Chapter 1

Reformulation and Decomposition Techniques

1.1 Lagrangean relaxation

Consider the MIP problem

$$z_{IP} = \max\{c^T x : A^1 x \leq b^1,\ A^2 x \leq b^2,\ x \in \mathbb{Z}^n\}, \qquad (1)$$

where $A^i \in \mathbb{R}^{m_i \times n}$ and $b^i \in \mathbb{R}^{m_i}$, for $i = 1, 2$. Denote $X = \{x \in \mathbb{R}^n : A^2 x \leq b^2,\ x \in \mathbb{Z}^n\}$.

Definition 1 (Lagrangean relaxation)**.** *Given $u \in \mathbb{R}_+^{m_1}$, the Lagrangean relaxation is the following MIP*

$$v(u) = \max\{c^T x + u^T(b^1 - A^1 x) : x \in X\}.$$

Remark 2. *Observe that we have we have $v(u) \geq z_{IP}$ for all $u \in \mathbb{R}_+^{m_1}$.*

Definition 3 (Lagrangean dual)**.** *The Lagrangean dual is the following optimization problem*

$$v_{LD} = \min\{v(u) : u \in \mathbb{R}_+^{m_1}\}.$$

Remark 4. *v_{LD} is the 'best possible value' for the upper bound $v(u)$.*

Theorem 5. *Denote z_{LP} the optimal value of the continuous relaxation of the MIP problem (1). Then*

1. *Consider the following linear program*

   $$w_{LD} = \max\{c^T x : A^1 x \leq b^1,\ x \in \mathrm{conv}(X)\}.$$

   Then $v_{LD} = w_{LD}$.

2. *We have that $z_{IP} \leq v_{LD} \leq z_{LP}$.*

To Do #1: Describe subgradient algorithm of optimization the Lagrangian Dual.

To Do #2: Add example and connect this to code. Show that using the lagrangian dual is faster than solving the original problem. Example might be chosen to have an efficiently solvable lagrangian dual, such as TU constraints.

1.2 Column generation

1.2.1 The master problem and the pricing subproblem

Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. Consider the following linear program

$$z_{LP} = \min \left\{ c^T x : \sum_{i=1}^{n} A^i x_i = b, x \geq 0 \right\} \qquad (LP).$$

Definition 6 (Master problem)**.** *Given $I \subseteq \{1, \dots, n\}$, consider the following restriction of the above LP:*

$$z_{LP}(I) = \min \left\{ c^T x : \sum_{i \in I} A^i x_i = b \; x_i \geq 0, \textit{ for all } i \in I \right\} \qquad (MLP).$$

Let $\bar{x}_i, i \in I$ be an optimal solution to (MLP) and let $\bar{y} \in \mathbb{R}^m$ be an optimal solution to its dual. Then $\bar{x}_i, i \in I,\ \bar{x}_i := 0, i \notin I$ is an optimal solution to (LP) if and only if

$$c_i - \bar{y}^T A^i \geq 0, \text{ for all } i = 1, \dots, n.$$

(that is, if and only if $\bar{y}$ is also a feasible solution to the dual of (LP).)

Definition 7 (The pricing subproblem)**.** *Let $\bar{y} \in \mathbb{R}^m$ be an optimal solution to the dual of (MLP). Consider the following optimization problem*

$$w_{SP} = \min\{c_i - \bar{y}^T A^i : i = 1, \dots, n\}.$$

Theorem 8. *Let $\bar{x}_i, i \in I$ be an optimal solution to (MLP) and let $\bar{y} \in \mathbb{R}^m$ be an optimal solution to its dual.*

1. *If $w_{SP} \geq 0$, then $\bar{x}_i, i \in I,\ \bar{x}_i := 0, i \notin I$ is an optimal solution to (LP). Thus, we can solve (LP) by solving the restricted problem (MLP).*
2. *If $w_{SP} < 0$, in order to solve solve (LP) by solving the restricted problem (MLP), we need to add more columns to (MLP).*

To Do #3: Connect this to example in first part of book (or rewrite this). Create or find code example to connect this to and show column generation version is faster.

1.2.2 Dantzig-Wolf decomposition

Consider the following MIP

$$z_{IP} = \max \left\{ \sum_{k=1}^{K} (c^k)^T x^k : A^1 + \cdots + A^K = b,\ x^k \in X^k,\ \text{for all } k = 1, \dots, K \right\}, \qquad (2)$$

where $X^k = \{x \in \mathbb{Z}^{n_k} : D^k x^k \leq d^k\}$, for $k = 1, \dots, K$.

Assume that the sets X^k, $k = 1, \dots, K$ are bounded. Consequently, for each $k = 1, \dots, K$, we can write

$$X^k = \{x^{k,t} : t = 1, \dots, T_k\}.$$

Definition 9 (Dantzig-Wolf reformulation)**.** *The following MIP is the Dantzig-Wolf reformulation of (2)*

$$\begin{aligned}
z_{DW} = \max & \sum_{k=1}^{K}\sum_{t=1}^{T_k}(c^k)^T x^{k,t} \\
& \sum_{k=1}^{K}\sum_{t=1}^{T_k} A^k x^{k,t}\lambda_{kt} = b \\
& \sum_{t=1}^{T_k}\lambda_{kt} = 1, \qquad \textit{for all } k = 1,\dots,K \\
& \lambda_{kt} \in \{0,1\} \qquad \textit{for all } k = 1,\dots,K,\ t = 1,\dots,T_k.
\end{aligned}$$

Remark 10. *Clearly,* $z_{IP} = z_{DW}$.

The continuous relaxation of the above MIP can be solved by using the column generation approach.

To Do #4: Elaborate and sonnect this to example in first part of book (or rewrite this). Create or find code example to connect this to and show decomposition version is faster.

1.3 Extended formulations

Let $P = \{x \in \mathbb{R}^n : Ax \le b\}$ be a polyhedron.

Definition 11 (Extended formulation)**.** *An extended formulation for P is a polyhedron*

$$Q = \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m : Ex + Fy = g, y \ge 0\},$$

with the property that $x \in P$ if and only if exists $y \in \mathbb{R}^m$ such that $(x,y) \in Q$.

Definition 12 (Extension complexity)**.** *The extension complexity of P is the minimum m such that there exists an extended formulation $Q \subseteq \mathbb{R}^n \times \mathbb{R}^m$ of P.*

Theorem 13. *Let P be the TSP polytope (that is, P is the convex hull of the 0-1 vectors that represent valid tours). Then the extension complexity of P is at least $2^{\Omega(n^{1/2})}$.*

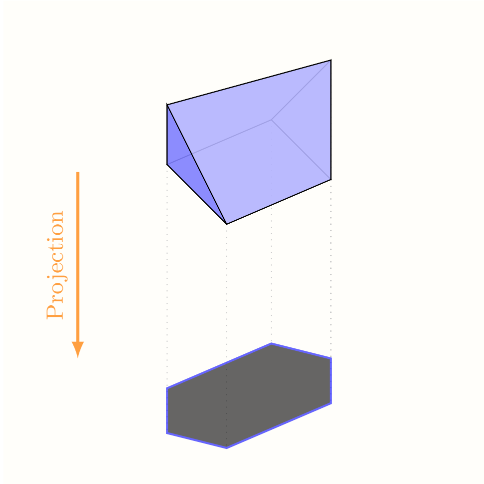
Remark 14. *This result implies that there is no* **ideal** *formulation for the TSP problem such that the formulation is of polynomial size. Conversely, if we have a formulation for the TSP problem that is of polynomial size, the this formulation cannot be ideal.*

To Do #5: Add bib references for these figures.

[1]**tikz/extended-formulation**, from **tikz/extended-formulation**. **tikz/extended-formulation**, **tikz/extended-formulation**.

[2]**tikz/extended-formulation2**, from **tikz/extended-formulation2**. **tikz/extended-formulation2**, **tikz/extended-formulation2**.

[3]**tikz/extended-formulation3**, from **tikz/extended-formulation3**. **tikz/extended-formulation3**, **tikz/extended-formulation3**.



© **tikz/extended-formulation**[1]

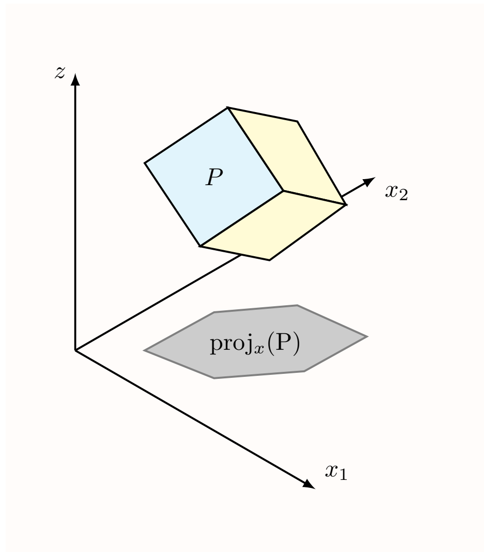
Figure 1.1: **tikz/extended-formulation**

1.4 Benders decomposition

From Wikipedia, the free encyclopedia:

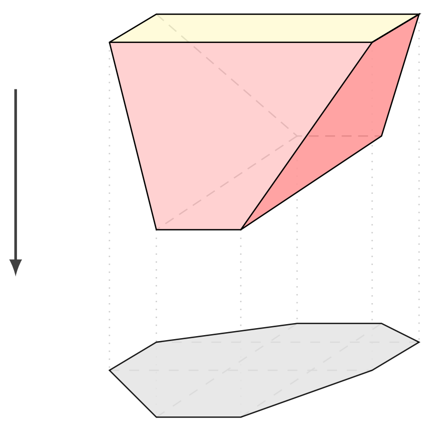
Benders' decomposition (alternatively, Benders's decomposition; named after Jacques F. Benders) is a technique in mathematical programming that allows the solution of very large linear programming problems that have a special block structure. This structure often occurs in applications such as stochastic programming.

As it progresses towards a solution, Benders' decomposition adds new constraints, so the approach is called "row generation". In contrast, Dantzig-Wolfe decomposition uses "column generation".



© **tikz/extended-formulation2**[2]

Figure 1.2: **tikz/extended-formulation2**



© **tikz/extended-formulation3**[3]

Figure 1.3: **tikz/extended-formulation3**