables)
- Does the resulting model have an equivalent LP relaxation?

Alternative formulations

- If routing costs are symmetric ($c_{ij} = c_{ji}$ for each (i, j)), how would you formulate the problem?
- Think of another way of expressing **subtour elimination constraints (SEC)** (using the same variables)
- Does the resulting model have an equivalent LP relaxation?
- Both models have an exponential number of constraints: what to do?

Alternative formulations

- If routing costs are symmetric ($c_{ij} = c_{ji}$ for each (i, j)), how would you formulate the problem?
- Think of another way of expressing **subtour elimination constraints (SEC)** (using the same variables)
- Does the resulting model have an equivalent LP relaxation?
- Both models have an exponential number of constraints: what to do?
- Introduce a set of commodities $K = N \setminus \{1\}$ such that $O(k) = 1$, $D(k) = k$ and $d^k = 1$ for each $k \in K$
- Define multicommodity flow variables x_{ij}^k: flow of commodity k on arc (i, j)
- Using these multicommodity flow variables, express SECs with a polynomial number of constraints

Alternative formulations

- If routing costs are symmetric ($c_{ij} = c_{ji}$ for each (i, j)), how would you formulate the problem?
- Think of another way of expressing **subtour elimination constraints (SEC)** (using the same variables)
- Does the resulting model have an equivalent LP relaxation?
- Both models have an exponential number of constraints: what to do?
- Introduce a set of commodities $K = N \setminus \{1\}$ such that $O(k) = 1$, $D(k) = k$ and $d^k = 1$ for each $k \in K$
- Define multicommodity flow variables x_{ij}^k: flow of commodity k on arc (i, j)
- Using these multicommodity flow variables, express SECs with a polynomial number of constraints
- Does the resulting model have an equivalent LP relaxation?

Cutting-plane method

- Since there is an exponential number of SECs, we use a cutting-plane method to solve the LP relaxation
- **Separation problem:** given a solution $\overline{y}$ to the current LP relaxation, we have to find the most violated SEC or determine that no violated SEC exists
- Consider the following optimization problem:

$$\gamma = \min \left\{ \sum_{(i,j) \in A(S, N \setminus S)} \overline{y}_{ij} \mid S \subset N, S \neq \emptyset \right\}$$

- If $\gamma \geq 1$, there is no violated SEC; otherwise, if $\gamma < 1$, the optimal solution $\overline{S}$ corresponds to the most violated SEC

Solving the separation problem

- The equivalent optimization problem can be reformulated as:

$$\gamma = \min_{j=2,3,\ldots,n} \gamma_j$$

$$\gamma_j = \min \left\{ \sum_{(i,j)\in A(S,N\setminus S)} \overline{y}_{ij} \mid \{1,2,\ldots,j-1\} \subset N, j \in N \setminus S \right\}$$

- Computing γ_j corresponds to finding the **minimum cut** between 1 and j with capacities $\overline{y}_{ij}$ on each arc (i,j)
- How would you solve this problem?

Solving the separation problem

- The equivalent optimization problem can be reformulated as:

$$\gamma = \min_{j=2,3,\ldots,n} \gamma_j$$

$$\gamma_j = \min \left\{ \sum_{(i,j)\in A(S,N\setminus S)} \overline{y}_{ij} \mid \{1,2,\ldots,j-1\} \subset N, j \in N \setminus S \right\}$$

- Computing γ_j corresponds to finding the **minimum cut** between 1 and j with capacities $\overline{y}_{ij}$ on each arc (i,j)
- How would you solve this problem?
- Any efficient *maximum flow* algorithm solves this problem in polynomial time

Embedding cutting-plane into branch-and-bound

- **Branch-and-cut**: apply cutting-plane at each node
- **Cut-and-branch**: apply cutting-plane only at the root
- Is this standard cut-and-branch approach enough to identify an optimal solution?

Embedding cutting-plane into branch-and-bound

- **Branch-and-cut**: apply cutting-plane at each node
- **Cut-and-branch**: apply cutting-plane only at the root
- Is this standard cut-and-branch approach enough to identify an optimal solution?
- We also need to verify that there is no violated SEC at each node where an integer solution is found!
- How do we do that? Remember that $\overline{y}$ is now integer!

Embedding cutting-plane into branch-and-bound

- **Branch-and-cut**: apply cutting-plane at each node
- **Cut-and-branch**: apply cutting-plane only at the root
- Is this standard cut-and-branch approach enough to identify an optimal solution?
- We also need to verify that there is no violated SEC at each node where an integer solution is found!
- How do we do that? Remember that $\overline{y}$ is now integer!
- A linear-time graph traversal algorithm to verify the connectivity of the subgraph induced by $\overline{y}$ does the job!

Integer multicommodity flow problem

- 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}^k on arc (i, j) and commodity k
- Capacity u_{ij} on each arc (i, j)
- Problem description: satisfy the demand of each commodity using **only one path per commodity** at minimum cost, while respecting capacity constraints

Arc-based model

- x_{ij}^k: 1, if arc (i, j) is used in the chosen path for commodity k, 0 otherwise

Arc-based model

- x_{ij}^k: 1, if arc (i, j) is used in the chosen path for commodity k, 0 otherwise

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

Arc-based model

- x_{ij}^k: 1, if arc (i, j) is used in the chosen path for commodity k, 0 otherwise

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

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

Arc-based model

- x_{ij}^k: 1, if arc (i,j) is used in the chosen path for commodity k, 0 otherwise

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

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

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

Arc-based model

- x_{ij}^k: 1, if arc (i,j) is used in the chosen path for commodity k, 0 otherwise

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

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

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

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

Lagrangian relaxation of capacity constraints

- Derive the Lagrangian subproblem obtained after relaxing capacity constraints using multipliers $\alpha \geq 0$

Lagrangian relaxation of capacity constraints

- Derive the Lagrangian subproblem obtained after relaxing capacity constraints using multipliers $\alpha \geq 0$

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

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

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

- How do you solve this subproblem? Does it have the integrality property?

Dantzig-Wolfe reformulation

- Now, we know that the Lagrangian dual is equivalent to the LP relaxation
- We also know that the Lagrangian dual can be solved with **Dantzig-Wolfe reformulation** (obtained from the primal interpretation of Lagrangian duality)
- Write down Dantzig-Wolfe reformulation

Dantzig-Wolfe reformulation

- Now, we know that the Lagrangian dual is equivalent to the LP relaxation
- We also know that the Lagrangian dual can be solved with **Dantzig-Wolfe reformulation** (obtained from the primal interpretation of Lagrangian duality)
- Write down Dantzig-Wolfe reformulation
- Using the fact that the Lagrangian subproblem decomposes into $|K|$ shortest path problems, write down an equivalent **disaggregated** Dantzig-Wolfe reformulation

Dantzig-Wolfe reformulation

- Now, we know that the Lagrangian dual is equivalent to the LP relaxation
- We also know that the Lagrangian dual can be solved with **Dantzig-Wolfe reformulation** (obtained from the primal interpretation of Lagrangian duality)
- Write down Dantzig-Wolfe reformulation
- Using the fact that the Lagrangian subproblem decomposes into $|K|$ shortest path problems, write down an equivalent **disaggregated** Dantzig-Wolfe reformulation
- Show that this disaggregated Dantzig-Wolfe reformulation is equivalent to the LP relaxation of the **path-based model**

Path-based model

- $\mathcal{P}^k$: circuit-free paths between $O(k)$ and $D(k)$ for each k
- δ_{ij}^{kp}: 1, if arc (i,j) is on path $p \in \mathcal{P}^k$; 0, otherwise

$$Z = \min \sum_{k \in K} \sum_{p \in \mathcal{P}^k} \left(\sum_{(i,j) \in A} c_{ij}^k \delta_{ij}^{kp} \right) \lambda^{kp}$$

Path-based model

- $\mathcal{P}^k$: circuit-free paths between $O(k)$ and $D(k)$ for each k
- δ_{ij}^{kp}: 1, if arc (i,j) is on path $p \in \mathcal{P}^k$; 0, otherwise

$$Z = \min \sum_{k \in K} \sum_{p \in \mathcal{P}^k} \left(\sum_{(i,j) \in A} c_{ij}^k \delta_{ij}^{kp} \right) \lambda^{kp}$$

$$\sum_{p \in \mathcal{P}^k} \lambda^{kp} = 1 \quad k \in K \quad (\theta^k)$$

Path-based model

- $\mathcal{P}^k$: circuit-free paths between $O(k)$ and $D(k)$ for each k
- δ_{ij}^{kp}: 1, if arc (i,j) is on path $p \in \mathcal{P}^k$; 0, otherwise

$$Z = \min \sum_{k \in K} \sum_{p \in \mathcal{P}^k} \left(\sum_{(i,j) \in A} c_{ij}^k \delta_{ij}^{kp} \right) \lambda^{kp}$$

$$\sum_{p \in \mathcal{P}^k} \lambda^{kp} = 1 \quad k \in K \quad (\theta^k)$$

$$\sum_{k \in K} \sum_{p \in \mathcal{P}^k} d^k \delta_{ij}^{kp} \lambda^{kp} \leq u_{ij} \quad (i,j) \in A \quad (\alpha_{ij})$$

Path-based model

- $\mathcal{P}^k$: circuit-free paths between $O(k)$ and $D(k)$ for each k
- δ_{ij}^{kp}: 1, if arc (i,j) is on path $p \in \mathcal{P}^k$; 0, otherwise

$$Z = \min \sum_{k \in K} \sum_{p \in \mathcal{P}^k} \left(\sum_{(i,j) \in A} c_{ij}^k \delta_{ij}^{kp} \right) \lambda^{kp}$$

$$\sum_{p \in \mathcal{P}^k} \lambda^{kp} = 1 \quad k \in K \quad (\theta^k)$$

$$\sum_{k \in K} \sum_{p \in \mathcal{P}^k} d^k \delta_{ij}^{kp} \lambda^{kp} \leq u_{ij} \quad (i,j) \in A \quad (\alpha_{ij})$$

$$\lambda^{kp} \in \{0,1\} \quad k \in K,\ p \in \mathcal{P}^k$$

Column generation method

- The LP relaxation of the path-based model has an exponential number of variables: what to do?

Column generation method

- The LP relaxation of the path-based model has an exponential number of variables: what to do?
- Write down the expression of the reduced cost of λ^{kp}

Column generation method

- The LP relaxation of the path-based model has an exponential number of variables: what to do?
- Write down the expression of the reduced cost of λ^{kp}
- The reduced cost of variable λ^{kp} is:

$$\sum_{(i,j)\in A} (c_{ij}^k + \alpha_{ij} d^k)\delta_{ij}^{kp} - \theta^k$$

- What is the pricing problem?

Column generation method

- The LP relaxation of the path-based model has an exponential number of variables: what to do?
- Write down the expression of the reduced cost of λ^{kp}
- The reduced cost of variable λ^{kp} is:

$$\sum_{(i,j)\in A} (c_{ij}^k + \alpha_{ij} d^k)\delta_{ij}^{kp} - \theta^k$$

- What is the pricing problem?
- Solve the Lagrangian subproblem for each k to obtain an optimal solution that corresponds to a path $\overline{p} \in \mathcal{P}^k$
- If $\sum_{(i,j)\in A}(c_{ij}^k + \alpha_{ij} d^k)\delta_{ij}^{k\overline{p}} - \theta^k < 0$, the variable $\lambda^{k\overline{p}}$ corresponding to the optimal path $\overline{p} \in \mathcal{P}^k$ is added
- If $\sum_{(i,j)\in A}(c_{ij}^k + \alpha_{ij} d^k)\delta_{ij}^{k\overline{p}} - \theta^k \geq 0$ for each k, the column generation method has converged to the optimal LP relaxation

Branch-and-price algorithm

- At each node, we perform the column generation method
- What happens in the pricing problem if we branch on λ^{kp}?

Branch-and-price algorithm

- At each node, we perform the column generation method
- What happens in the pricing problem if we branch on λ^{kp}?
- When we branch, we have to make sure that we do not destroy the structure of the pricing problem!
- A better way to branch is as follows: choose k for which there are at least two paths $p(q)$ such that $\overline{\lambda}^{kp(q)} > 0, q = 1, 2$
- Follow the flow from the origin $O(k)$ up to the first node l where paths $p(1)$ and $p(2)$ differ; let $(l, j(q))$ be the arc originating at l and belonging to path $p(q)$, $q = 1, 2$
- Let $(l, j(q)) \in N_l^+(q), q = 1, 2$, where $N_l^+ = N_l^+(1) \cup N_l^+(2)$, $N_l^+(1) \cap N_l^+(2) = \emptyset$; generate the two child nodes defined by the branching constraints:
  1) $\sum_{j \in N_l^+(1)} x_{lj}^k = 0$
  2) $\sum_{j \in N_l^+(2)} x_{lj}^k = 0$
- Show that this branching rule is valid and does not change the way we solve the pricing problem

Budget 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 $k \in K$
- Transportation cost c_{ij}^k on arc (i, j) and commodity k
- Fixed charge f_{ij} incurred whenever arc (i, j) is used to transport some commodity units
- Total budget $B > 0$ on the fixed charges
- Problem description: satisfy the demand of each commodity at minimum cost, while respecting the fixed charge budget

Problem formulation

- x_{ij}^k: fraction of the demand d^k carried on arc (i, j)
- y_{ij}: 1, if arc (i, j) is used; 0 otherwise

Problem formulation

- x_{ij}^k: fraction of the demand d^k carried on arc (i, j)
- y_{ij}: 1, if arc (i, j) is used; 0 otherwise

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

Problem formulation

- x_{ij}^k: fraction of the demand d^k carried on arc (i,j)
- y_{ij}: 1, if arc (i,j) is used; 0 otherwise

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

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

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

Problem formulation

- x_{ij}^k: fraction of the demand d^k carried on arc (i,j)
- y_{ij}: 1, if arc (i,j) is used; 0 otherwise

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

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

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

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

Problem formulation

- x_{ij}^k: fraction of the demand d^k carried on arc (i,j)
- y_{ij}: 1, if arc (i,j) is used; 0 otherwise

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

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

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

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

$$\sum_{(i,j)\in A} f_{ij} y_{ij} \leq B$$

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

Lagrangian relaxation of linking constraints

- Derive the Lagrangian subproblem obtained after relaxing linking constraints using multipliers $\beta \geq 0$

Lagrangian relaxation of linking constraints

- Derive the Lagrangian subproblem obtained after relaxing linking constraints using multipliers $\beta \geq 0$

$$Z(LR(\beta)) = \min \sum_{(i,j)\in A} \sum_{k \in K} (c_{ij}^k + \beta_{ij}^k) x_{ij}^k + \sum_{(i,j)\in A} \left(\sum_{k\in K} -\beta_{ij}^k \right) y_{ij}$$

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

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

$$\sum_{(i,j)\in A} f_{ij} y_{ij} \leq B$$

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

- How do you solve this subproblem? Does it have the integrality property?

Partial Dantzig-Wolfe reformulation of Lagrangian dual

- Q: extreme points of $conv\{y \in \{0,1\}^{|A|} \mid \sum_{(i,j)\in A} f_{ij} y_{ij} \leq B\}$
- δ_{ij}^{q}: 1, if arc (i,j) is chosen in extreme point q; 0, otherwise

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

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

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

Partial Dantzig-Wolfe reformulation of Lagrangian dual

- Q: extreme points of $conv\{y \in \{0,1\}^{|A|} \mid \sum_{(i,j)\in A} f_{ij} y_{ij} \leq B\}$
- δ_{ij}^q: 1, if arc (i,j) is chosen in extreme point q; 0, otherwise

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

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

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

$$x_{ij}^k \leq \sum_{q\in Q} \delta_{ij}^q \lambda^q \quad (i,j) \in A, k \in K \quad (\beta_{ij}^k)$$

Partial Dantzig-Wolfe reformulation of Lagrangian dual

- Q: extreme points of $conv\{y \in \{0,1\}^{|A|} \mid \sum_{(i,j)\in A} f_{ij} y_{ij} \leq B\}$
- δ_{ij}^q: 1, if arc (i,j) is chosen in extreme point q; 0, otherwise

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

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

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

$$x_{ij}^k \leq \sum_{q\in Q} \delta_{ij}^q \lambda^q \quad (i,j) \in A, k \in K \quad (\beta_{ij}^k)$$

$$\sum_{q\in Q} \lambda^q = 1 \quad (\theta)$$

$$\lambda^q \geq 0 \quad q \in Q$$

Solving the Lagrangian dual

- Exponential number of λ variables: column generation!
- The number of x variables/linking constraints is polynomial, but can be extremely large: column-and-row generation!

Solving the Lagrangian dual

- Exponential number of λ variables: column generation!
- The number of x variables/linking constraints is polynomial, but can be extremely large: column-and-row generation!
- Reduced cost of each λ^q:

$$\sum_{(i,j)\in A} \delta_{ij}^q \left(\sum_{k\in K} -\beta_{ij}^k \right) - \theta$$

Solving the Lagrangian dual

- Exponential number of λ variables: column generation!
- The number of x variables/linking constraints is polynomial, but can be extremely large: column-and-row generation!
- Reduced cost of each λ^q:

$$\sum_{(i,j)\in A} \delta_{ij}^q \left(\sum_{k\in K} -\beta_{ij}^k \right) - \theta$$

- **Problem**: since we use column-and-row generation for x variables/linking constraints, many β_{ij}^k values are unknown!

Solving the Lagrangian dual

- Exponential number of λ variables: column generation!
- The number of x variables/linking constraints is polynomial, but can be extremely large: column-and-row generation!
- Reduced cost of each λ^q:

$$\sum_{(i,j)\in A} \delta_{ij}^q \left(\sum_{k\in K} -\beta_{ij}^k \right) - \theta$$

- **Problem**: since we use column-and-row generation for x variables/linking constraints, many β_{ij}^k values are unknown!
- **Solution**: use LP duality to derive the "missing" β_{ij}^k values

Dual of partial Dantzig-Wolfe reformulation

$$\max \sum_{k \in K} \left(\pi^k_{D(k)} - \pi^k_{O(k)} \right) + \theta$$

Dual of partial Dantzig-Wolfe reformulation

$$\max \sum_{k \in K} \left(\pi^k_{D(k)} - \pi^k_{O(k)} \right) + \theta$$

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

Dual of partial Dantzig-Wolfe reformulation

$$\max \sum_{k \in K} \left(\pi^k_{D(k)} - \pi^k_{O(k)} \right) + \theta$$

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

$$\sum_{(ij) \in A} \sum_{k \in K} \delta^q_{ij} \beta^k_{ij} + \theta \leq 0 \quad q \in Q$$

Dual of partial Dantzig-Wolfe reformulation

$$\max \sum_{k \in K} \left(\pi^k_{D(k)} - \pi^k_{O(k)} \right) + \theta$$

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

$$\sum_{(ij) \in A} \sum_{k \in K} \delta^q_{ij} \beta^k_{ij} + \theta \leq 0 \quad q \in Q$$

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

Dual of partial Dantzig-Wolfe reformulation

$$\max \sum_{k \in K} \left(\pi^k_{D(k)} - \pi^k_{O(k)} \right) + \theta$$

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

$$\sum_{(ij) \in A} \sum_{k \in K} \delta^q_{ij} \beta^k_{ij} + \theta \leq 0 \quad q \in Q$$

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

- Any optimal solution satisfies $\theta = -\sum_{(i,j) \in A} \sum_{k \in K} \delta^q_{ij} \beta^k_{ij}$

Dual of partial Dantzig-Wolfe reformulation

$$\max \sum_{k \in K} \left(\pi^k_{D(k)} - \pi^k_{O(k)} \right) + \theta$$

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

$$\sum_{(ij) \in A} \sum_{k \in K} \delta^q_{ij} \beta^k_{ij} + \theta \leq 0 \quad q \in Q$$

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

- Any optimal solution satisfies $\theta = -\sum_{(i,j) \in A} \sum_{k \in K} \delta^q_{ij} \beta^k_{ij}$
- Therefore, any optimal solution also satisfies

$$-\beta^k_{ij} = \min\{0, c^k_{ij} + \pi^k_i - \pi^k_j\} \quad (i,j) \in A, k \in K$$

Pricing problem

- Finding variable λ^q with the smallest reduced cost is thus equivalent to

$$\min_{q \in Q} \left\{ \sum_{(i,j) \in A} \delta_{ij}^q \left(\sum_{k \in K} \min\{0, c_{ij}^k + \pi_i^k - \pi_j^k\} \right) \right\} - \theta$$

Pricing problem

- Finding variable λ^q with the smallest reduced cost is thus equivalent to

$$\min_{q \in Q} \left\{ \sum_{(i,j) \in A} \delta_{ij}^q \left(\sum_{k \in K} \min\{0, c_{ij}^k + \pi_i^k - \pi_j^k\} \right) \right\} - \theta$$

- Therefore, it is obtained by solving the Lagrangian subproblem

$$\min \sum_{(i,j) \in A} \left(\sum_{k \in K} \min\{0, c_{ij}^k + \pi_i^k - \pi_j^k\} \right) y_{ij}$$

$$\sum_{(i,j) \in A} f_{ij} y_{ij} \leq B$$

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

Column-and-row generation method

- $\overline{K}_{ij}$: commodities corresponding to the x variables that are NOT in the current restricted master problem
- $\overline{y}$, $\widetilde{y}$: optimal values of design variables from current restricted master and Lagrangian subproblem ($\overline{y}_{ij} = \sum_{q \in Q} \delta_{ij}^q \overline{\lambda}^q$)
- Generation of x variables, for each (i, j):

Column-and-row generation method

- $\overline{K}_{ij}$: commodities corresponding to the x variables that are NOT in the current restricted master problem
- $\overline{y}$, $\widetilde{y}$: optimal values of design variables from current restricted master and Lagrangian subproblem ($\overline{y}_{ij} = \sum_{q \in Q} \delta_{ij}^q \overline{\lambda}^q$)
- Generation of x variables, for each (i, j):
  - If $\overline{y}_{ij} > 0$, generate $k \in \overline{K}_{ij}$ such that $c_{ij}^k + \pi_i^k - \pi_j^k < 0$
  - If $\overline{y}_{ij} = 0$ and $\widetilde{y}_{ij} = 1$, generate $k \in \overline{K}_{ij}$ such that $c_{ij}^k + \pi_i^k - \pi_j^k < 0$
  - If $\overline{y}_{ij} = 0$ and $\widetilde{y}_{ij} = 0$, do not generate any new variables related to (i, j)

Column-and-row generation method

- $\overline{K}_{ij}$: commodities corresponding to the x variables that are NOT in the current restricted master problem
- $\overline{y}$, $\widetilde{y}$: optimal values of design variables from current restricted master and Lagrangian subproblem ($\overline{y}_{ij} = \sum_{q \in Q} \delta_{ij}^q \overline{\lambda}^q$)
- Generation of x variables, for each (i, j):
  - If $\overline{y}_{ij} > 0$, generate $k \in \overline{K}_{ij}$ such that $c_{ij}^k + \pi_i^k - \pi_j^k < 0$
  - If $\overline{y}_{ij} = 0$ and $\widetilde{y}_{ij} = 1$, generate $k \in \overline{K}_{ij}$ such that $c_{ij}^k + \pi_i^k - \pi_j^k < 0$
  - If $\overline{y}_{ij} = 0$ and $\widetilde{y}_{ij} = 0$, do not generate any new variables related to (i, j)
- After column generation, add linking constraints as cuts

Column-and-row generation method

- $\overline{K}_{ij}$: commodities corresponding to the x variables that are NOT in the current restricted master problem
- $\overline{y}$, $\widetilde{y}$: optimal values of design variables from current restricted master and Lagrangian subproblem ($\overline{y}_{ij} = \sum_{q \in Q} \delta_{ij}^{q} \overline{\lambda}^{q}$)
- Generation of x variables, for each (i, j):
  - If $\overline{y}_{ij} > 0$, generate $k \in \overline{K}_{ij}$ such that $c_{ij}^{k} + \pi_{i}^{k} - \pi_{j}^{k} < 0$
  - If $\overline{y}_{ij} = 0$ and $\widetilde{y}_{ij} = 1$, generate $k \in \overline{K}_{ij}$ such that $c_{ij}^{k} + \pi_{i}^{k} - \pi_{j}^{k} < 0$
  - If $\overline{y}_{ij} = 0$ and $\widetilde{y}_{ij} = 0$, do not generate any new variables related to (i, j)
- After column generation, add linking constraints as cuts
- Subgradient optimization to initialize the master problem
- With n best solutions from subgradient optimization:
- Generate subset of extreme points $\overline{Q} \subseteq Q$
- If $\delta_{ij}^{q} = 1$ for at least one $q \in \overline{Q}$ and ($x_{ij}^{k} > 0$ or $\beta_{ij}^{k} > 0$), generate x_{ij}^{k} and corresponding linking constraint

Column-and-row generation method

- $\overline{K}_{ij}$: commodities corresponding to the x variables that are NOT in the current restricted master problem
- $\overline{y}$, $\widetilde{y}$: optimal values of design variables from current restricted master and Lagrangian subproblem ($\overline{y}_{ij} = \sum_{q \in Q} \delta_{ij}^q \overline{\lambda}^q$)
- Generation of x variables, for each (i, j):
  - If $\overline{y}_{ij} > 0$, generate $k \in \overline{K}_{ij}$ such that $c_{ij}^k + \pi_i^k - \pi_j^k < 0$
  - If $\overline{y}_{ij} = 0$ and $\widetilde{y}_{ij} = 1$, generate $k \in \overline{K}_{ij}$ such that $c_{ij}^k + \pi_i^k - \pi_j^k < 0$
  - If $\overline{y}_{ij} = 0$ and $\widetilde{y}_{ij} = 0$, do not generate any new variables related to (i, j)
- After column generation, add linking constraints as cuts
- Subgradient optimization to initialize the master problem
- With n best solutions from subgradient optimization:
- Generate subset of extreme points $\overline{Q} \subseteq Q$
- If $\delta_{ij}^q = 1$ for at least one $q \in \overline{Q}$ and ($x_{ij}^k > 0$ or $\beta_{ij}^k > 0$), generate x_{ij}^k and corresponding linking constraint
- Do not forget artificial flow variables, one per commodity!

Branch-and-price-and-cut

- Apply column-and-row generation at every node
- How to branch?

Branch-and-price-and-cut

- Apply column-and-row generation at every node
- How to branch?
- Design variables are related to λ^q variables as follows:

$$y_{ij} = \sum_{q \in Q} \delta_{ij}^q \lambda^q \quad (i,j) \in A$$

Branch-and-price-and-cut

- Apply column-and-row generation at every node
- How to branch?
- Design variables are related to λ^q variables as follows:

$$y_{ij} = \sum_{q \in Q} \delta_{ij}^q \lambda^q \quad (i,j) \in A$$

- If $\sum_{q \in Q} \delta_{ij}^q \lambda^q \in \{0,1\}$, $(i,j) \in A$, the node can be fathomed
- Otherwise, select an arc (i,j) such that $\sum_{q \in Q} \delta_{ij}^q \lambda^q$ is fractional and generate the two child nodes defined by:
- $\sum_{q \in Q} \delta_{ij}^q \lambda^q = 0$
- $\sum_{q \in Q} \delta_{ij}^q \lambda^q = 1$
- What happens in the Lagrangian subproblem?

Branch-and-price-and-cut

- Apply column-and-row generation at every node
- How to branch?
- Design variables are related to λ^q variables as follows:

$$y_{ij} = \sum_{q \in Q} \delta_{ij}^q \lambda^q \quad (i,j) \in A$$

- If $\sum_{q \in Q} \delta_{ij}^q \lambda^q \in \{0,1\}$, $(i,j) \in A$, the node can be fathomed
- Otherwise, select an arc (i,j) such that $\sum_{q \in Q} \delta_{ij}^q \lambda^q$ is fractional and generate the two child nodes defined by:
- $\sum_{q \in Q} \delta_{ij}^q \lambda^q = 0$
- $\sum_{q \in Q} \delta_{ij}^q \lambda^q = 1$
- What happens in the Lagrangian subproblem?
- Constraint $y_{ij} = 0$ or $y_{ij} = 1$ is added without any problem!

Uncapacitated facility location problem (UFLP)

- K: set of customers
- J: set of locations for potential facilities
- $f_j \geq 0$: fixed cost for opening facility at location j
- $c_{jk} \geq 0$: cost of satisfying the demand of customer k from facility at location j
- Problem description: determine the locations of the facilities to satisfy customers' demands at minimum cost

Problem formulation

$$\min \sum_{j \in J} \sum_{k \in K} c_{jk} x_{jk} + \sum_{j \in J} f_j y_j$$

$$\sum_{j \in J} x_{jk} = 1, \quad k \in K$$

$$x_{jk} \leq y_j, \quad j \in J, k \in K$$

$$x_{jk} \geq 0, \quad j \in J, k \in K$$

$$y_j \in \{0, 1\}, \quad j \in J$$

Benders subproblem

$$\min \sum_{j \in J} \sum_{k \in K} c_{jk} x_{jk}$$

$$\sum_{j \in J} x_{jk} = 1, \quad k \in K \quad (\pi_k)$$

$$x_{jk} \leq \overline{y}_j, \quad j \in J, k \in K \quad (\alpha_{jk} \geq 0)$$

$$x_{jk} \geq 0, \quad j \in J, k \in K$$

- How to solve Benders subproblem?

Benders subproblem

$$\min \sum_{j \in J} \sum_{k \in K} c_{jk} x_{jk}$$

$$\sum_{j \in J} x_{jk} = 1, \quad k \in K \quad (\pi_k)$$

$$x_{jk} \leq \overline{y}_j, \quad j \in J, k \in K \quad (\alpha_{jk} \geq 0)$$

$$x_{jk} \geq 0, \quad j \in J, k \in K$$

- How to solve Benders subproblem?
- Decomposes for each k: assign the cheapest open location
- Give a simple condition to ensure feasibility

Benders subproblem

$$\min \sum_{j \in J} \sum_{k \in K} c_{jk} x_{jk}$$

$$\sum_{j \in J} x_{jk} = 1, \quad k \in K \quad (\pi_k)$$

$$x_{jk} \leq \overline{y}_j, \quad j \in J, k \in K \quad (\alpha_{jk} \geq 0)$$

$$x_{jk} \geq 0, \quad j \in J, k \in K$$

- How to solve Benders subproblem?
- Decomposes for each k: assign the cheapest open location
- Give a simple condition to ensure feasibility
- Feasibility requires that at least one location is open!

Dual of decomposed Benders subproblem

$$Z_k(\overline{y}) = \max\{\pi_k - \sum_{j \in J} \alpha_{jk}\overline{y}_j \,|\, \pi_k - \alpha_{jk} \leq c_{jk}, \alpha_{jk} \geq 0, j \in J\}$$

Dual of decomposed Benders subproblem

$$Z_k(\overline{y}) = \max\{\pi_k - \sum_{j \in J} \alpha_{jk} \overline{y}_j \mid \pi_k - \alpha_{jk} \leq c_{jk}, \alpha_{jk} \geq 0, j \in J\}$$

- Every optimal solution satisfies $\pi_k = \min_{j \in J}\{c_{jk} + \alpha_{jk}\}$ and $\alpha_{jk} = \max\{0, \pi_k - c_{jk}\} = (\pi_k - c_{jk})^+$

Dual of decomposed Benders subproblem

$$Z_k(\overline{y}) = \max\{\pi_k - \sum_{j \in J} \alpha_{jk}\overline{y}_j \,|\, \pi_k - \alpha_{jk} \leq c_{jk}, \alpha_{jk} \geq 0, j \in J\}$$

- Every optimal solution satisfies $\pi_k = \min_{j \in J}\{c_{jk} + \alpha_{jk}\}$ and $\alpha_{jk} = \max\{0, \pi_k - c_{jk}\} = (\pi_k - c_{jk})^+$
- The last equation implies there exists $l \in J$ such that $\pi_k = c_{lk}$
- Make sure you understand why!

Dual of decomposed Benders subproblem

$$Z_k(\overline{y}) = \max\{\pi_k - \sum_{j \in J} \alpha_{jk}\overline{y}_j \,|\, \pi_k - \alpha_{jk} \leq c_{jk}, \alpha_{jk} \geq 0, j \in J\}$$

- Every optimal solution satisfies $\pi_k = \min_{j \in J}\{c_{jk} + \alpha_{jk}\}$ and $\alpha_{jk} = \max\{0, \pi_k - c_{jk}\} = (\pi_k - c_{jk})^+$
- The last equation implies there exists $l \in J$ such that $\pi_k = c_{lk}$
- Make sure you understand why!
- The decomposed dual can therefore be reformulated as:

$$Z_k(\overline{y}) = \max_{l \in J} \left\{ c_{lk} - \sum_{j \in J} (c_{lk} - c_{jk})^+ \overline{y}_j \right\}$$

Benders reformulation

- Feasibility is ensured if at least one location is open: $\sum_{j \in J} y_j \geq 1$ is the only needed Benders feasibility cut!
- What is the link with extreme ray of the dual polyhedron?

Benders reformulation

- Feasibility is ensured if at least one location is open: $\sum_{j \in J} y_j \geq 1$ is the only needed Benders feasibility cut!
- What is the link with extreme ray of the dual polyhedron?
- $\pi_k = \alpha_{jk} = 1$, $j \in J$, defines an extreme ray for which $\pi_k - \sum_{j \in J} \alpha_{jk} \overline{y}_j > 0$ if all locations are closed!

Benders reformulation

- Feasibility is ensured if at least one location is open: $\sum_{j \in J} y_j \geq 1$ is the only needed Benders feasibility cut!
- What is the link with extreme ray of the dual polyhedron?
- $\pi_k = \alpha_{jk} = 1$, $j \in J$, defines an extreme ray for which $\pi_k - \sum_{j \in J} \alpha_{jk} \bar{y}_j > 0$ if all locations are closed!
- Using our simplified reformulation of the decomposed dual, the Benders reformulation is:

$$\min \sum_{j \in J} f_j y_j + \sum_{k \in K} z_k$$

Benders reformulation

- Feasibility is ensured if at least one location is open: $\sum_{j \in J} y_j \geq 1$ is the only needed Benders feasibility cut!
- What is the link with extreme ray of the dual polyhedron?
- $\pi_k = \alpha_{jk} = 1$, $j \in J$, defines an extreme ray for which $\pi_k - \sum_{j \in J} \alpha_{jk} \bar{y}_j > 0$ if all locations are closed!
- Using our simplified reformulation of the decomposed dual, the Benders reformulation is:

$$\min \sum_{j \in J} f_j y_j + \sum_{k \in K} z_k$$

$$c_{lk} - \sum_{j \in J} (c_{lk} - c_{jk})^+ y_j \leq z_k \quad k \in K, l \in J$$

Benders reformulation

- Feasibility is ensured if at least one location is open: $\sum_{j \in J} y_j \geq 1$ is the only needed Benders feasibility cut!
- What is the link with extreme ray of the dual polyhedron?
- $\pi_k = \alpha_{jk} = 1$, $j \in J$, defines an extreme ray for which $\pi_k - \sum_{j \in J} \alpha_{jk} \bar{y}_j > 0$ if all locations are closed!
- Using our simplified reformulation of the decomposed dual, the Benders reformulation is:

$$\min \sum_{j \in J} f_j y_j + \sum_{k \in K} z_k$$

$$c_{lk} - \sum_{j \in J} (c_{lk} - c_{jk})^+ y_j \leq z_k \quad k \in K, l \in J$$

$$1 - \sum_{j \in J} y_j \leq 0$$

Benders reformulation

- Feasibility is ensured if at least one location is open: $\sum_{j \in J} y_j \geq 1$ is the only needed Benders feasibility cut!
- What is the link with extreme ray of the dual polyhedron?
- $\pi_k = \alpha_{jk} = 1$, $j \in J$, defines an extreme ray for which $\pi_k - \sum_{j \in J} \alpha_{jk} \bar{y}_j > 0$ if all locations are closed!
- Using our simplified reformulation of the decomposed dual, the Benders reformulation is:

$$\min \sum_{j \in J} f_j y_j + \sum_{k \in K} z_k$$

$$c_{lk} - \sum_{j \in J} (c_{lk} - c_{jk})^+ y_j \leq z_k \quad k \in K, l \in J$$

$$1 - \sum_{j \in J} y_j \leq 0$$

$$y_j \in \{0, 1\}, \quad j \in J$$

Benders decomposition

- Initial Benders master problem: add $\sum_{j \in J} y_j \geq 1$ + one constraint for each k corresponding to the cheapest location
- At each iteration, we solve the Benders master problem to get $\bar{y}$ and the Benders subproblem for that $\bar{y}$
- The solution to the Benders subproblem for each k is $x_{j^*k} = 1$ for the cheapest open location j^*, otherwise $x_{jk} = 0$, $j \neq j^*$
- The solution to the dual of the Benders subproblem for each k is $\pi_k = c_{j^*k}$ (because of strong duality!)

Benders decomposition

- Initial Benders master problem: add $\sum_{j \in J} y_j \geq 1$ + one constraint for each k corresponding to the cheapest location
- At each iteration, we solve the Benders master problem to get $\overline{y}$ and the Benders subproblem for that $\overline{y}$
- The solution to the Benders subproblem for each k is $x_{j^*k} = 1$ for the cheapest open location j^*, otherwise $x_{jk} = 0$, $j \neq j^*$
- The solution to the dual of the Benders subproblem for each k is $\pi_k = c_{j^*k}$ (because of strong duality!)
- A Benders optimality cut is added for each k such that

$$c_{j^*k} - \sum_{j \in J} (c_{j^*k} - c_{jk})^+ \overline{y}_j > z_k$$