Decomposition and Dynamic Cut Generation in Integer Programming

Matthew V. Galati

Ted K. Ralphs

http://sagan.ie.lehigh.edu/mgalati

Department of Industrial and Systems Engineering

Lehigh University, Bethlehem, PA

The International Symposium on Mathematical Programming

The Technical University of Denmark, Copenhagen, Denmark, August 20, 2003

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 Library Framework

Preliminaries

- Consider the following pure integer linear program (PILP):

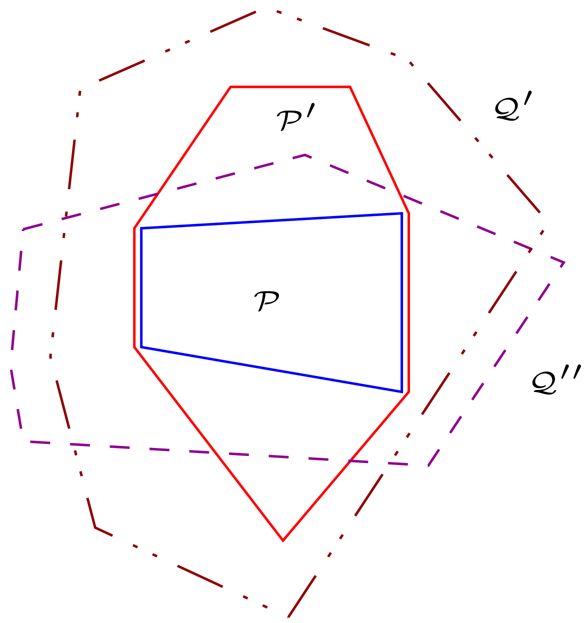
$$z_{IP} = \min_{x \in \mathcal{F}} \{c^\top x\} = \min_{x \in \mathcal{P}} \{c^\top x\} = \min_{x \in \mathbb{Z}^n} \{c^\top x : Ax \geq b\}$$

where

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

- Denote $\mathcal{P} = conv(\mathcal{F})$ and $\mathcal{P}' = conv(\mathcal{F}')$.
- Assume that optimization/separation over $\mathcal{P}$ is *difficult*.
- Assume that optimization/separation over $\mathcal{P}'$ can be done *effectively*.

Polyhedra, LP Bound, LD/DW/CP Bound



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

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

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

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

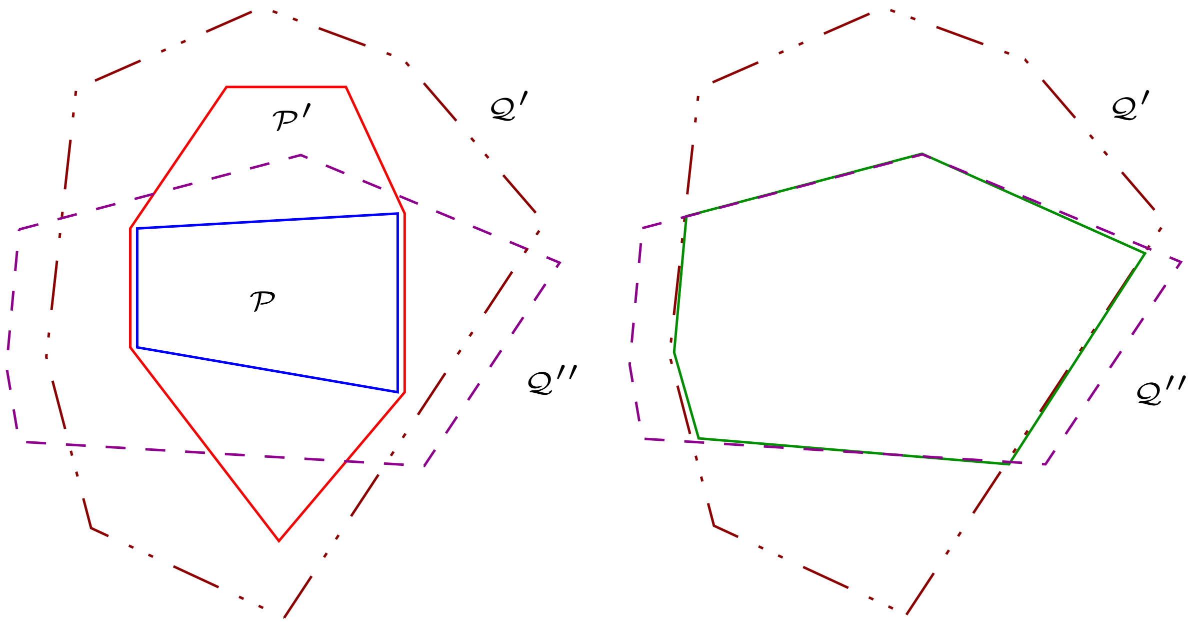
Bounding

- Goal: Compute a lower bound on z_{IP}.
- The most straightforward approach is to solve the initial LP relaxation

$$z_{LP} = \min_{x \in \mathcal{Q}} \{c^\top x\} = \min_{x \in \mathbb{R}^n} \{c^\top x : A'x \geq b', A''x \geq b''\}$$

- Decomposition approaches attempt to improve on this bound by utilizing our implicit knowledge of $\mathcal{P}'$.
- Express the constraints of $\mathcal{Q}''$ explicitly.
- Express the constraints of $\mathcal{P}'$ implicitly through solution of a subproblem.
  - Dantzig-Wolfe Decomposition
  - Lagrangian Relaxation
  - Cutting Plane Method

Polyhedra, LP Bound, LD/DW/CP Bound



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

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

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

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

$\mathcal{Q} = \mathcal{Q}' \cap \mathcal{Q}''$ (LP Bound)

Dantzig-Wolfe Decomposition

- The bound is obtained by solving the Dantzig-Wolfe LP:

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

- Solution method: simplex algorithm with dynamic column generation
- Subproblem: optimization over $\mathcal{P}'$
- Let $\hat{\lambda}$ be an optimal solution to (1) and

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

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

Lagrangian Relaxation

- The bound is obtained by solving 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 z_{LD} as an LP we see it is the dual of the Dantzig-Wolfe LP

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

- So we have $z_{IP} \geq z_{LD} = z_{DW} \geq z_{LP}$.

Cutting Plane Methods

- The bound is obtained by augmenting the initial LP relaxation with facets of $\mathcal{P}'$.
- This approach yields the bound

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

- Solution method: simplex algorithm with dynamic cut generation
- Subproblem: separation from $\mathcal{P}'$
- Note that $\hat{x}$ from (2) is an optimal solution to (6), so $z_{IP} \geq z_{CP} = z_{DW} \geq z_{LP}$.

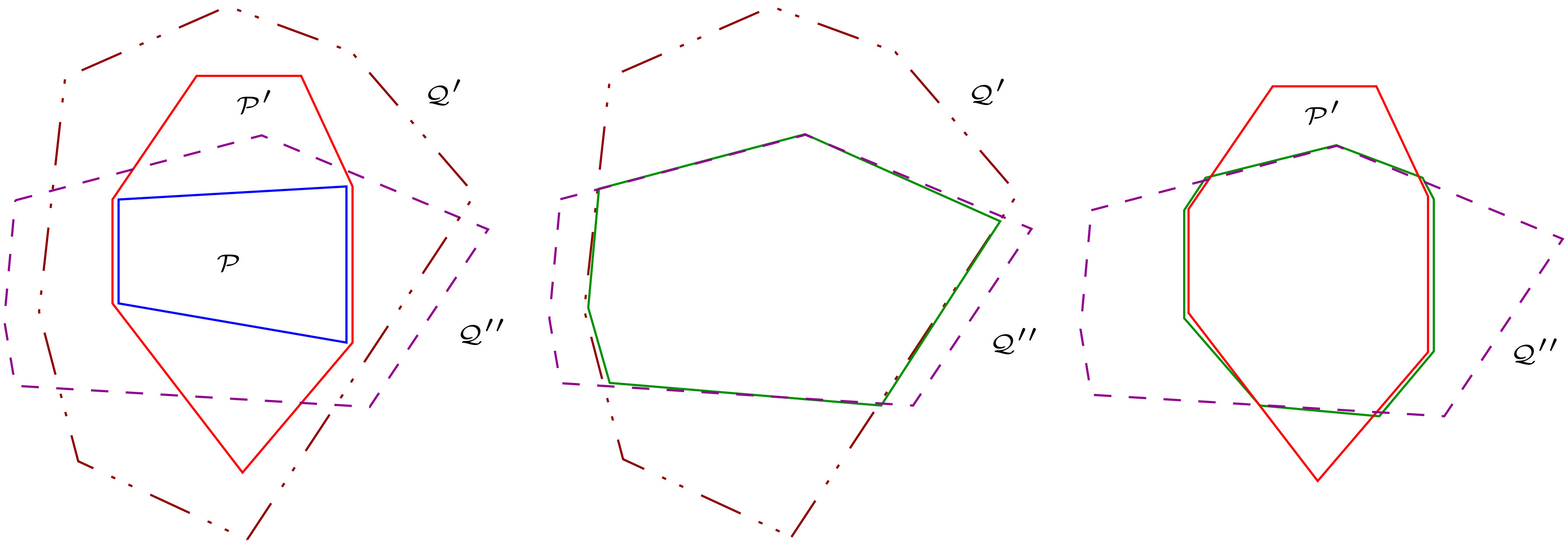
A Common Framework

- All three decomposition methods compute the same quantity [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 (inner representation).

Polyhedra, LP Bound, LD/DW/CP Bound



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

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

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

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

$\mathcal{Q} = \mathcal{Q}' \cap \mathcal{Q}''$ (LP Bound)

$\mathcal{P}' \cap \mathcal{Q}''$ (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 Library Framework

Cutting Plane Method

Cutting Plane Method (CPM)

1. Construct the initial LP relaxation LP^0 and set $i \leftarrow 0$.
   $$z_{LP} = \min_{x \in \mathbb{R}^n} \{c^\top x : A'x \geq b', A''x \geq b''\}$$
2. Solve LP^i to obtain an optimal solution $\hat{x}^i$ and lower bound $z^i \leftarrow c^\top \hat{x}^i$.
3. Attempt to separate $\hat{x}^i$ from $\mathcal{P}$, generating violated inequalities $[D^i, d^i]$.
4. If $[D^i, d^i] \neq \emptyset$, set $[A'', b''] \leftarrow \left[\begin{smallmatrix} A'' & b'' \\ D^i & d^i \end{smallmatrix}\right], i \leftarrow i + 1$ and go to Step 2, else output z^i.

- Advantage (over traditional decomposition methods): Step 3 may generate inequalities that cut off parts of $\mathcal{P}'$.
- The traditional cutting plane paradigm attempts to generate inequalities that violate $\hat{x}$.
- Adding a cut that violates $\hat{x}$ does not necessarily improve the bound.

Improving Inequalities

- An improving inequality is a valid inequality that when added to the explicit polyhedron results in an increase in the bound.

Theorem 1 *Let F be the face of optimal solutions to LP^i. Then $(a, \beta) \in \mathbb{R}^{n+1}$ is an improving inequality if and only if $a^\top y < \beta$ for all $y \in F$.*

- Violation of the optimal face is a necessary and sufficient condition for an inequality to be improving but is difficult to verify.

Corollary 1 *If $(a, \beta) \in \mathbb{R}^{n+1}$ is an improving inequality, then $a^\top \hat{x} < \beta$.*

- Violation of $\hat{x}$ is necessary (not sufficient) but is easy to verify.

Dynamic Decomposition Methods

- Goal: Improve the bound $\min_{x \in \mathcal{P}'}\{c^\top x : A''x \geq b''\}$ by dynamic tightening of the explicit polyhedron ($\mathcal{Q}''$).

Dynamic Decomposition Method

1. Construct the initial bounding subproblem P^0 and set $i \leftarrow 0$.
   $$z_{DW} = \min_{\lambda \in \mathbb{R}_+^{\mathcal{F}'}}\{c^\top(\sum_{s \in \mathcal{F}'} s\lambda_s) : A''(\sum_{s \in \mathcal{F}'} s\lambda_s) \geq b'', \sum_{s \in \mathcal{F}'} \lambda_s = 1\}$$
   $$z_{LD} = \max_{u \in \mathbb{R}_+^n} \min_{x \in \mathcal{P}'}\{(c^\top - u^\top A'')x + u^\top b''\}$$
   $$z_{CP} = \min_{x \in \mathcal{P}'}\{c^\top x : A''x \geq b''\}$$
2. Solve P^i to obtain a lower bound z^i.
3. Attempt to generate a set of improving inequalities $[D^i, d^i]$.
4. If $[D^i, d^i] \neq \emptyset$, set $[A'', b''] \leftarrow \left[\begin{smallmatrix} A'' & b'' \\ D^i & d^i \end{smallmatrix}\right], i \leftarrow i + 1$ and go to Step 2, else output z^i.

- The key is Step 3 where we attempt to generate improving inequalities.

Price and Cut (PC)

Price and Cut: use DW as the bounding subproblem

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

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

- Generation of the cuts takes place in original space - which maintains the structure of the column generation subproblem (optimization over $\mathcal{P}'$).
- PC vs CPM:
  - Both try to separate $\hat{x}$ from $\mathcal{P}$ (which is typically hard)
  - Corollary 1 provides us with motivation.
- Question: Can we take advantage of the additional information in PC (the optimal decomposition $\hat{\lambda}$) to help improve the bound?

Relax and Cut (RC)

Relax and Cut: use LD as the bounding subproblem and attempt to separate $\hat{s} \in \mathcal{F}'$.

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

- RC vs CPM - Advantage: 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 vs CPM - Disadvantage: Solving LD with subgradient — no access to original primal solution $\hat{x}$ — no way to verify the necessary condition in Corollary 1.
- 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 primal solutions to LD is

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

and $\hat{s}$ is any optimal primal solution to the Lagrangian dual.

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

- Note that separation of $\mathcal{S}$ is sufficient for an inequality to be improving.

Theorem 3 *If $\hat{\lambda}$ is an optimal solution to the DW-LP, then $D = \{s \in \mathcal{F}' : \hat{\lambda}_s > 0\} \subseteq \mathcal{S}$*

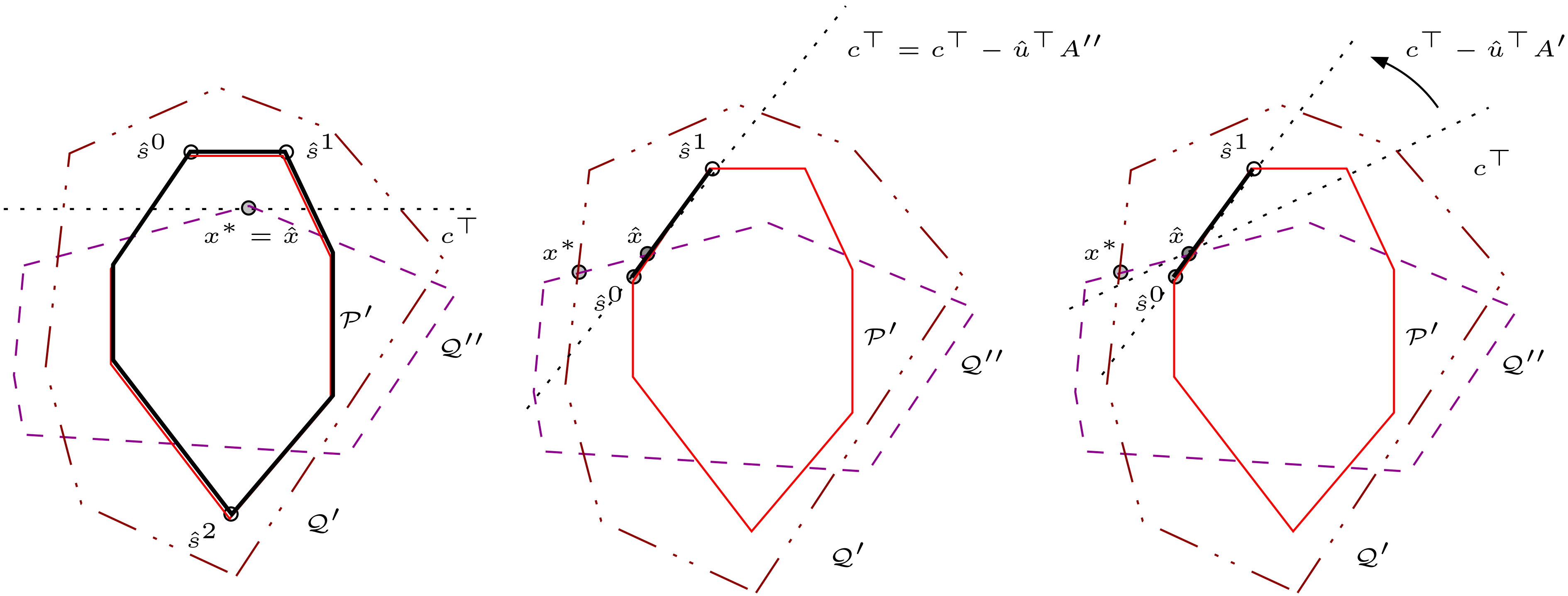
- Any $s \in D$ is an optimal primal solution for the Lagrangian dual.

Theorem 4 *If $(a, \beta) \in \mathbb{R}^{(n+1)}$ is an improving inequality, then there must exist an $s \in D$ such that $a^\top s < \beta$.*

Price and Cut (revisited)

- Idea: Rather than (or in addition to) separating $\hat{x}$, separate each $s \in D$.
- PC vs CPM - Advantage:
  - Theorem 4 gives us an alternative necessary condition for finding improving inequalities. PC gives us the optimal decomposition D.
  - Recall: It is often much easier to separate a member of $\mathcal{F}'$ from $\mathcal{P}$ than an arbitrary real vector, such as $\hat{x}$.
- PC vs RC - Advantage: RC only gives us one member of S, while PC gives us a set $D \subseteq \mathcal{S}$.

Illustration



(a) $z_{DW} = z_{LD} = z_{LP}$ (b) $z_{DW} = z_{LD} > z_{LP}$ (c) $z_{DW} = z_{LD} > z_{LP}$

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

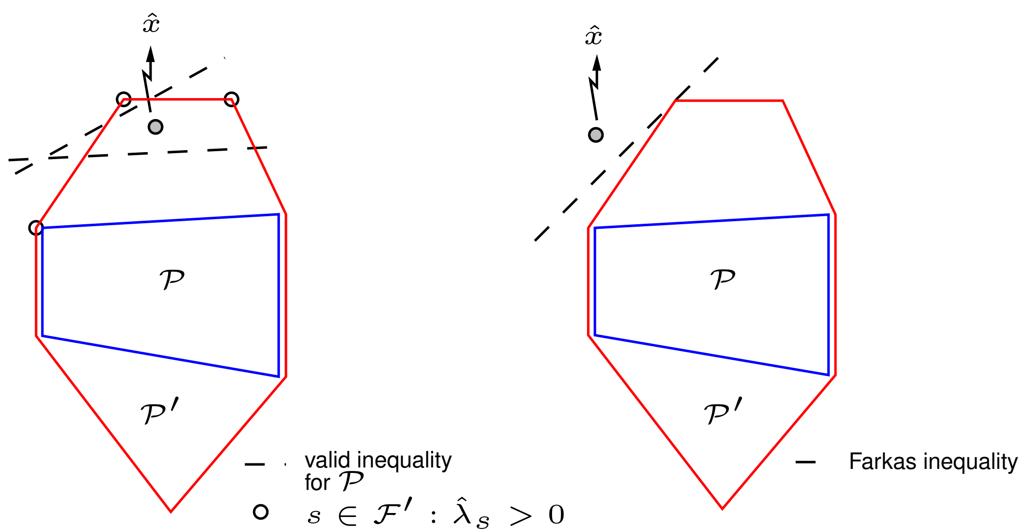
$s \in \mathcal{F}' : \hat{\lambda}_s > 0$

Decompose and Cut (DC)

Decompose and Cut: use CP as the bounding subproblem.

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

- Idea: Using a standard CPM framework — given a fractional point $\hat{x}$, compute the decomposition $\hat{\lambda}$, then separate each $s \in D$ as in PC (*inverse DW*).
- PC vs DC - Advantage: DC may be more efficient than PC since we only compute the decomposition when standard CPM separation fails.



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

Decompose and Cut Algorithm

- **Separation in Decompose and Cut**

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

- **Column Generation in Decompose and Cut**

  1.0 Generate an initial subset $\mathcal{G}$ of $\mathcal{F}'$.

  1.1 Solve (7) over $\mathcal{G}$ 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 Library Framework

Vehicle Routing Problem

ILP Formulation:

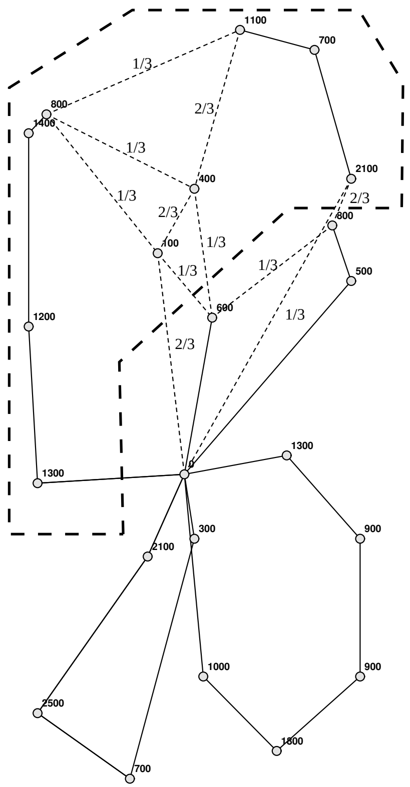
$$\begin{array}{rcll} \sum_{e\in\delta(0)} x_e & = & 2k & & (1) \\ \sum_{e\in\delta(i)} x_e & = & 2 & \forall i \in V \setminus \{0\} & (2) \\ \sum_{e\in\delta(S)} x_e & \geq & 2b(S) & \forall S \subset V \setminus \{0\},\ |S| > 1 & (3) \end{array}$$

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

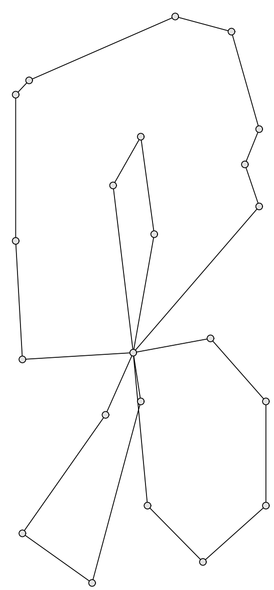
- Relaxations:
  - Multiple Traveling Salesman Problem: Set $C = \sum_{i\in S} d_i$.
  - k-Tree: Set $C = \sum_{i\in S} d_i$. Relax (2) but leave $\sum_{e\in E} x_e = n + k$.
- Facets of VRP (under certain conditions): GSECs (3), Combs, Multistars
- *Decompose and Cut* - VRP/kTSP for GSECs [Ralphs, et 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]

Example of Decomposition VRP/k-TSP

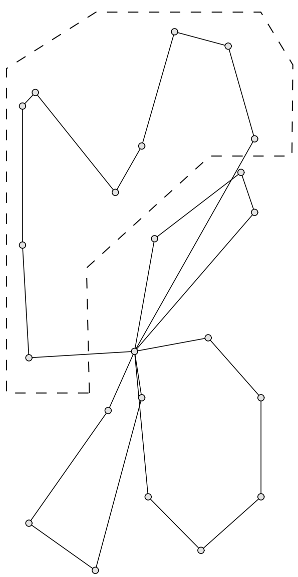
- Optimization over $kTSP$ can be done *efficiently* - TSP
- Separation of $\hat{x}$ for GSECs $\mathcal{NP}$-Complete
- Separation of a $kTSP \in \mathcal{F}'$ for GSECs in $O(n)$



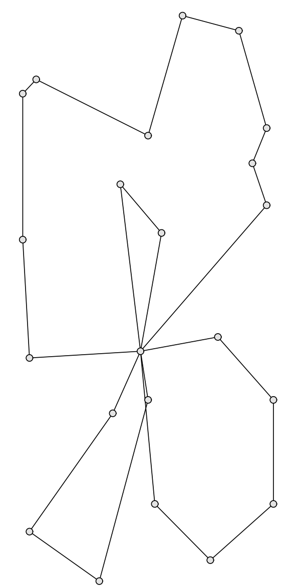
$\hat{x}$



$\hat{\lambda}^1 = \frac{1}{3}$



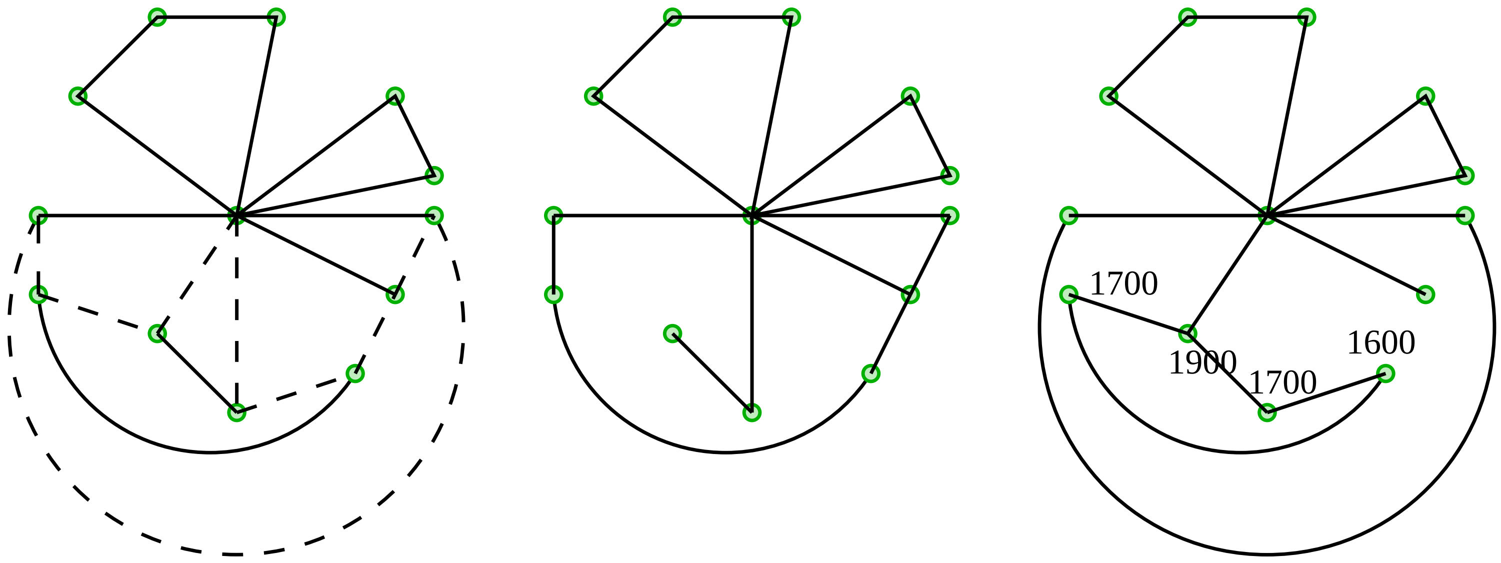
$\hat{\lambda}^2 = \frac{1}{3}$



$\hat{\lambda}^3 = \frac{1}{3}$

Example of Decomposition VRP/k-Tree

- Optimization over $kTree$ in $O(n^2 \log n)$ [Wei and Yu]
- Separation of $\hat{x}$
  - for GSECs $\mathcal{NP}$-Complete
  - for Combs and Multistars is *difficult*
- Separation of a $kTree \in \mathcal{F}'$
  - for GSECs in $O(n)$
  - for Combs and Multistars can be done *efficiently*



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

Axial Assignment Problem

PILP Formulation:

$$
\begin{array}{llcll r}
\min & \sum_{(i,j,k)\in T} c_{ijk} x_{ijk} & & & & \\
 & \sum_{(j,k)\in J\times K} x_{ijk} & = & 1 & \forall i \in I & (1) \\
 & \sum_{(i,k)\in I\times K} x_{ijk} & = & 1 & \forall j \in J & (2) \\
 & \sum_{(i,j)\in I\times J} x_{ijk} & = & 1 & \forall k \in K & (3) \\
 & x_{ijk} \in \{0,1\} & & & \forall (i,j,k) \in T = I \times J \times K & (4)
\end{array}
$$

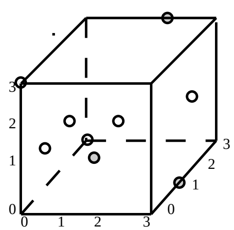
- Relaxation: Assignment Problem - relax (1)
- Facets of AAP: $Q_1(u)$ and $P_1(u,v)$ - cliques of the intersection graph of $K_{n,n,n}$
- Let $C(u) = \{w \in T : |u \cap w| = 2\}$, $C(u,v) = \{w \in T : |u \cap w| = 1, |w \cap v| = 2\}$

$$
\begin{array}{lcll r}
x_u + \sum_{w\in C(u)} x_w & \leq & 1 & \forall u \in T & (5) \\
x_u + \sum_{w\in C(u,v)} x_w & \leq & 1 & \forall u,v \in T, u \cap v = \emptyset & (6)
\end{array}
$$

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

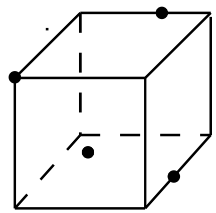
Example of Decomposition AAP/AP

- Optimization over AP in $O(n^{5/2} \log(nC))$
- Separation of $\hat{x}$ for Clique Facets in $O(n^3)$
- Separation of an $AP \in \mathcal{F}'$ for Clique Facets in $O(n)$



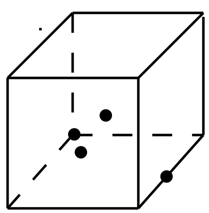
$(0, 0, 3)$	$1/3$	$(0, 3, 1)$	$2/3$
$(1, 0, 1)$	$1/3$	$(1, 1, 2)$	$2/3$
$(2, 1, 0)$	$1/3$	$(2, 2, 0)$	$1/3$
$(2, 3, 2)$	$1/3$	$(3, 0, 0)$	$1/3$
$(3, 2, 3)$	$2/3$		

(a) $\hat{x}$



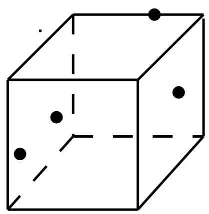
$(3, 0, 0)$
$(0, 3, 1)$
$(1, 1, 2)$
$(3, 2, 3)$

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



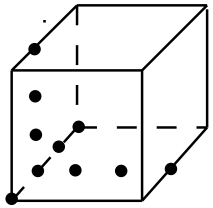
$(2, 2, 0)$
$(0, 3, 1)$
$(1, 1, 2)$
$(0, 0, 3)$

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



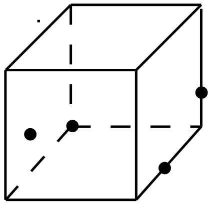
$(2, 1, 0)$
$(1, 0, 1)$
$(2, 3, 2)$
$(3, 2, 3)$

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

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 Library Framework

DECOMP Library Framework

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

Decompose and Cut Implementation Details

- Initialization of $\mathcal{G}$: solve over $\mathcal{P}'$ with $c = -\hat{x}^{\epsilon}$.
- Active LP column management.
- Lifting the Farkas inequality ($\hat{x} \notin \mathcal{P}'$).
- Consistency Condition - restriction of column generation search
  - $\hat{x}_i = 0 \Rightarrow s_i = 0, \forall s \in D$
  - $\hat{x}_i = 1 \Rightarrow s_i = 1, \forall s \in D$
- Is it necessary to be exact in solving the column generation subproblem?
  - Try optimizing over $\mathcal{P}'$ heuristically first - need negative reduced cost.
  - Do we necessarily want *extreme points* of $\mathcal{P}'$?
- 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

Applications Under Development

- Vehicle Routing Problem
  - k-Traveling Salesman Problem : GSECs
  - k-Tree : GSECs, Combs, Multistars
- Axial Assignment Problem
  - Assignment Problem : Clique-Facets
- Steiner Problem in Graphs
  - 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
- Subtour Elimination Problem [G. Benoit / S. Boyd] (LP context)
  - Fractional 2-Factor Problem : SECs

Conclusions

- Provided some insight into the relationship between: the optimal LP face F, the optimal DW solution $\hat{x}$, the optimal LD solution $\hat{s}$ and the knowledge gained from the optimal decomposition $\hat{\lambda}$.
- Alternative (and often much easier) methods for separation: over $\mathcal{F}'$ vs $\mathcal{Q}$.
    - Incorporated this idea into traditional *Price and Cut*.
    - Introduced a promising new paradigm for separation *Decompose and Cut*.
- Presented a unifying framework for dynamic cut generation in traditional decomposition methods.
    - We are currently in the process of developing a software framework DECOMP to implement and directly compare each of these methods.