Integer Programming: Relaxations

Eduardo Camponogara

Department of Automation and Systems Engineering
Federal University of Santa Catarina

October 2016

Summary

Optimality Conditions

- **Optimality Conditions**: How do we assess the quality of a given candidate solution with respect the optimum, without knowing the optimum?
- This can be achieved by solving an easier problem to optimality, however within a space that includes the feasible space of the problem at hand.

Optimality Conditions

- **Optimality Conditions**: How do we assess the quality of a given candidate solution with respect the optimum, without knowing the optimum?
- This can be achieved by solving an easier problem to optimality, however within a space that includes the feasible space of the problem at hand.

Optimality Conditions

Relaxations can be obtained by:

- disconsidering some constraints;
- neglecting the integrality of discrete variables;
- solving a simplified problem.

Optimality Conditions

- Take the following Integer Program:

$$IP: \; z = \max \{c^{\mathrm{T}}x : x \in X \subseteq \mathbb{Z}^n\}$$

- Given a candidate solution $x^\star$, how can we prove that $x^\star$ is an optimal solution?
- We look for optimality conditions that provide a stopping condition for algorithms.

Optimality Conditions

- Take the following Integer Program:

$$\color{red}{IP}: \; z = \max \{c^{\mathrm{T}}x : x \in X \subseteq \mathbb{Z}^n\}$$

- Given a candidate solution $x^\star$, how can we prove that $x^\star$ is an optimal solution?
- We look for optimality conditions that provide a stopping condition for algorithms.

Optimality Conditions

- Take the following Integer Program:

$$IP: \; z = \max \{c^{\mathrm{T}} x : x \in X \subseteq \mathbb{Z}^n\}$$

- Given a candidate solution $x^\star$, how can we prove that $x^\star$ is an optimal solution?
- We look for optimality conditions that provide a stopping condition for algorithms.

Optimality Conditions

- One method consists of finding a lower bound $\underline{z} \leq z$ and an upper bound $\overline{z} \geq z$ such that $\underline{z} = \overline{z}$, in which case z is the optimum.
- Typically, an algorithm produce two sequences of bounds:
  - a sequence of upper bounds $\overline{z}_1 > \overline{z}_2 > \ldots > \overline{z}_t \geq z$
  - a sequence of lower bounds $\underline{z}_1 < \underline{z}_2 < \ldots < \underline{z}_k \leq z$.
- Given $\xi \geq 0$, we can define a stopping criterion such as $|\overline{z}_t - \underline{z}_k| \leq \xi$.

Optimality Conditions

- One method consists of finding a lower bound $\underline{z} \leq z$ and an upper bound $\overline{z} \geq z$ such that $\underline{z} = \overline{z}$, in which case z is the optimum.
- Typically, an algorithm produce two sequences of bounds:
  - a sequence of upper bounds $\overline{z}_1 > \overline{z}_2 > \ldots > \overline{z}_t \geq z$
  - a sequence of lower bounds $\underline{z}_1 < \underline{z}_2 < \ldots < \underline{z}_k \leq z$.
- Given $\xi \geq 0$, we can define a stopping criterion such as $|\overline{z}_t - \underline{z}_k| \leq \xi$.

Optimality Conditions

- One method consists of finding a lower bound $\underline{z} \leq z$ and an upper bound $\overline{z} \geq z$ such that $\underline{z} = \overline{z}$, in which case z is the optimum.
- Typically, an algorithm produce two sequences of bounds:
  - a sequence of upper bounds $\overline{z}_1 > \overline{z}_2 > \ldots > \overline{z}_t \geq z$
  - a sequence of lower bounds $\underline{z}_1 < \underline{z}_2 < \ldots < \underline{z}_k \leq z$.
- Given $\xi \geq 0$, we can define a stopping criterion such as $|\overline{z}_t - \underline{z}_k| \leq \xi$.

Optimality Conditions

- When an algorithm terminates, the conditions ascertain that the candidate solution z is at most ξ units worse than the optimum.
- The ability to estimate the quality of a candidate solution is of fundamental importance in optimization.

Optimality Conditions

- When an algorithm terminates, the conditions ascertain that the candidate solution z is at most ξ units worse than the optimum.
- The ability to estimate the quality of a candidate solution is of fundamental importance in optimization.

Primal Bound

Finding a feasible solution may be a hard or easy task. It is a problem dependent issue.

- Any feasible solution x, $x \in X$, induces a bound (lower bound for maximization) given that $c^{\mathrm{T}}x \leq c^{\mathrm{T}}x^{\star} = z^{\star}$.
- For NP-Complete problems, the search for a feasible solution corresponds to solving the problem.
- On the other hand, finding a feasible route for the traveling salesman probllem is easy, the hardness being in reaching the shortest route.

Primal Bound

Finding a feasible solution may be a hard or easy task. It is a problem dependent issue.

- Any feasible solution x, $x \in X$, induces a bound (lower bound for maximization) given that $c^{\mathrm{T}}x \leq c^{\mathrm{T}}x^{\star} = z^{\star}$.
- For NP-Complete problems, the search for a feasible solution corresponds to solving the problem.
- On the other hand, finding a feasible route for the traveling salesman probllem is easy, the hardness being in reaching the shortest route.

Primal Bound

Finding a feasible solution may be a hard or easy task. It is a problem dependent issue.

- Any feasible solution x, $x \in X$, induces a bound (lower bound for maximization) given that $c^{\mathrm{T}}x \leq c^{\mathrm{T}}x^{\star} = z^{\star}$.
- For NP-Complete problems, the search for a feasible solution corresponds to solving the problem.
- On the other hand, finding a feasible route for the traveling salesman probllem is easy, the hardness being in reaching the shortest route.

Primal Bound

Finding a feasible solution may be a hard or easy task. It is a problem dependent issue.

- Any feasible solution x, $x \in X$, induces a bound (lower bound for maximization) given that $c^{\mathrm{T}}x \leq c^{\mathrm{T}}x^{\star} = z^{\star}$.
- For NP-Complete problems, the search for a feasible solution corresponds to solving the problem.
- On the other hand, finding a feasible route for the traveling salesman probllem is easy, the hardness being in reaching the shortest route.

Dual Bound

- A method to find an upper bound relies on a relaxation of the problem at hand.
- That is, a simpler problem whose optimal solution has an objective value not inferior to the optimum of the problem of concern.

Dual Bound

Definition

A problem R:

$$z^{\mathrm{R}} = \max \{f(x) : x \in T \subseteq \mathbb{R}^n\}$$

is a relaxation of the problem IP:

$$z = \max \{c(x) : x \in X \subseteq \mathbb{R}^n\}$$

if:

i) $X \subseteq T$

ii) $f(x) \geq c(x)$ for all $x \in X$.

Dual Bound

Proposition

IF (R) is a relaxation of (IP) then $z^{\mathrm{R}} \geq z$.

Summary

Linear Relaxation

Linear Relaxation

For the integer programming problem:

$$\max \{c^{\mathrm{T}}x : x \in P \cap \mathbb{Z}^n\}$$

with formulation:

$$P = \{x \in \mathbb{R}^n_+ : Ax \leq b\}$$

the linear relaxation is:

$$z^{\mathrm{PL}} = \max \{c^{\mathrm{T}}x : x \in P\}$$

Linear Relaxation

- Example

$$
\begin{array}{rlrrcr}
z = & \max & 4x_1 & -x_2 & & \\
& \text{s.t.}: & 7x_1 & -2x_2 & \leq & 14 \\
& & & x_2 & \leq & 3 \\
& & 2x_1 & -2x_2 & \leq & 3
\end{array}
$$

where $x \in \mathbb{Z}_+^2$.

- Lower bound: notice that $\underline{x} = (2, 1)$ is feasible, therefore $z \geq 7$.

Linear Relaxation

- Example

$$\begin{array}{rlrrcr} z = & \max & 4x_1 & -x_2 & & \\ & \text{s.t.}: & 7x_1 & -2x_2 & \leq & 14 \\ & & & x_2 & \leq & 3 \\ & & 2x_1 & -2x_2 & \leq & 3 \end{array}$$

  where $x \in \mathbb{Z}^2_+$.

- Lower bound: notice that $\underline{x} = (2, 1)$ is feasible, therefore $z \geq 7$.

Linear Relaxation

- Example

$$
\begin{array}{rlrrcr}
z = & \max & 4x_1 & -x_2 & & \\
& \text{s.t.}: & 7x_1 & -2x_2 & \leq & 14 \\
& & & x_2 & \leq & 3 \\
& & 2x_1 & -2x_2 & \leq & 3
\end{array}
$$

where $x \in \mathbb{Z}_+^2$.

- Upper bound: an optimal solution to the linear relaxation $x^\star = \left(\frac{20}{7}, 3\right)$ with $z^{\mathrm{LP}} = \frac{59}{7}$, thus one concludes that $z \leq 8 \leq \frac{59}{7}$, in which $8 = \lfloor \frac{59}{7} \rfloor$.

Linear Relaxation

- Example

$$
\begin{array}{rrrrrl}
z = & \max & 4x_1 & -x_2 & & \\
& \text{s.t.}: & 7x_1 & -2x_2 & \leq & 14 \\
& & & x_2 & \leq & 3 \\
& & 2x_1 & -2x_2 & \leq & 3
\end{array}
$$

where $x \in \mathbb{Z}^2_+$.

- Upper bound: an optimal solution to the linear relaxation $x^\star = \left(\frac{20}{7}, 3\right)$ with $z^{\mathrm{LP}} = \frac{59}{7}$, thus one concludes that $z \leq 8 \leq \frac{59}{7}$, in which $8 = \lfloor \frac{59}{7} \rfloor$.

Combinatorial Relaxation

- When the relaxation consists of a combinatorial problem, the relaxation is said to be a combinatorial relaxation.
- An example of a combinatorial relaxation for the traveling salesman problem and the knapsack problem are given below.

The Traveling Salesman Problem (TSP)

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

$$\text{s.t. :} \quad \sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,..,n \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,..,n \tag{3}$$

$$\sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1 \qquad \forall S \subset \{1,\dots,n\}, |S| \geq 2 \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{5}$$

The Traveling Salesman Problem

- Discarding the family (4) of constraints, the allocation problem is obtained.
- Notice that the allocation problem can be solved efficiently.

The Knapsack Problem

$$
\begin{aligned}
(KP) \quad \max \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.}: \quad & \sum_{j=1}^{n} a_j x_j \leq b \\
& x_j \in \{0,1\}, j = 1, \ldots, n
\end{aligned}
$$

The Knapsack Problem

A first relaxation is obtained if we take the integers immediately inferior to the weights of the items:

$$
\begin{aligned}
(RP_1) \quad \max \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.}: \quad & \sum_{j=1}^{n} \lfloor a_j \rfloor x_j \leq b \\
& x_j \in \{0,1\}, j = 1, \ldots, n
\end{aligned}
\tag{6}
$$

The Knapsack Problem

The above relaxation is equivalent to the problem:

$$
\begin{aligned}
(RP_2) \quad \max \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.}: \quad & \sum_{j=1}^{n} \lfloor a_j \rfloor x_j \leq \lfloor b \rfloor \\
& x_j \in \{0,1\}, j = 1, \ldots, n
\end{aligned}
\tag{7}
$$

Summary

Introduction

Relaxations

Lagrangian Relaxation

Lagrangian Relaxation

First, consider the general integer program given by:

$$(P) \quad z = \max \; c^{\mathrm{T}}x \\ \text{s.t.}: \; Ax \leq b \\ x \in \mathcal{X} \subseteq \mathbb{Z}^n$$

As seen above, a relaxation is obtained if we neglect the constraints $\{Ax \leq b\}$.

Lagrangian Relaxation

Rather than discarding $\{Ax \leq b\}$, the Lagrangian relaxation introduces these constraints in the objective:

$$
\begin{aligned}
LR(u) \quad z = \quad & \max \; c^{\mathrm{T}} x + u^{\mathrm{T}}(b - Ax) \\
& \text{s.t.}: \quad x \in \mathcal{X} \\
& \qquad\quad\; u \geq 0
\end{aligned}
$$

Lagrangian Relaxation

Proposition

- Let $z(u) = \max \{c^{\mathrm{T}}x + u^{\mathrm{T}}(b - Ax) : x \in X\}$.
- Then $z(u) \geq z$ for all $u \geq 0$.

Lagrangian Relaxation: Remarks

- Given $u \geq 0$, the optimal solution to (LR) induces an upper bound $z(u) \geq z^*$.
- We can verify that $z(u) \geq c^{\mathrm{T}}x$, for all $x \in \mathcal{P} = \{x : x \in X \text{ e } Ax \leq b\}$:
  - $c^{\mathrm{T}}x \leq c^{\mathrm{T}}x + u^{\mathrm{T}}(b - Ax)$ since $b - Ax \geq 0$, x is feasible, and $u \geq 0$;
  - Since the feasible solution to the integer program (IP) is also within the feasible space of LR, we conclude that $z(u) \geq z^*$.

Lagrangian Relaxation: Remarks

- Given $u \geq 0$, the optimal solution to (LR) induces an upper bound $z(u) \geq z^*$.
- We can verify that $z(u) \geq c^{\mathrm{T}}x$, for all $x \in \mathcal{P} = \{x : x \in X \text{ e } Ax \leq b\}$:
  - $c^{\mathrm{T}}x \leq c^{\mathrm{T}}x + u^{\mathrm{T}}(b - Ax)$ since $b - Ax \geq 0$, x is feasible, and $u \geq 0$;
  - Since the feasible solution to the integer program (IP) is also within the feasible space of LR, we conclude that $z(u) \geq z^*$.

Convexity of Dual Function

Take $u, v \geq 0$ and $\theta \in [0, 1]$. Then

$$\begin{aligned}
z(\theta u + (1-\theta)v) &= \max_{x \in \mathcal{X}} c^{\mathrm{T}} x + (\theta u + (1-\theta)v)^{\mathrm{T}}(b - Ax) \\
&= \max_{x \in \mathcal{X}} c^{\mathrm{T}}(\theta + (1-\theta))x + (\theta u + (1-\theta)v)^{\mathrm{T}}(b - Ax) \\
&= \max_{x \in \mathcal{X}} \theta[c^{\mathrm{T}} x + u^{\mathrm{T}}(b - Ax)] + (1-\theta)[c^{\mathrm{T}} x + v^{\mathrm{T}}(b - Ax)] \\
&\leq \theta \cdot \max_{x \in \mathcal{X}} [c^{\mathrm{T}} x + u^{\mathrm{T}}(b - Ax)] + (1-\theta) \cdot \max_{x \in \mathcal{X}} [c^{\mathrm{T}} x + v^{\mathrm{T}}(b - Ax)] \\
&= \theta z(u) + (1-\theta) z(v)
\end{aligned}$$

Lagrangian Dual

Since $z(u)$ is an upper bound for z^*, naturally one would wish to minimize the upper bound, giving rise to the Lagrangian Dual:

$$LD: \quad \min_{u \geq 0} \; z(u) \;\cong\; \min_{u \geq 0} \; \begin{array}{ll} \max & c^{\mathrm{T}}x + u^{\mathrm{T}}(b - Ax) \\ \text{s.t.}: & x \in \mathcal{X} \end{array}$$

Lagrangian Relaxation

Lagrangian Dual

- The Lagrangian dual is convex but not a differentiable problem.
- An approximate solution for LD could be sought.
- Under certain conditions, an optimal solution can be found using the subgradient algorithm.

Subgradient

Definition

Given a function $z : \mathbb{R}^n \to \mathbb{R}$, a vector $d(u) \in \mathbb{R}^n$ is a subgradient for z at $u \in \mathcal{U}$ if:

$$z(v) \geq z(u) + d(u)^{\mathrm{T}}(v - u), \forall v \in \mathcal{U}$$

in which $\mathcal{U}$ is the domain of z.

Subgradient

- Take $u \geq 0$ for LD, and let $x(u) \in \mathcal{X}$ be the point that induces the value $z(u)$.
- Then, a subgradient for z at u is given by:

$$d(u) = b - Ax(u)$$

because $z(u)$ is a convex function in u.

Subgradient

- Take $u \geq 0$ for LD, and let $x(u) \in \mathcal{X}$ be the point that induces the value $z(u)$.
- Then, a subgradient for z at u is given by:

$$d(u) = b - Ax(u)$$

because $z(u)$ is a convex function in u.

Subgradient Algorithm

1. Input parameters: $u_0 \geq 0$, μ_0, and $\rho < 1$.
2. Let $k = 0$.
3. Compute $z(u_k)$ and obtain $x(u_k)$ by solving $LR(u_k)$.
4. If $x(u_k)$ is feasible for IP, or stopping criteria is satisfied, then halt.
5. Let $d_k = b - Ax(u_k)$ be the subgradient.
6. $u_{k+1} = u_k - \mu_k d_k$.
7. $\mu_{k+1} = \rho \mu_k$, $k = k + 1$, and repeat from step 3.

Lagrangian Dual as an LP Program

Example: Consider the following instance of the knapsack problem:

$$
\begin{aligned}
z = \max \quad & 10x_1 + 7x_2 + 25x_3 + 24x_4 \\
\text{s.t.}: \ & 2x_1 + x_2 + 6x_3 + 5x_4 \leq 7 \\
& x_1, x_2, x_3, x_4 \in \{0, 1\}
\end{aligned}
$$

Challenge: develop an LP program equivalent to the Lagrangian Dual.

Lagrangian Dual as an LP Program

Example: Consider the following instance of the knapsack problem:

$$\begin{aligned} z = \max \;\; & 10x_1 + 7x_2 + 25x_3 + 24x_4 \\ \text{s.t.} : \; & 2x_1 + x_2 + 6x_3 + 5x_4 \leq 7 \\ & x_1, x_2, x_3, x_4 \in \{0, 1\} \end{aligned}$$

Challenge: develop an LP program equivalent to the Lagrangian Dual.

Integer Programming: Relaxations

- Thank you for attending this lecture!!!