Integer Programming: Large Scale Methods Part II (Chapter 12 and Chapter 16)

University of Chicago
Booth School of Business
Kipp Martin

November 29, 2016
Outline

List of Files

Key Concepts

Column Generation

Cut Generation

Lagrangian Approach

Capstone Case: Capacitated Lot Sizing

Lagrangian Approach Application – Convex Optimization
Files

- genAssign.gms
- genAssignSub.gms
- genAssignData.txt
the quality of the formulation is what counts, not the size – quality usually measured by the lower bound

if we can read the problem into memory we can solve the linear relaxation (of course it may be a poor relaxation)

many interesting applications have too many constraints or variables to read into memory

if there are too many variables we generate variables as needed

we generate variables as needed by a pricing algorithm

if there are too many constraints we generate constraints as needed

we generate constraints as needed by a separation algorithm
Key Concepts

The mixed integer linear program under consideration is:

\[
\begin{align*}
\min & \quad c^\top x \\
\text{s.t.} & \quad Ax \geq b \\
& \quad Bx \geq d \\
& \quad x \geq 0 \\
& \quad x_j \in \mathbb{Z}, \ j \in I
\end{align*}
\]

\((MIP)\)

One solution approach is to solve the LP relaxation within branch and bound

\[
\begin{align*}
\min & \quad c^\top x \\
\text{s.t.} & \quad Ax \geq b \\
& \quad Bx \geq d \\
& \quad x \geq 0
\end{align*}
\]

\((LP)\)
Key Concepts

Solving the LP relaxation may not be a good idea?

Why?

What is one of our major themes?

\[ \Gamma = \left\{ x \mid Bx \geq d, \ x \geq 0, \ x_j \in \mathbb{Z}, \ j \in I \right\} \]
Key Concepts

As an alternative, we try to define $A$ and $B$ appropriately and solve:

\[
\begin{align*}
\min & \quad c^\top x \\
\text{s.t.} & \quad Ax \geq b \\
& \quad x \in \text{conv}(\Gamma)
\end{align*}
\]

How to break the constraints into $A$ and $B$ takes a fair amount of skill and practice. We want the characterization of \text{conv}(\Gamma) to be an easy hard problem.

If it is too easy there may not be a big win in terms of better lower bounds.
Key Concepts

See page 567 of the text.
Key Concepts

There are three approaches to actually solving

\[
\begin{align*}
\min & \quad c^\top x \\
\text{subject to} & \quad Ax \geq b \\
& \quad x \in \text{conv}(\Gamma)
\end{align*}
\]

(\text{conv}(\text{MIP}))

**Approach 1:** Characterize \text{conv}(\Gamma) by the inequalities that define the polyhedron. This is the \textbf{cutting plane} approach. (Tonight)

**Approach 2:** Characterize \text{conv}(\Gamma) through \textbf{inverse projection}. Often called \textbf{decomposition} or \textbf{column generation}.

**Approach 3:** Lagrangian Relaxation (Tonight)
Mixed Integer Finite Basis Theorem: If \( \Gamma = \{ x \in \mathbb{R}^n \mid Bx \geq d, x \geq 0, x_j \in \mathbb{Z}, j \in I \} \), then \( \text{conv}(\Gamma) \) is a polyhedron and there exist \( x^1, \ldots, x^r \in \Gamma \) such that

\[
\text{conv}(\Gamma) = \{ x \in \mathbb{R}^n \mid x = \sum_{i=1}^{q} z_i x^i + \sum_{i=q+1}^{r} z_i x^i, \\
\sum_{i=1}^{q} z_i = 1, z_i \geq 0, i = 1, \ldots, r \} \tag{1}
\]
Column Generation

Apply finite basis theorem to $\Gamma$

$$\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax \geq b \\
& \quad x \in \text{conv}(\Gamma)
\end{align*}$$

($\text{conv}(\text{MIP})$)

$$\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax \geq b \\
& \quad \sum_{i=1}^{q} z_i x^i + \sum_{i=q+1}^{r} z_i x^i = x \\
& \quad \sum_{i=1}^{q} z_i = 1 \\
& \quad z_i \geq 0, \quad i = 1, \ldots, r
\end{align*}$$

($\text{conv}(\text{MIPFB})$)
Column Generation

Substitute out the $x$ variables and solve the **convex hull** relaxation:

\[
\min \sum_{i=1}^{q} z_i c^\top x^i + \sum_{i=q+1}^{r} z_i c^\top x^i \\
\sum_{i=1}^{q} z_i A x^i + \sum_{i=q+1}^{r} z_i A x^i \geq b \\
\sum_{i=1}^{q} z_i = 1 \\
z_i \geq 0 \quad i = 1, \ldots, q
\]
The formulation in the $z$ variables is often called an extended variable formulation.

Using the finite basis theorem is just one way to get an extended variable formulation.

Think back to our lot sizing example – we gave an extended variable formulation without using the finite basis theorem.

Finding good extended variable formulations is an important research area.
**Important:** If *life were fair* the solution to the convex hull relaxation will binary (0/1) $z_i$. Why do I want the $z_i$ binary?

Unfortunately, what do we know about life?

So what do we do if we have fractional $z_i$?

Branch of course!

Next slide please.
Oh no, no, no: It is time for me to do what I do best and what I do best is worry!

By this point you know me pretty well.

What am I worried about? I think this has homework potential.
Let’s go back and solve the cut generation problem. The problem is:

\[
\min \sum_{e \in E} c_e x_e \\
\sum_{e \in \delta(i)} x_e = 2, \quad i \in V \\
\sum_{e \in E(S)} x_e \leq |S| - 1, \quad S \subset V \setminus \{1\} \\
\sum_{e \in E} x_e = n \\
x_e \in \{0, 1\}, \quad \forall e \in E
\]
Cut Generation

Rewrite this as

\[
\begin{align*}
\min & \sum_{e \in E} c_e x_e \\
\sum_{e \in \delta(i)} x_e &= 2, \quad i \in V \\
x & \in \Gamma
\end{align*}
\]

where \( \Gamma \) corresponds to constraints (4)-(6). In the previous slide deck, we characterized \( \Gamma \) by its **extreme points** and used column generation.

However, we can also generate the constraints in (4) through separation.
Cut Generation

We now work with this example:
Cut Generation

So a problem relaxation by deleting the tour-breaking constraints.

\[
\begin{align*}
\min & \quad \sum_{e \in E} c_e x_e \\
\sum_{e \in \delta(i)} x_e &= 2, \quad i \in V \\
\sum_{e \in E} x_e &= n \\
x_e &\in \{0, 1\}, \quad \forall e \in E
\end{align*}
\]
Cut Generation

Here is the solution. Argh! Subtours!
Cut Generation

The separation problem

\[
\max \sum_{e \in E} x_e \alpha_e - \sum_{i=1}^{n} \theta_i \\
(7)
\]

\[(SP_k(\bar{x})) \quad \text{s.t.} \quad \alpha_e \leq \theta_i, \quad e \in \delta(i), \quad i \in V \quad (8)\]

\[\theta_i \geq 0. \quad (9)\]
Let $k = 2$. The solution is:

$\theta_2 = 1, \quad \theta_3 = 1, \quad \theta_6 = 1$

$\alpha_{(2,3)} = 1, \quad \alpha_{(2,6)} = 1, \quad \alpha_{(3,6)} = 1$
Lagrangian Approach

The MIP is

\[ \begin{align*}
\text{min} & \quad c^\top x \\ 
\text{s.t.} & \quad Ax \geq b \\
& \quad x \in \Gamma
\end{align*} \]

\[(MIP)\]

where, as always, the set $\Gamma$ has special structure. Dualize the constraints $Ax \geq b$ with a conformable vector $\lambda \geq 0$ of Lagrange multipliers and get

\[ \begin{align*}
\text{min} & \quad c^\top x + \lambda^\top (b - Ax) \\
\text{s.t.} & \quad x \in \Gamma
\end{align*} \]
Lagrangian Approach

Define $L(\lambda) := \min\{c^\top x + \lambda^\top (b - Ax) \mid x \in \Gamma\}$.

The **Lagrangian Dual** problem (LD) is:

$$(LD) \quad \max\{L(\lambda) : \lambda \geq 0\}.$$ 

**Theorem**: (Weak Duality) $\nu(LD) \leq \nu(MIP)$

**Proof?**

**Theorem**: $L(\lambda)$ is a concave function of $\lambda$.

**Proof?**

**Remark**: No matter how ugly and non-convex the set $\Gamma$ is, $L(\lambda)$ is concave, so at least we are trying to maximize a concave function.
Lagrangian Approach – Using Subgradients

Concave is not too bad, but this is a **linear programming** class! I don’t have the ambition to tackle a concave function. You know me, 1) I worry; and 2) I like easy problems.

$\gamma^k$ is a **subgradient** of the concave function $L(\lambda)$ at $\lambda^k$ iff

$$L(\lambda) \leq L(\lambda^k) + (\gamma^k)^\top (\lambda - \lambda^k)$$

for all $\lambda$.

Solve the following problem **linear programming relaxation**: 

$$\begin{align*}
\max \theta \\
\theta & \leq L(\lambda^k) + (\gamma^k)^\top (\lambda - \lambda^k), \quad k = 1, \ldots, K \\
\lambda & \geq 0
\end{align*}$$

(10)

where $\gamma^k$, $k = 1, \ldots, K$ are subgradients. What is the geometry of this approach?
Lagrangian Approach – Finding Subgradients

Here is how to find a subgradient. Given a $\bar{\lambda} > 0$, solve

$$L(\bar{\lambda}) = \min \{ c^T x + \bar{\lambda}^T (b - Ax) \mid x \in \Gamma \}.$$

and obtain an optimal $x(\bar{\lambda})$.

**Theorem:** The vector $\bar{\gamma} = b - Ax(\bar{\lambda})$ is a subgradient of $L(\lambda)$ at $\lambda = \bar{\lambda}$.

**Proof:** Next slide.
Lagrangian Approach – Finding Subgradients

If

\[ L(\bar{\lambda}) = \min \{ c^T x + \bar{\lambda}^T (b - Ax) \mid x \in \Gamma \} \]

then

\[ L(\bar{\lambda}) + (b - Ax(\bar{\lambda}))^T (\lambda - \bar{\lambda}) = c^T x(\bar{\lambda}) + (b - Ax(\bar{\lambda}))^T \bar{\lambda} \]

\[ +(b - Ax(\bar{\lambda}))^T (\lambda - \bar{\lambda}) \]

\[ = c^T x(\bar{\lambda}) + (b - Ax(\bar{\lambda}))^T \bar{\lambda} \]

\[ +(b - Ax(\bar{\lambda}))^T \lambda - (b - Ax(\bar{\lambda}))^T \bar{\lambda} \]

\[ = c^T x(\bar{\lambda}) + (b - Ax(\bar{\lambda}))^T \lambda \]

\[ \geq L(\lambda) \]
Lagrangian Approach - Subgradient Algorithm

Step 0: Initialize: Choose tolerance $\epsilon$ and $K \leftarrow 1$ and $\lambda^K \leftarrow 0$. Solve $L(\lambda^K)$ and obtain subgradient $\gamma^K$.

Step 1: Solve the relaxed Lagrangian dual (11) and obtain optimal $\bar{\theta}$ and $\bar{\lambda}$.

Step 2: $K \leftarrow K + 1$, $\lambda^K \leftarrow \bar{\lambda}$, solve

$$L(\lambda^K) = \min\{c^T x + (\lambda^K)^T (b - Ax) \mid x \in \Gamma\}$$

save $\gamma^K = b - Ax(\lambda^K)$ and $L(\lambda^K)$.

Step 3: Stop if we satisfy $\epsilon$—tolerance. Otherwise, add

$$\theta \leq L(\lambda^K) + (\gamma^K)^T (\lambda - \lambda^K)$$

to the relaxed Lagrangian dual (11).

Step 4: Go to Step 1
Consider the **generalized assignment problem**. Let $x_{ij}$ be 1 if task $i$ is assigned person $j$, and 0 otherwise.

$$
\min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij}
$$

$$
\sum_{j=1}^{m} x_{ij} = 1, \quad i = 1, \ldots, n
$$

$$
\sum_{i=1}^{n} a_{ij} x_{ij} \leq b_j, \quad j = 1, \ldots, m
$$

$x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m$

How do we partition into $Ax \geq b$ and $\Gamma$? We want an easy hard problem.
Lagrangian Approach - Subgradient Algorithm

The model **genAssign.gms** solves the integer programming formulation for the generalized assignment problem.

SETS
   i "task", j "server" ;
PARAMETERS
   c(i, j), a(i, j), b(j) ;
VARIABLES
   z "minimize the cost" ;
BINARY VARIABLES
   x(i, j); 
EQUATIONS
   cost, assign( i), resource(j) ;
   cost ..   z =e=  sum( (i, j), c(i, j)*x(i, j) ) ;
   assign( i) ..   sum(j, x(i, j) ) =g=  1;
   resource(j) ..   sum(i, a(i, j)*x(i, j) ) =l= b(j);
MODEL genAssign / cost, assign, resource/;
Lagrangian Approach - Subgradient Algorithm

The data file is `genAssignData.txt`. For these data the optimal LP value is 20.32, and the optimal IP value is 34.0. A huge gap!

The GAMS model file `genAssignSub.gms` solves the Lagrangian dual. It contains two models: 1) `genAssign` without the assignment constraints; and 2) `lagDual` which is the relaxed Lagrangian dual.

Model `lagDual` calls model `genAssign` inside the cut generation loop.

The value of the Lagrangian dual is 29.5, so we are really closing the gap a lot.
Lagrangian Approach - Subgradient Algorithm

cost .. z =e= sum( (i, j), adj_c(i, j)*x(i, j) ) ;
assign( i) .. sum(j, x(i, j) ) =g= 1;
resource(j) .. sum(i, a(i, j)*x(i, j) ) =l= b(j);
MODEL genAssign / cost, resource/;

cut(allcuts) .. theta =l= LagValue( allcuts)
   + sum( i, cutCoeffSub( allcuts, i)*(lambda(i) -
                    cutCoeffLambda( allcuts, i) ) )
MODEL lagDual / cut /;

Inside the loop,

adj_c(i, j) = c(i, j) - lambda.l(i);
SOLVE genAssign USING MIP MINIMIZING z;

SOLVE lagDual USING LP MAXIMIZING theta;
Lagrangian Approach

If the Lagrangian dual is

$$\max \{ L(\lambda) : \lambda \geq 0 \}.$$ 

where $$L(\lambda) := \min \{ c^\top x + \lambda^\top (b - Ax) | x \in \Gamma \}$$. Then this is equivalent to

$$(LD) \quad \max \theta \quad \begin{array}{c}
\theta \leq c^\top x + \lambda^\top (b - Ax), \\
\forall x \in \Gamma \quad \lambda \geq 0.
\end{array}$$ (11)
Lagrangian Approach

**KEY Theorem:** The optimal value of $(LD)$ is equal to the optimal value of $CONV(MIP)$.

**Proof:** I am too tired! You do it!

All of our approaches are equivalent!
Capstone Case: Capacitated Lot Sizing

Dynamic Lot Sizing:

Variables:

\[ x_{it} \] – units of product \( i \) produced in period \( t \)
\[ I_{it} \] – inventory level of product \( i \) at the end of the period \( t \)
\[ y_{it} \] – is 1 if there is nonzero production of product \( i \) during period \( t \), 0 otherwise

Parameters:

\[ d_{it} \] – demand product \( i \) in period \( t \)
\[ f_{it} \] – fixed cost associated with nonzero production of product \( i \) in period \( t \)
\[ c_{it} \] – marginal production cost for product \( i \) in period \( t \)
\[ h_{it} \] – marginal holding cost charged to product \( i \) at the end of period \( t \)
\[ g_t \] – production capacity in period \( t \)
Capstone Case: Capacitated Lot Sizing

**Objective:** Minimize sum of marginal production cost, holding cost, fixed cost

\[
\sum_{i=1}^{N} \sum_{t=1}^{T} (c_{it}x_{it} + h_{it}l_{it} + f_{it}y_{it})
\]

**Constraint 1:** Do not exceed total capacity in each period

\[
\sum_{i=1}^{N} x_{it} \leq g_t, \quad t = 1, \ldots, T
\]
Capstone Case: Capacitated Lot Sizing

**Constraint 2:** Inventory balance equations

\[ I_{i,t-1} + x_{it} - I_{it} = d_{it}, \quad i = 1, \ldots, N, t = 1, \ldots, T \]

**Constraint 3:** Fixed cost forcing constraints

\[ x_{it} - M_{it} y_{it} \leq 0, \quad i = 1, \ldots N, t = 1, \ldots, T \]
Capstone Case: Capacitated Lot Sizing

Dynamic Lot Sizing: A “standard formulation” is:

$$\min \sum_{i=1}^{N} \sum_{t=1}^{T} (c_{it}x_{it} + h_{it}l_{it} + f_{it}y_{it})$$

s.t. $$\sum_{i=1}^{N} x_{it} \leq g_{t}, \quad t = 1, \ldots, T$$

$$l_{i,t-1} + x_{it} - l_{it} = d_{it}, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T$$

$$x_{it} - M_{it}y_{it} \leq 0, \quad i = 1, \ldots N, \quad t = 1, \ldots, T$$

$$x_{it}, l_{it} \geq 0, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T$$

$$y_{it} \in \{0, 1\}, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T.$$
Capstone Case: Capacitated Lot Sizing

The generic model is

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad Ax \geq b \\
& \quad x \in \Gamma
\end{align*}
\]

\((MIP)\)

Map the capacitated lot sizing model into this as follows: the capacity constraints \(\sum_{i=1}^{N} x_{it} \leq g_t\) map into the \(Ax \geq b\) constraints and the rest of the constraints

- inventory balance
- fixed charge big M
- nonnegativity and binary

define the \(\Gamma\).
Capstone Case: Capacitated Lot Sizing

Four approaches for generating bounds to find the optimal value of

$$\min c^T x$$

$$\text{conv}(\text{MIP}) \quad \text{s.t.} \quad Ax \geq b$$

$$x \in \text{conv}(\Gamma)$$

1. Column generation

2. Subgradient optimization to find optimal $L(\lambda)$

3. Generate the cuts that characterize $\text{conv}(\Gamma)$

4. The $z_{itk}$ formulation that also gives $\text{conv}(\Gamma)$ (via Homework 5, question 5)
Capstone Case: Capacitated Lot Sizing

**Dynamic Lot Sizing**: An alternate formulation:

$z_{itk}$ is 1, if for product $i$ in period $t$, the decision is to produce enough items to satisfy demand for periods $t$ through $k$, 0 otherwise.

\[
\sum_{k=1}^{T} z_{i1k} = 1
\]

\[
- \sum_{j=1}^{t-1} z_{ij,t-1} + \sum_{k=t}^{T} z_{itk} = 0, \quad i = 1, \ldots, N, \quad t = 2, \ldots, T
\]

\[
\sum_{k=t}^{T} z_{itk} \leq y_{it}, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T
\]
Capstone Case: Capacitated Lot Sizing

**Dynamic Lot Sizing**: Characterize $\text{conv}(\Gamma)$ through cutting planes. Barany, Van Roy, and Wolsey have characterized $\text{conv}(\Gamma)$ by the $(\ell, S)$ inequalities:

$$\sum_{t \in S} x_t \leq \sum_{\ell} \left( \sum_{k=t} d_k \right) y_t + l_{\ell}$$

for $\ell = 1, \ldots, T$ and $\forall S \subseteq \{1, \ldots, \ell\}$.

The separation algorithm (formulated as an LP/IP) is given $(\overline{x}, \overline{l}, \overline{y})$ and $\ell = \{1, \ldots, T\}$, solve:

$$\max \sum_{t=1}^\ell \overline{x}_t \alpha_t - \sum_{t=1}^\ell \left( \sum_{k=t} d_k \right) \overline{y}_t \alpha_t - \overline{l}_\ell$$

with $\alpha_t \in \{0, 1\}, \quad t = 1, \ldots, \ell$

and the $\alpha_t = 1$ index the elements in the set $S$. 
Capstone Case: Capacitated Lot Sizing

**Dynamic Lot Sizing**: An alternate formulation: use subgradient, Lagrangian duality to get the result. The Lagrangian problem to solve in order to generate the subgradient is

\[
\min \sum_{i=1}^{N} \sum_{t=1}^{T} ((c_{it} + \lambda_t) x_{it} + h_{it} l_{it} + f_{it} y_{it}) - \sum_{t=1}^{T} \lambda_t g_t
\]

s.t.

\[
l_{i,t-1} + x_{it} - l_{it} = d_{it}, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T
\]

\[
x_{it} - M_{it} y_{it} \leq 0, \quad i = 1, \ldots N, \quad t = 1, \ldots, T
\]

\[
x_{it}, l_{it} \geq 0, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T
\]

\[
y_{it} \in \{0, 1\}, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T.
\]

What is the economic interpretation of \(\lambda_t\) in the term \((c_{it} + \lambda_t)x_{it}\)?
Dynamic Lot Sizing: An alternate formulation: use column generation to get convex hull.

This is what we did with the TSP problem in the previous set of slides. See also Example 12.20 in the text.
Capstone Case: Capacitated Lot Sizing

Final Exam: Take the two product, five period data:

http://faculty.chicagobooth.edu/kipp.martin/root/htmls/coursework/36900/datafiles/lotsizeDataExam.txt

and use one of the following techniques to find the convex hull relaxation.

- the Barany, Van Roy, and Wolsey cuts
- subgradient optimization
- column generation

IMPORTANT: The final exam must be done on an individual basis. You are not allowed to collaborate in any way with your classmates.
Lagrangian Approach Application – Convex Optimization

Objective: Apply projection and the Lagrangian duality to convex optimization.

\[
\sup_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad g_i(x) \geq 0 \quad \text{for } i = 1, \ldots, p \quad \text{(CP)}
\]

where

- \( f(x) \) and \( g_i(x) \) for \( i = 1, \ldots, p \) are concave functions
- \( \Omega \) is a closed, convex set.
The Lagrangian function $L(\lambda)$ is

$$L(\lambda) := \sup \left\{ f(x) + \sum_{i=1}^{p} \lambda_i g_i(x) : x \in \Omega \right\}.$$ 

The Lagrangian dual is

$$\inf_{\lambda \geq 0} L(\lambda).$$

\textbf{Weak Duality:} $\inf_{\lambda \geq 0} L(\lambda) \geq \nu(CP)$ where $\nu(CP)$ is the optimal value of $(CP)$. 
Lagrangian Approach Application – Convex Optimization

Unfortunately, even for a convex problem, there may be a duality gap, that is

\[ \inf_{\lambda \geq 0} L(\lambda) > v(CP). \]

A great deal of research has gone into finding sufficient conditions that guarantee a zero duality gap.

These are often called constraint qualifications because they typically involve conditions on the constraints.

One of the most well-know constraint qualifications is the Slater constraint qualification.
The Slater constraint qualification is that there exists an \( x^* \in \Omega \) such that \( g_i(x^*) > 0 \) for all \( i = 1, \ldots, p \).

The Slater constraint qualification gives the following classic theorem in convex optimization.

**Theorem:** Suppose the convex program (CP) is feasible and bounded. Moreover, suppose there exists an \( x^* \in \Omega \) such that \( g_i(x^*) > 0 \) for all \( i = 1, \ldots, p \). Then there is zero duality gap between the convex program (CP) and its Lagrangian dual (11). Moreover, there exists a \( \lambda^* \geq 0 \) such that \( v(CP) = L(\lambda^*) \), i.e., the Lagrangian dual is solvable.

We prove this result by using projection (because, as you know by now, that is all I can do).
Lagrangian Approach Application – Convex Optimization

Construct the semi-infinite linear program (yes, linear)

\[
\inf \sigma \\
\text{s.t. } \sigma - \sum_{i=1}^{p} \lambda_i g_i(x) \geq f(x) \quad \text{for } x \in \Omega \quad \text{(CP-SILP)}
\]

This is equivalent to the Lagrangian dual

\[
\inf L(\lambda) \\
\text{s.t. } L(\lambda) = \sup \{f(x) + \sum_{i=1}^{p} \lambda_i g_i(x) : x \in \Omega\} \quad \text{(LD)}
\]

Replace

\[
L(\lambda) = \sup \{f(x) + \sum_{i=1}^{p} \lambda_i g_i(x) : x \in \Omega\}
\]

with

\[
L(\lambda) \geq f(x) + \sum_{i=1}^{p} \lambda_i g_i(x), \quad x \in \Omega.
\]
Theorem: Assume

- the convex program (CP) is feasible and bounded, and
- there exists a $x^* \in \Omega$ such that $g_i(x^*) > 0$ for all $i = 1, \ldots, p$

then

- there is a zero duality gap between the convex program (CP) and its Lagrangian dual (LD), and
- there exists a $\lambda^* \geq 0$ such that $\nu(\text{LD}) = L(\lambda^*)$, i.e., the Lagrangian dual is solvable.
**KEY IDEA:** Show that the semi-infinite linear program (CP-SILP) is solvable and has zero duality gap. Do this using, yes you guessed it, projection!

Put (CP-SILP) into our linear programming format and solve with Fourier-Motzkin elimination.

\[
egin{align*}
    z & - \sigma & \geq & 0 \\
    \sigma - \sum_{i=1}^{p} \lambda_i g_i(x) & \geq & f(x) & \forall x \in \Omega \\
    \lambda_i & \geq & 0 & i = 1, \ldots, p
\end{align*}
\]

Eliminate \(\sigma\) and end up with

\[
egin{align*}
    z - \sum_{i=1}^{p} \lambda_i g_i(x) & \geq & f(x) & \forall x \in \Omega \\
    \lambda_i & \geq & 0 & i = 1, \ldots, p
\end{align*}
\]
Claim: The variables $\lambda_1, \ldots, \lambda_p$ remain clean as the Fourier Motzkin elimination procedure proceeds.

Use induction to show that after eliminating variables $\lambda_1, \ldots, \lambda_k$, there is an inequality

$$z - \sum_{i=k+1}^{p} \lambda_i g_i(x^*) \geq f(x^*).$$

1. This implies variables $\lambda_i$, for $i = k + 1$, are clean Why?

2. If there are no dirty variables, there is no duality gap. Why?
We can also show:

\[ \nu(\text{CP-SILP}) = \nu(\text{LD}) \geq \nu(\text{CP}) = \nu(\text{CP-FDSILP}) \]

See http://www.optimization-online.org/DB_HTML/2013/04/3817.html.

We just showed

\[ \nu(\text{CP-SILP}) = \nu(\text{CP-FDSILP}) \]

therefore

\[ \nu(\text{LD}) = \nu(\text{CP}). \]
The Slater result can be strengthened.

**Theorem:** An instance of (CP-SILP) with finite optimal value is tidy if and only if the set of optimal solutions to (CP-SILP) is bounded.

See http://www.optimization-online.org/DB_HTML/2013/04/3817.html.