Decomposition and Dynamic Cut Generation in Integer Programming

Ted K. Ralphs
Matthew V. Galati

Department of Industrial and Systems Engineering
Lehigh University, Bethlehem, PA

http://www.lehigh.edu/~tkr2

8^{th} International Workshop on Combinatorial Optimization, January 5-9, 2004, Aussois, France

Outline

- Preliminaries, Traditional Decomposition Methods
  - Dantzig-Wolfe Decomposition
  - Lagrangian Relaxation
  - Cutting Plane Method
- Dynamic Decomposition Methods
  - Price and Cut
  - Relax and Cut
  - Decompose and Cut
- Applications/Examples
- DECOMP Framework

Preliminaries

- We consider the following pure integer linear program:

$$z_{IP} = \min_{x \in \mathcal{F}} \{c^\top x\} = \min_{x \in \mathcal{P}} \{c^\top x\}$$

where

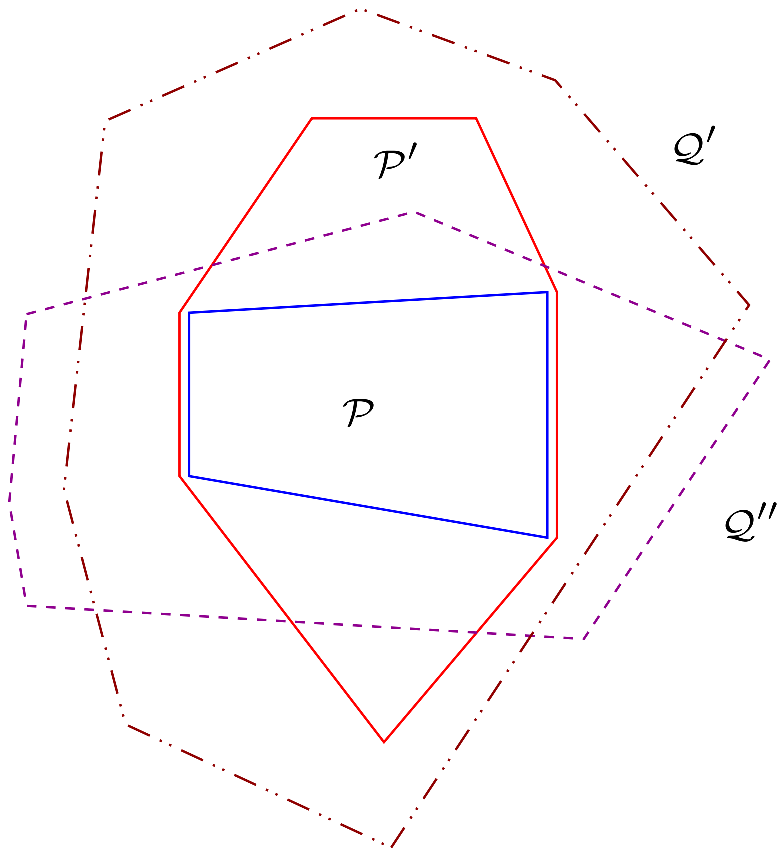
$$\mathcal{F} = \{x \in \mathbb{Z}^n : A'x \geq b', A''x \geq b''\} \qquad \mathcal{Q} = \{x \in \mathbb{R}^n : A'x \geq b', A''x \geq b''\}$$

$$\mathcal{F}' = \{x \in \mathbb{Z}^n : A'x \geq b'\} \qquad \mathcal{Q}' = \{x \in \mathbb{R}^n : A'x \geq b'\}$$

$$\mathcal{Q}'' = \{x \in \mathbb{R}^n : A''x \geq b''\}$$

- We will consider $\mathcal{P} = conv(\mathcal{F})$ and $\mathcal{P}' = conv(\mathcal{F}')$.
- Assumptions
  - All input data are rational.
  - $\mathcal{P}$ is bounded.
  - Optimization/separation over $\mathcal{P}$ is "difficult."
  - Optimization/separation over $\mathcal{P}'$ is "easy."

Polyhedra



$\mathcal{P} = conv(\{x \in \mathbb{Z}^n : Ax \geq b\})$

$\mathcal{P}' = conv(\{x \in \mathbb{Z}^n : A'x \geq b'\})$

$\mathcal{Q}' = \{x \in \mathbb{R}^n : A'x \geq b'\}$

$\mathcal{Q}'' = \{x \in \mathbb{R}^n : A''x \geq b''\}$

Bounding

- Goal: Compute a lower bound on z_{IP} by solving a *bounding problem*.
- The most commonly used bounding problem is the initial LP relaxation.

$$\min_{x \in \mathcal{Q}} \{c^\top x\}$$

- Decomposition approaches attempt to improve on this bound by utilizing implicit knowledge of $\mathcal{P}'$.
  - We have an explicit description of $\mathcal{Q}''$.
  - $\mathcal{P}'$ is represented *implicitly* through solution of a subproblem.
- Decomposition methods
  - Dantzig-Wolfe decomposition
  - Lagrangian relaxation
  - Cutting plane method

Dantzig-Wolfe Decomposition (DW)

- The bounding problem is the *Dantzig-Wolfe LP*:

$$z_{DW} = \min\{c(\sum_{s \in \mathcal{F}'} s\lambda_s) : A''(\sum_{s \in \mathcal{F}'} s\lambda_s) \geq b'', \sum_{s \in \mathcal{F}'} \lambda_s = 1, \lambda_s \geq 0 \ \forall s \in \mathcal{F}'\} \quad (1)$$

- Solution method: Simplex algorithm with dynamic column generation.
- Subproblem: Optimization over $\mathcal{P}'$.
- Let $\hat{\lambda}$ be an optimal solution to (1) (the *optimal decomposition*) and

$$\hat{x} = \sum_{s \in \mathcal{F}'} s\hat{\lambda}_s \in \mathcal{P}' \cap \mathcal{Q}'' \quad (2)$$

Then, $z_{IP} \geq z_{DW} = c^{\top}\hat{x} \geq z_{LP}$.

Lagrangian Relaxation (LD)

- The bounding problem is the *Lagrangian dual*:

$$z_{LR}(u) = \min_{s \in \mathcal{F}'} \{(c^\top - u^\top A'')s + u^\top b''\} \tag{3}$$

$$z_{LD} = \max_{u \in \mathbb{R}^{m''}_+} \{z_{LR}(u)\} \tag{4}$$

- Solution method: Subgradient optimization.
- Subproblem: Optimization over $\mathcal{P}'$.
- Rewriting (4) as a linear program, we see it is dual to the DW LP.

$$z_{LD} = \max_{\eta \in \mathbb{R}, u \in \mathbb{R}^{m''}_+} \{\eta : \eta \leq (c - uA'')s + ub'' \ \forall s \in \mathcal{F}'\} \tag{5}$$

- So we have $z_{IP} \geq z_{LD} = z_{DW} \geq z_{LP}$.
- We denote by $\hat{u}$ an optimal (dual) solution to (4).

Cutting Plane Method (CP)

- The bounding problem is the initial LP relaxation augmented with facet-defining inequalities from $\mathcal{P}'$:

$$z_{CP} = \min_{x \in \mathcal{P}'} \{cx : A''x \geq b''\} \tag{6}$$

.

- Solution method: Simplex with dynamic cut generation.
- Subproblem: Separation from $\mathcal{P}'$.
- We assumed that separation over $\mathcal{P}'$ was also "easy."
- Note that $\hat{x}$ from (2) is an optimal solution to (6), so $z_{IP} \geq z_{CP} = z_{DW} \geq z_{LP}$.

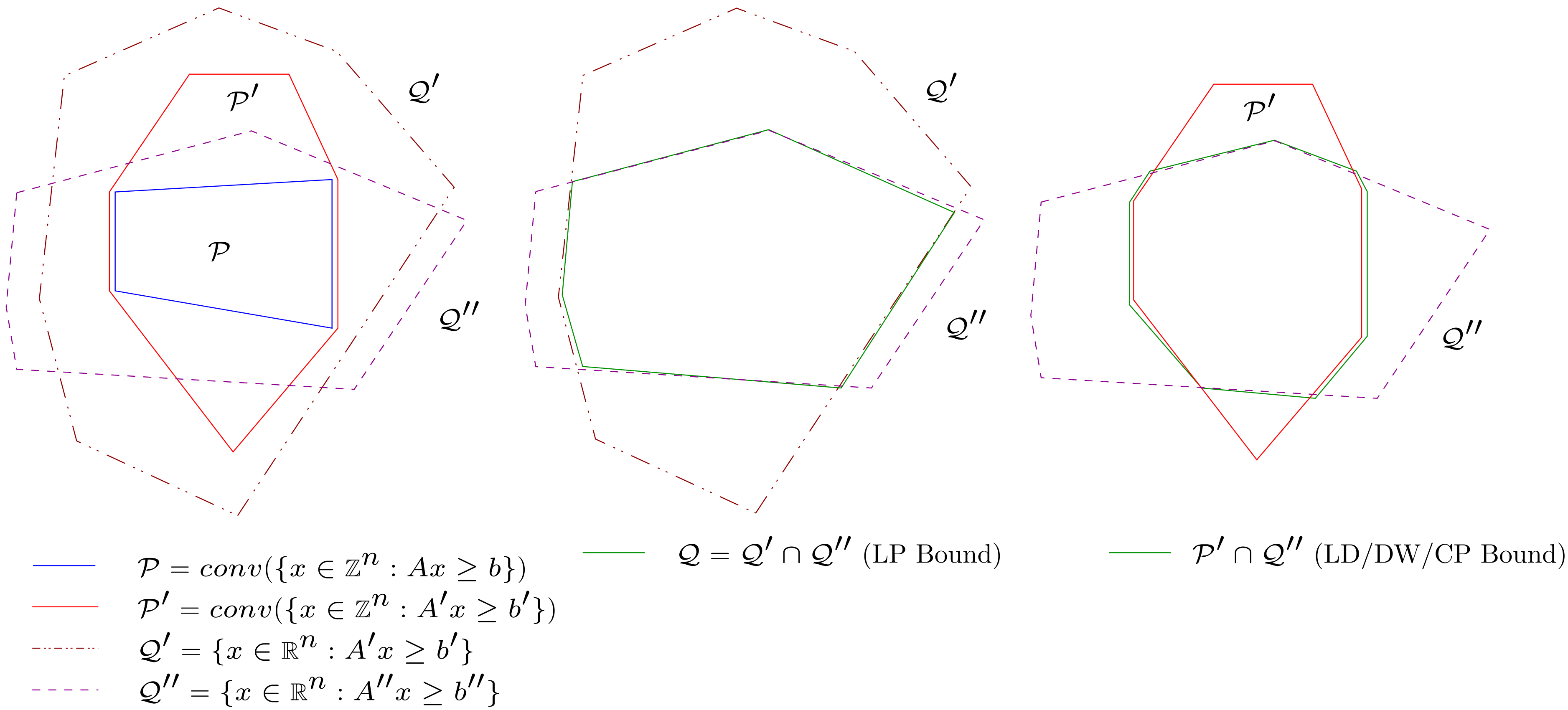
A Common Framework

- The three methods compute the same bound [Geoffrion74].

$$z_{IP} \geq c^\top \hat{x} = z_{LD} = z_{DW} = z_{CP} \geq z_{LP}$$

- The basic ingredients are the same:
  - the *original polyhedron* ($\mathcal{P}$),
  - an *implicit polyhedron* ($\mathcal{P}'$), and
  - an *explicit polyhedron* ($\mathcal{Q}''$).
- The essential difference is how the implicit polyhedron is represented:
  - CP: as the intersection of half-spaces (the *outer representation*), or
  - DW/LD: as the convex hull of a finite set (the *inner representation*).

Polyhedra, LP Bound, LD/DW/CP Bound



Outline

- Preliminaries, Traditional Decomposition Methods
  - Dantzig-Wolfe Decomposition
  - Lagrangian Relaxation
  - Cutting Plane Method
- Dynamic Decomposition Methods
  - Price and Cut
  - Relax and Cut
  - Decompose and Cut
- Applications/Examples
- DECOMP Framework

Improving the Bound

- With traditional methods, we achieve the bound $\min_{x \in \mathcal{P}'}\{cx : A''x \geq b''\}$.
- With the cutting plane method, this bound can be further improved.

Cutting Plane Method

1. Construct the initial bounding problem CP^0 and set $i \leftarrow 0$.

   $$z_{CP} = \min_{x \in \mathcal{P}'}\{c^\top x : A''x \geq b''\}$$

2. Solve CP^i to obtain an optimal solution $\hat{x}^i$ and valid lower bound $z^i = c^\top \hat{x}^i$.
3. Attempt to separate $\hat{x}^i$ from $\mathcal{P}$, generating a set $[D^i, d^i]$ of valid inequalities violated by $\hat{x}^i$.
4. If valid inequalities were found in Step 3, form the augmented bounding problem CP^{i+1} by setting $[A'', b''] \leftarrow \begin{bmatrix} A'' & b'' \\ D^i & d^i \end{bmatrix}$. Then, set $i \leftarrow i + 1$ and go to Step 2.
5. If no valid inequalities were found in Step 3, then output z^i.

Dynamic Decomposition Methods

- The same principle can also be employed with other bounding methods.

Dynamic Decomposition Method

1. Construct the initial bounding problem P^0 and set $i \leftarrow 0$.

$$z_{CP} = \min_{x \in \mathcal{P}'} \{c^\top x : A''x \geq b''\}$$
$$z_{LD} = \max_{u \in \mathbb{R}^n_+} \min_{x \in \mathcal{P}'} \{(c^\top - u^\top A'')x + u^\top b''\}$$
$$z_{DW} = \min_{\lambda \in \mathbb{R}^{\mathcal{F}'}_+} \{c^\top (\textstyle\sum_{s \in \mathcal{F}'} s\lambda_s) : A''(\textstyle\sum_{s \in \mathcal{F}'} s\lambda_s) \geq b'', \textstyle\sum_{s \in \mathcal{F}'} \lambda_s = 1\}$$

2. Solve P^i to obtain a valid lower bound z^i.

3. Try to generate a set of *improving inequalities* $[D^i, d^i]$ valid for $\mathcal{P}$.

4. If valid inequalities were found in Step 3, form the bounding problem P^{i+1} by setting $[A'', b''] \leftarrow \begin{bmatrix} A'' & b'' \\ D^i & d^i \end{bmatrix}$. Then, set $i \leftarrow i + 1$ and go to Step 2.

5. If no valid inequalities were found in Step 3, then output z^i.

Improving Inequalities

- All three procedures employ a similar dynamic tightening of the explicit polyhedron.
- Hence, we call these *dynamic decomposition methods*.
- The challenge is in performing Step 3.
- An inequality found in Step 3 that improves the current bound when added to the bounding problem is an *improving inequality*.
- A necessary and sufficient condition for an inequality to be improving is that it is violated by all optimal primal solutions to (6).
- This condition is difficult to verify.
- In the cutting plane method, we typically use the necessary condition that the generated inequality be violated by $\hat{x}$.
- Questions: What can be done with the other bounding methods?

Price and Cut (PC)

Price and Cut: Use DW as the bounding problem.

$$z_{DW} = \min_{\lambda \in \mathbb{R}_+^{\mathcal{F}'}} \{c^\top (\sum_{s \in \mathcal{F}'} s\lambda_s) : A''(\sum_{s \in \mathcal{F}'} s\lambda_s) \geq b'', \sum_{s \in \mathcal{F}'} \lambda_s = 1\}$$

Attempt to separate $\hat{x} = \sum_{s \in \mathcal{F}'} s\hat{\lambda}_s$.

- Cut generation takes place in original space, maintaining the structure of the column generation subproblem.
- Both PC and CP try to separate $\hat{x}$ from $\mathcal{P}$.
- With PC, however, we get additional information, i.e., the optimal decomposition $\hat{\lambda}$.
- Question: Can we take advantage of this information?

Relax and Cut (RC)

Relax and Cut: Use LD as the bounding problem.

$$z_{LD} = \max_{u \in \mathbb{R}^n_+} \min_{s \in \mathcal{F}'} \{(c^\top - u^\top A'')s + u^\top b''\}$$

Attempt to separate $\hat{s} \in \mathcal{F}'$, a solution to $z_{LR}(\hat{u})$.

- It is often much easier to separate a member of $\mathcal{F}'$ from $\mathcal{P}$ than an arbitrary real vector, such as $\hat{x}$.
- However, there is no way to know whether the generated inequalities are improving or are violated by $\hat{x}$.
- Questions:
  - Can we improve our chances of generating an improving inequality?
  - Can we characterize the relationship between $\hat{s}$ and $\hat{x}$?

Improving Inequalities (cont.)

- The set of alternative optimal solutions to $z_{LR}(\hat{u})$ is

$$\mathcal{S} = \{s \in \mathcal{F}' : (c^\top - \hat{u}^\top A'')s = (c^\top - \hat{u}^\top A'')\hat{s}\}.$$

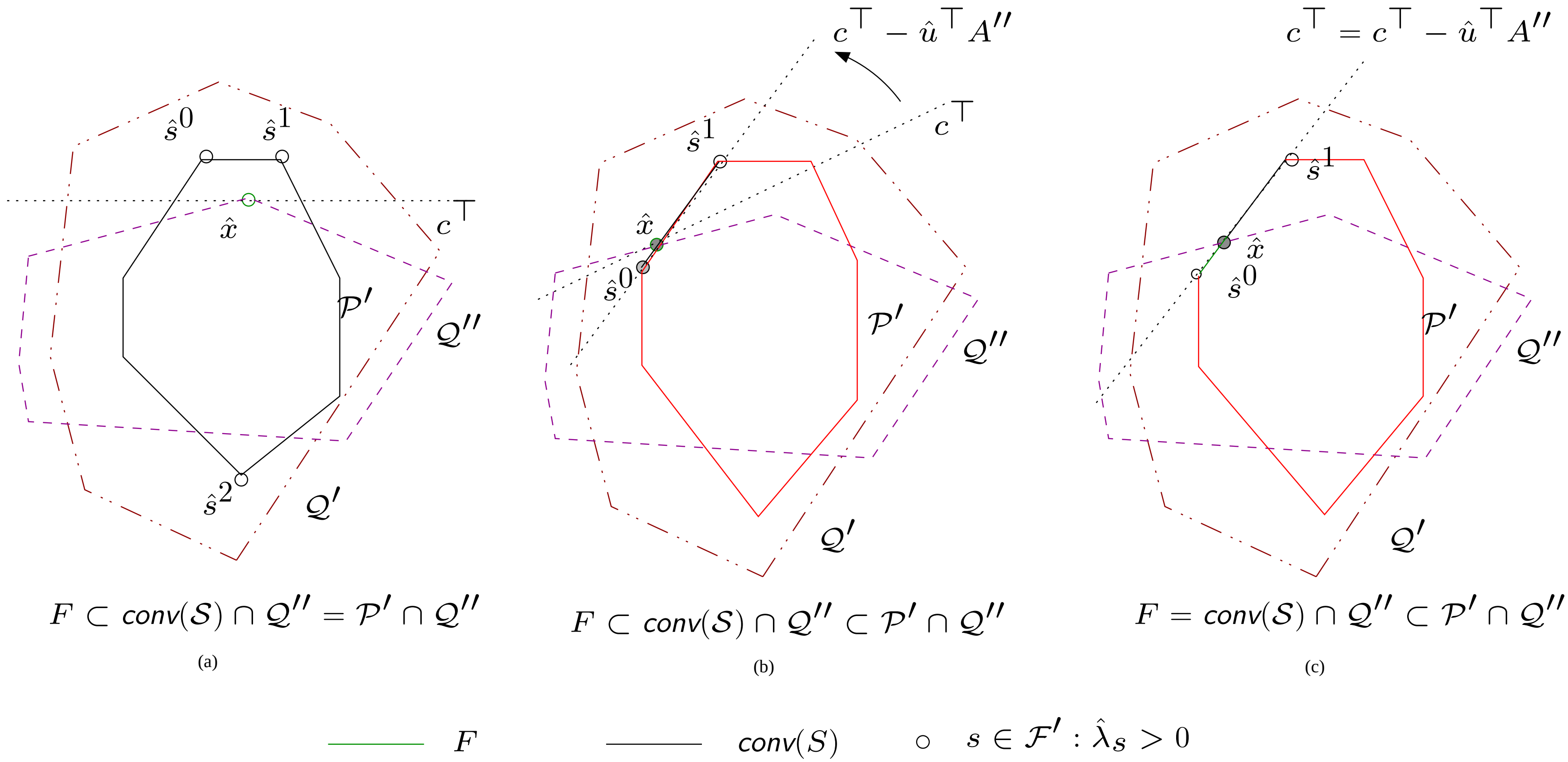
- $\mathcal{S}$ is also the set of columns of the Dantzig-Wolfe LP with reduced cost 0 at optimality, which leads to the following.

Theorem 1. $D = \{s \in \mathcal{F}' : \hat{\lambda}_s > 0\} \subseteq \mathcal{S}$.

Theorem 2. *The convex hull of* $\mathcal{S}$ *is a face of* $\mathcal{P}'$ *containing the optimal face* F *of* $\min_{x \in \mathcal{P}'}\{c^\top x : A''x \geq b''\}$.

Theorem 3. *If* $(a, \beta) \in \mathbb{R}^{(n+1)}$ *is an improving inequality, then there must exist an* $s \in D$ *such that* $a^\top s < \beta$.

- Hence, any improving inequality must be violated by
  - $\hat{x}$,
  - at least one alternative optimal solution to $z_{LR}(\hat{u})$, and
  - at least one $s \in \mathcal{F}'$ such that $\hat{\lambda}_s > 0$.



$F \subset conv(\mathcal{S}) \cap \mathcal{Q}'' = \mathcal{P}' \cap \mathcal{Q}''$

(a)

$F \subset conv(\mathcal{S}) \cap \mathcal{Q}'' \subset \mathcal{P}' \cap \mathcal{Q}''$

(b)

$F = conv(\mathcal{S}) \cap \mathcal{Q}'' \subset \mathcal{P}' \cap \mathcal{Q}''$

(c)

Price and Cut (revisited)

- Idea: Use the optimal decomposition to help generate improving inequalities.
- Rather than (or in addition to) separating $\hat{x}$, separate each $s \in D$.
- As with RC, it is often much easier to separate a member of $\mathcal{F}'$ from $\mathcal{P}$ than an arbitrary real vector, such as $\hat{x}$.
- RC only gives us one member of S to separate, while PC gives us a set $D \subseteq \mathcal{S}$, one of which must be violated.

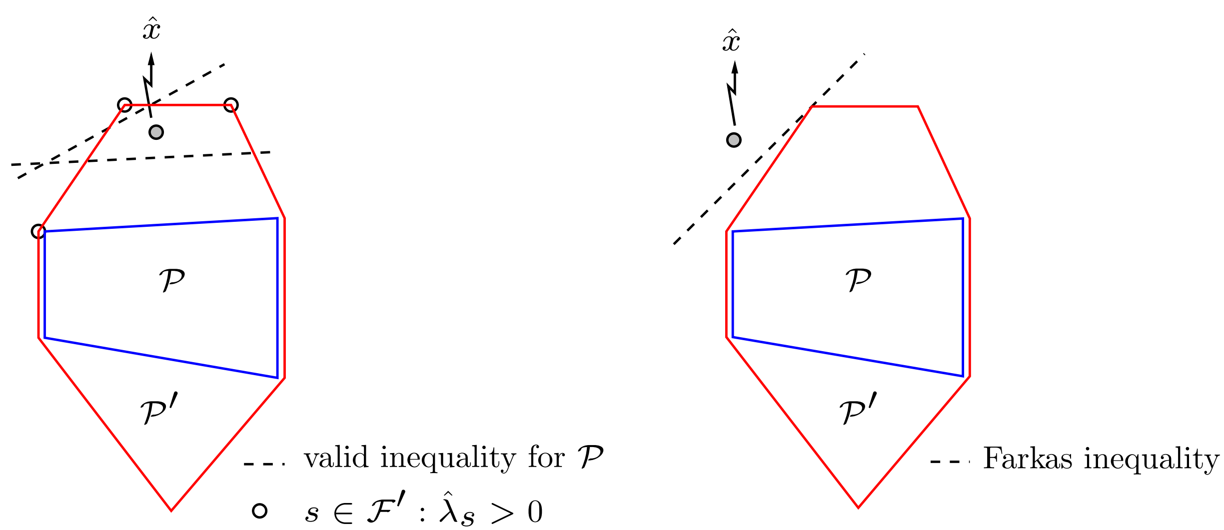
Decompose and Cut (DC)

Decompose and Cut: Use CP as the bounding problem.

$$z_{CP} = \min_{x \in \mathcal{P}'} \{c^\top x : A''x \geq b''\}$$

Compute the decomposition $\hat{\lambda}$ of $\hat{x}$, then separate each $s \in D$, as in PC.

- Both DC and PC separate the members of a decomposition of $\hat{x}$.
- DC may be more efficient than PC, since we only need to compute the decomposition when standard separation fails.



(a) $\hat{x} \in \mathcal{P}'$ (b) $\hat{x} \notin \mathcal{P}'$

Decompose and Cut (details)

- Separation in Decompose and Cut

  1. Attempt to decompose $\hat{x}$ into a convex combination of members of $\mathcal{F}'$ by solving the *Decomposition LP*:

$$\max_{\lambda \in \mathbb{R}_+^{\mathcal{F}'}} \{\mathbf{0}^\top \lambda : \sum_{s \in \mathcal{F}'} s \lambda_s = \hat{x}, \sum_{s \in \mathcal{F}'} \lambda_s = 1\}, \tag{7}$$

  2.1 If (7) is feasible, set $D = \{s \in \mathcal{F}' : \hat{\lambda}_s > 0\}$
  2.2 Else, return a Farkas inequality (a, β) valid for $\mathcal{P}' \subseteq \mathcal{P}$ which violates $\hat{x}$.
  3. Separate each $s \in D$ and return any cuts that also violate $\hat{x}$.

- Solving the Decomposition LP

  1.0 Generate an initial subset $\mathcal{G}$ of $\mathcal{F}'$.
  1.1 Solve (7) with $\mathcal{F}'$ replaced by $\mathcal{G}$ using the dual simplex algorithm.
  1.2a If (7) is feasible, return $D = \{s \in \mathcal{F}' : \hat{\lambda}_s > 0\}$.
  1.2b Else, optimize over $\mathcal{P}'$ using the resulting Farkas inequality (row of B^{-1}). If the result has negative reduced cost, add it to $\mathcal{G}$ and go to Step 1.1, else return the Farkas inequality.

Outline

- Preliminaries, Traditional Decomposition Methods
  - Dantzig-Wolfe Decomposition
  - Lagrangian Relaxation
  - Cutting Plane Method
- Dynamic Decomposition Methods
  - Price and Cut
  - Relax and Cut
  - Decompose and Cut
- Applications/Examples
- DECOMP Framework

Separating the Members of $\mathcal{F}'$

- All three dynamic decomposition methods rely on the existence of a class of valid inequalities for $\mathcal{P}$ for which it is
  - difficult to separate an arbitrary fractional solution, but
  - easy to separate members of $\mathcal{F}'$.
- Hence, we do not need to know how to separate an arbitrary fractional solution.
- Given a relaxation and such a class of inequalities, any of the three dynamic decomposition method can be applied.
- Question: Does such classes of inequalities exist in practice? Yes.
- Question: What is the complexity of the separation problem given a decomposition?

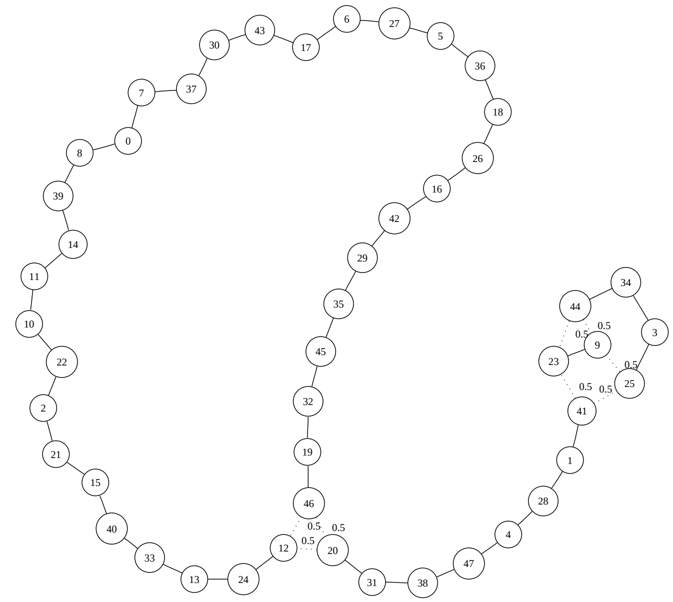
Traveling Salesman Problem

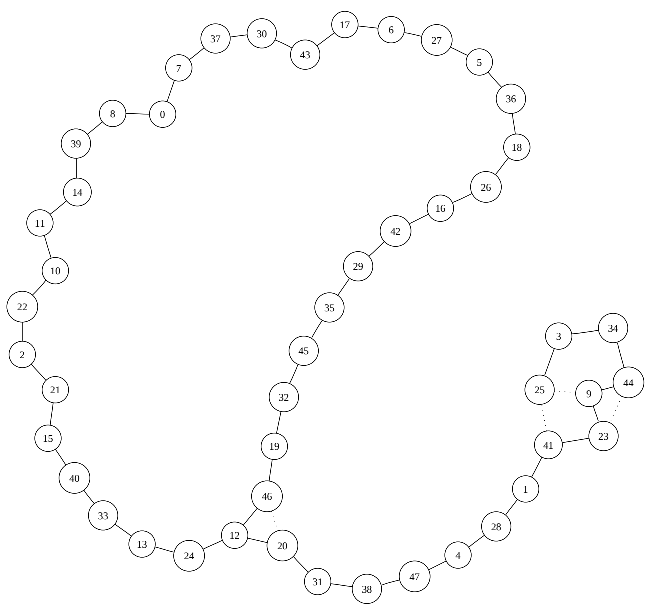
ILP Formulation:

$$\begin{array}{lcll} \sum_{e\in\delta(\{u\})} x_e & = & 2 & \forall u \in V & (1) \\ \sum_{e\in\delta(S)} x_e & \geq & 2 & \forall S \subset N, 2 \leq |S| \leq n-1 & (2) \end{array}$$

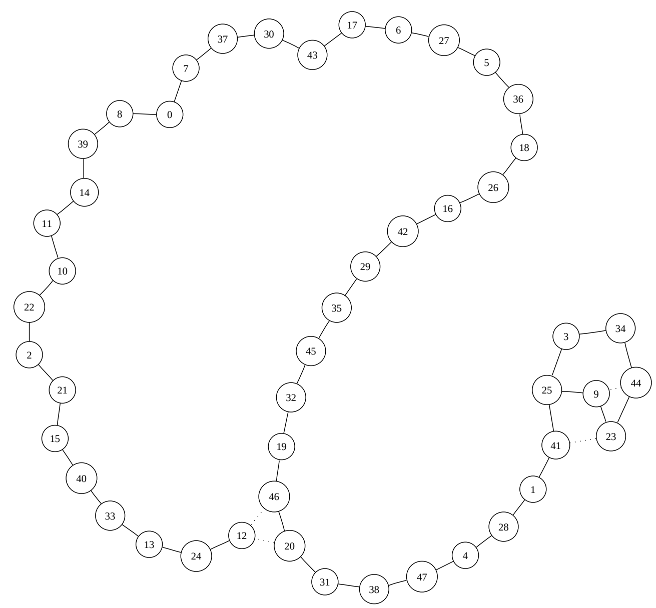
- Relaxation: 1-Tree: $\mathcal{F}' = \{x \in \mathbb{B}^{|E|} : \sum_{e\in E} x_e = n, \ \sum_{e\in\delta(S)} x_e \geq 1 \ \forall S \subset V\}$
- Facets of TSP (under certain conditions): SECS (2), Combs
  - SEC inequalities violated by a given 1-tree can be found by simply determining the cycle C (these can also be extended).
  - Comb inequalities violated by a given 1-Tree from the comb polyhedron can be found similarly by setting $H = C$ and T_i to be chains originating at nodes in C.

Example of Decomposition TSP/1-Tree



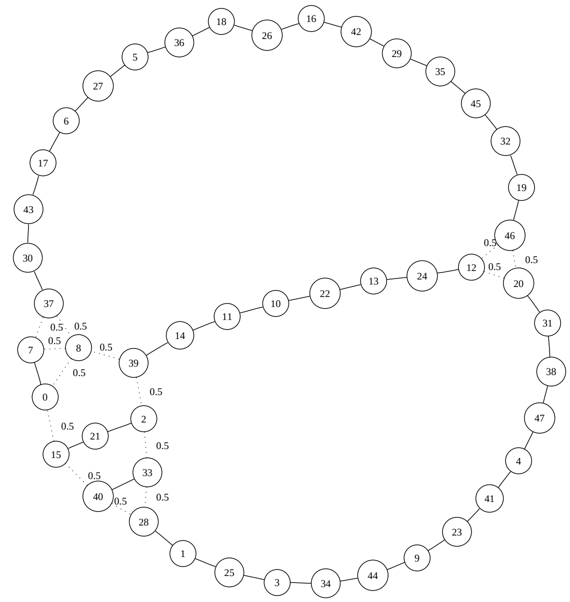


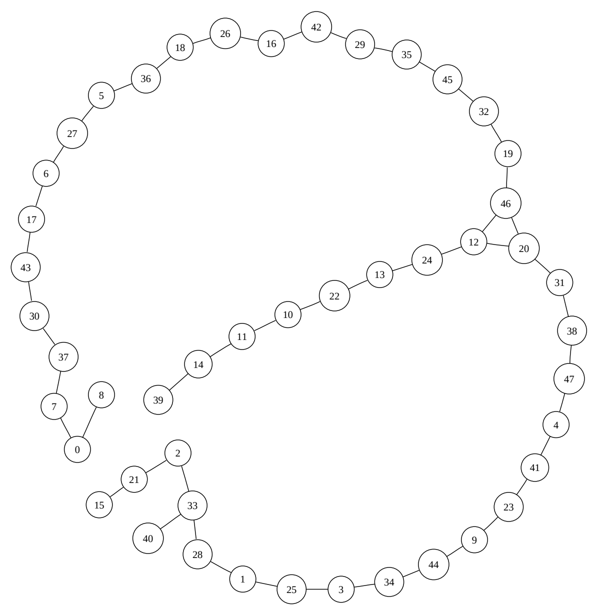
$\hat{\lambda}^1 = \frac{1}{2}$



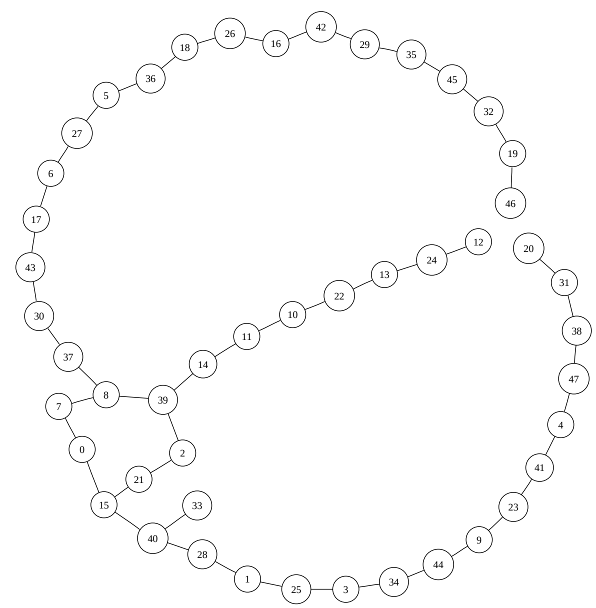
$\hat{\lambda}^2 = \frac{1}{2}$

Example of Decomposition TSP/1-Tree



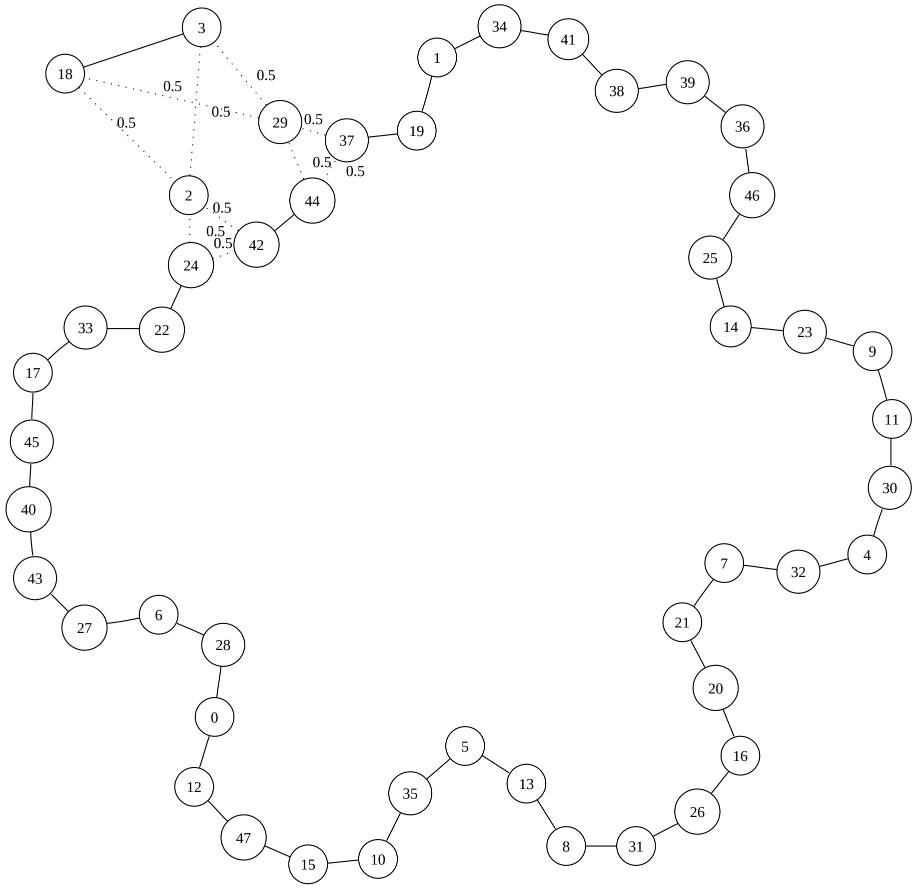


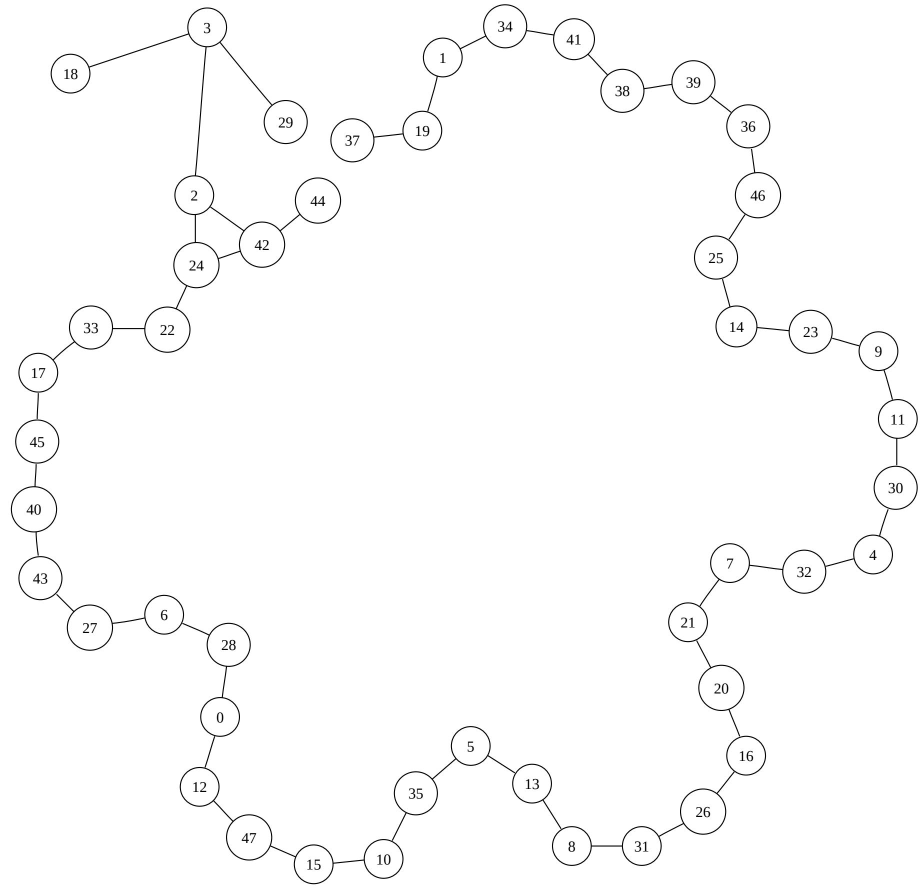
$\hat{\lambda}^1 = \frac{1}{2}$



$\hat{\lambda}^2 = \frac{1}{2}$

Example of Decomposition TSP/1-Tree





$\hat{\lambda}^2 = 0.28947$

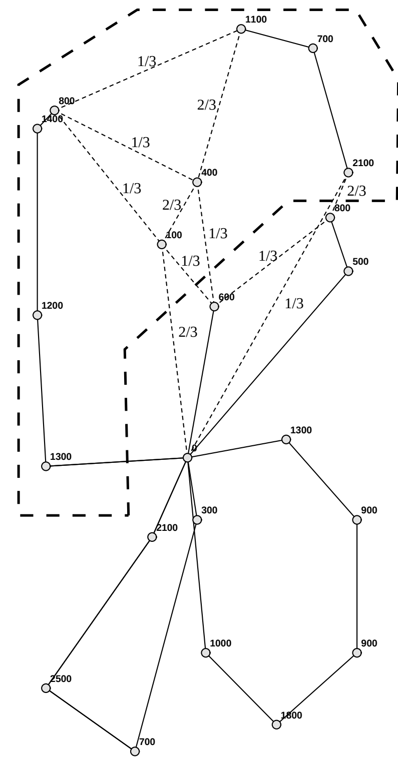
The Vehicle Routing Problem

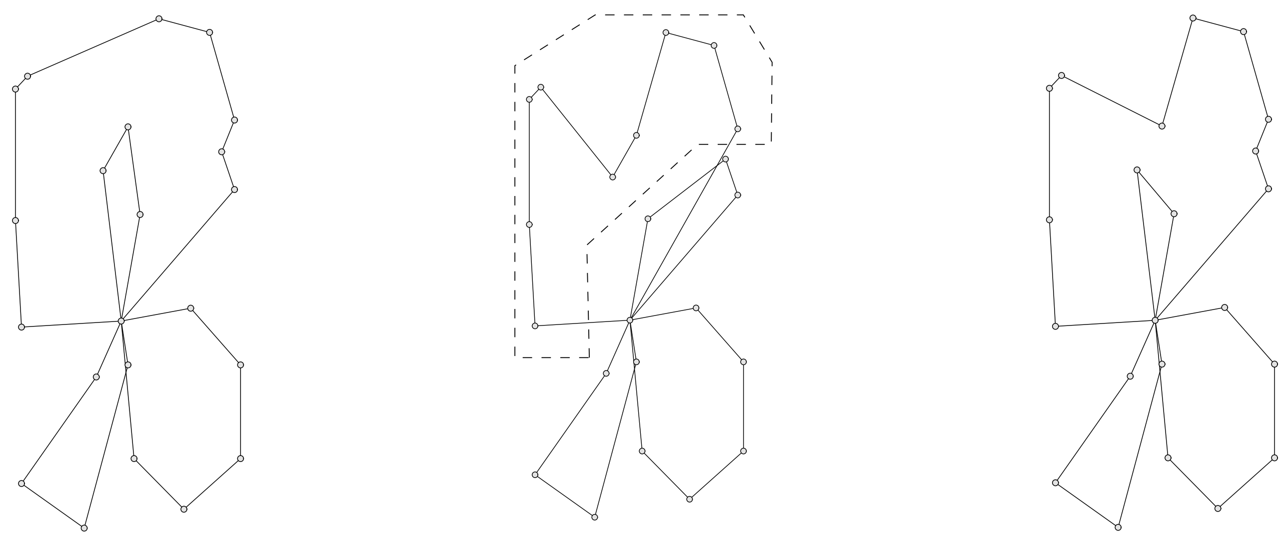
ILP Formulation:

$$
\begin{array}{rcll}
\sum_{e\in\delta(\{0\})} x_e & = & 2k & \quad (1)\\
\sum_{e\in\delta(\{u\})} x_e & = & 2 \qquad \forall u \in V \setminus \{0\} & \quad (2)\\
\sum_{e\in\delta(S)} x_e & \geq & 2b(S) \quad \forall S \subset V \setminus \{0\},\ |S| > 1 & \quad (3)
\end{array}
$$

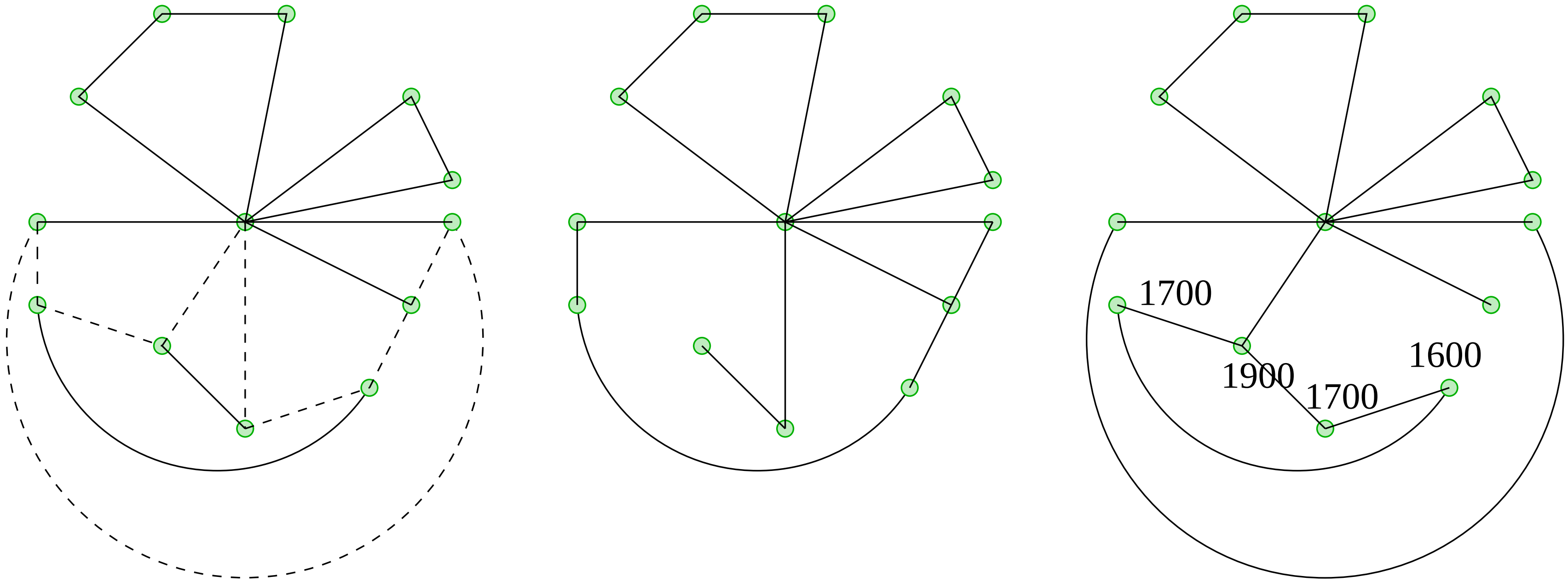
$b(S)$ = lower bound on the number of trucks required to service S
= $\lceil (\sum_{i\in S} d_i)/C \rceil$ (normally)

- Relaxations:
  - Multiple Traveling Salesman Problem: Set $C = \sum_{i\in S} d_i$.
  - k-Tree: Set $C = \sum_{i\in S} d_i$. Relax (2) but leave $\sum_{e\in E} x_e = n + k$.
- Facets of VRP (under certain conditions): GSECs (3), Combs, Multistars
- *Decompose and Cut* - VRP/kTSP for GSECs [Ralphs, et al. *On the Capacitated Vehicle Routing Problem*, Mathematical Programming 03]
- *Relax and Cut* - VRP/kTree for GSECs, Combs, Multistars [Martinhon, Lucena, Maculan, *Stronger K-Tree Relaxations for the VRP*, unpublished 01]





(a) $\hat{\lambda}^1 = \frac{1}{3}$ (b) $\hat{\lambda}^2 = \frac{1}{3}$ (c) $\hat{\lambda}^3 = \frac{1}{3}$



(a) $\hat{x}$ (b) $\hat{\lambda}^1 = \frac{1}{2}$ (c) $\hat{\lambda} = \frac{1}{2}$

Axial Assignment Problem

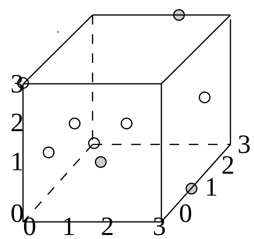
ILP Formulation:

$$\begin{array}{llll} \min & \sum_{(i,j,k)\in I\times J\times K} c_{ijk}x_{ijk} & & \\ \text{s.t.} & \sum_{(j,k)\in J\times K} x_{ijk} & = & 1 & \forall i \in I \\ & \sum_{(i,k)\in I\times K} x_{ijk} & = & 1 & \forall j \in J \\ & \sum_{(i,j)\in I\times J} x_{ijk} & = & 1 & \forall k \in K \\ & x_{ijk} & \in & \{0,1\} & \forall (i,j,k) \in I\times J\times K \end{array}$$

- Relaxation: Assignment Problem: Relax first set of constraints.
- Facets of AAP: $Q_1(u)$ and $P_1(u,v)$ - cliques of the intersection graph $K_{n,n,n}$
- Let $C(u) = \{w \in T : |u \cap w| = 2\}$, $C(u,v) = \{w \in T : |u \cap w| = 1, |w \cap v| = 2\}$

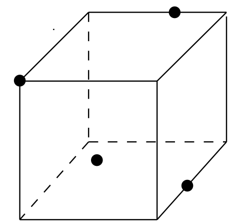
$$\begin{array}{llll} x_u + \sum_{w\in C(u)} x_w & \leq & 1 & \forall u \in T \\ x_u + \sum_{w\in C(u,v)} x_w & \leq & 1 & \forall u,v \in T, u\cap v = \emptyset \end{array}$$

- *Relax and Cut* - AP3/AP for Q_1 [Balas and Saltzman, *An Algorithm for the Three-Index Assignment Problem* Operations Research 91]



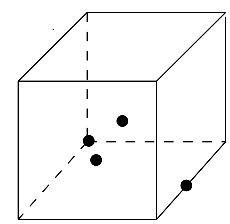
(0, 0, 3)	1/3	(0, 3, 1)	2/3
(1, 0, 1)	1/3	(1, 1, 2)	2/3
(2, 1, 0)	1/3	(2, 2, 0)	1/3
(2, 3, 2)	1/3	(3, 0, 0)	1/3
(3, 2, 3)	2/3		

(a) $\hat{x}$



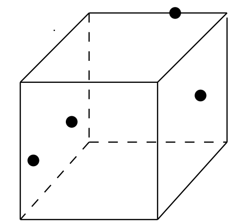
(3, 0, 0)
(0, 3, 1)
(1, 1, 2)
(3, 2, 3)

(b) $\hat{\lambda}_1 = \frac{1}{3}$



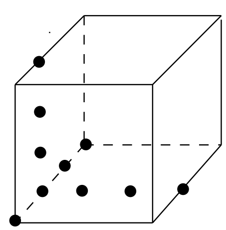
(2, 2, 0)
(0, 3, 1)
(1, 1, 2)
(0, 0, 3)

(c) $\hat{\lambda}_2 = \frac{1}{3}$



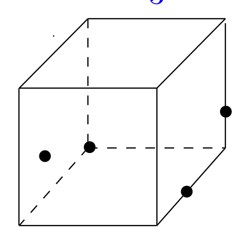
(2, 1, 0)
(1, 0, 1)
(2, 3, 2)
(3, 2, 3)

(d) $\hat{\lambda}_3 = \frac{1}{3}$



$\sum_{w \in C(0,0,1)} \hat{x}w = 1\ 1/3 > 1$

(e) $Q_1(0, 0, 1)$



$\sum_{w \in C((0,0,3),(1,3,1))} \hat{x}w = 1\ 1/3 > 1$

(f) $P_1((0, 0, 3), (1, 3, 1))$

Applications Under Development

- Traveling Salesman Problem
  - 1-Tree : SECs, Combs
- Vehicle Routing Problem
  - k-Traveling Salesman Problem : GSECs
  - k-Tree : GSECs, Combs, Multistars
- Axial Assignment Problem
  - Assignment Problem : Clique-Facets
- Steiner Tree Problem
  - Minimum Spanning Tree : Lifted SECs, Partition Inequalities
- Knapsack Constrained Circuit Problem
  - Knapsack Problem : Cycle Cover, Maximal-Set Inequalities
- Edge-Weighted Clique Problem
  - Tree Relaxation : Trees, Cliques

Outline

- Preliminaries, Traditional Decomposition Methods
  - Dantzig-Wolfe Decomposition
  - Lagrangian Relaxation
  - Cutting Plane Method
- Dynamic Decomposition Methods
  - Price and Cut
  - Relax and Cut
  - Decompose and Cut
- Applications/Examples
- DECOMP Framework

DECOMP Framework

- Goal: Framework to allow for direct comparison of all three dynamic decomposition methods.
- COIN-OR: **COmputational INfrastructure for Operations Research**
  - BCP: Parallel Branch, Cut, and Price (LP-based Bounding)
  - ALPs: Abstract Library for Parallel Search
    - BiCePS: Branch, Constrain and Price Software (Generic Bounding)
    - BLIS: BiCePS Linear Integer Solver
- DECOMP will provide
  - CGL-like full implementation of *Decompose and Cut*
  - BiCePS *plug-and-play* for *Price and Cut* and *Relax and Cut*
- DECOMP user implements two methods:
  - `solve_relaxed_problem` (includes several built-in solvers)
  - `separate_relaxed_point`

Some Implementational Details

- Consistency condition
  - $\hat{x}_i = 0 \Rightarrow s_i = 0, \forall s \in D$
  - $\hat{x}_i = 1 \Rightarrow s_i = 1, \forall s \in D$
  - Allows preprocessing of decomposition LP
    - remove all 0-rows
    - remove all 1-rows (redundant with convexity constraint)
  - Restricts column generation subproblem search space
- Initialization of $\mathcal{G}$
  - $\mathcal{G}^t = \{s \in \mathcal{G}^{t-1} : \hat{\lambda}_s > 0\}$
  - Solve over $\mathcal{P}'$ with $c = -\hat{x}^{\epsilon}$
- Column pool
  - For initialization of $\mathcal{G}$
  - For finding new columns (scan for negative reduced cost)

Some Implementational Details

- Finding new columns
  - Use multiple dual rays (scan B^{-1})
  - Optimize over $\mathcal{P}'$ heuristically—optimal solution is unnecessary.
  - Return several columns (not just the best) each time: heuristic exchanges
- Decomposition into members of $\mathcal{F}$ [Kopman 99]
  - Column generation subproblem is an optimization problem over $\mathcal{P}$!!
  - Applegate, Bixby, Chvátal, and Cook, *TSP Cuts Which Do Not Conform to the Template Paradigm*, Computational Combinatorial Optimization 2001
- Additional inequalities
  - With consistency, no columns cut: $\sum_{e \in E_1} x_e - \sum_{e \in E_0} x_e \leq |E_1| - 1$
  - Lifting the Farkas inequality: $\hat{x} \notin \mathcal{P}'$

Future Work

- Questions
  - How do we choose the relaxation (tradeoff between ease of finding decomposition and ease of separating members of the decomposition).
  - How do the cuts generated by separating members of $\mathcal{F}'$ compare to those generated by separating $\hat{x}$?
  - Of the cuts that are violated by a given member of $\mathcal{F}'$, which are the most likely to be improving?
  - Which members of D should we spend the most effort separating?
  - Can we "warmstart" the process of finding a decomposition in DC?
  - Can we use a global pool of members of $\mathcal{F}'$ similar to a cut pool?
  - Can parallelization help?
- Extensions
  - Consider tightening the implicit polyhedron instead of the explicit.
  - Make greater use of Farkas inequalities in Decompose and Cut.
  - Restrict column generation when finding the decomposition in DC.
  - Explore theoretical issues, such as complexity.

Conclusions

- The framework provides an alternative (and often much easier) paradigm for separation.
- It also provides a wide range of alternatives for using separation procedures to improve bounds.
- There are many variants on these basic themes.
- Little is known computationally about many of these variants.
- We are currently in the process of implementing these methods and working through the computational issues.
- Our preliminary results are promising, but there are many tradeoffs.
- Our software framework will end up in the COIN-OR repository, so that others may use it.
- A computational paper will be coming soon.