Lagrangean Duality

Jesper Larsen
Informatics and Mathematical Modelling
Technical University of Denmark
2800 Kgs. Lyngby – Denmark
Email: jla@imm.dtu.dk

Relaxation

- A problem (RP) $z^R = \max\{f(x) : x \in T \subseteq R^n\}$ is a **relaxation** of (IP) $z = \max\{c(x) : x \in X \subseteq R^n\}$ if:
  - *(i)* $X \subseteq T$
  - *(ii)* for all $x \in X$: $c(x) \leq f(x)$

Introduction

Lagrangean relaxation is a technique which has been known for many years.

- The technique has been very useful in conjuction with Branch and Bound
- Since the early 70's it has emerged as **the** bounding technique
- Has also served as the basis for the development of heuristics (dual ascent) and variable fixing.

Lagrangian Relaxation

Consider an integer programming problem:

$$\begin{array}{lll} \max & cx & \\ & Dx & \leq & d \\ & x & \in & X \end{array}$$

where $x \in X$ equals $x \in \{x : Ax \leq b, x \text{ integer}\}$ for our regular integer programming problem. Now assume that if we dropped $Dx \leq d$ the problem

$$\begin{array}{lll} \max & cx & \\ & x & \in & X \end{array}$$

would be "easy".

Now go one step further and add a penalty term (to the objective function) that is "active" when $Dx \leq d$ is violated, that is,

$$\begin{aligned} \max \quad & cx + u(d - Dx) \\ & x \in X \end{aligned}$$

where $u \geq 0$.

$z(u) = \max\{cx + u(d - Dx) : x \in X\}$ is called the **Lagrangian relaxation** of $z = \max\{cx : Dx \leq d, x \in X\}$.

Notation

The lagrangian relaxation is often denoted *IP(u)* which is

$$z(u) = \max_{x \in X}\{cx + u(d - Dx)\}$$

Proposition: For $u \geq 0$ IP(u) is a relaxation of IP (the original integer programming problem).

Proof: (i) Feasible region enlarged, and (ii) objective function pointwise larger on all feasible x.

Hence for $u \geq 0$ IP(u) provides a dual (upper) bound.

Next logical step. As IP(u) provides a dual (upper) bound for $u \geq 0$ let us look for the best one:

$$\begin{aligned} w_{\mathsf{LD}} &= \min_{u \geq 0}\{z(u)\} \\ &= \min_{u \geq 0}\{\max_{x \in X}\{cx + u(d - Dx)\}\} \end{aligned}$$

Central question: Best u? When does LD solve the original?

Proposition: If $u \geq 0$ and

1. $x(u)$ is an optimal solution of IP(u)
2. $Dx \leq d$
3. $Dx(u)_i = d_i$ whenever $u_i > 0$

then $x(u)$ is optimal in IP.

Issues

There are two issues that needs to be discussed when using Lagrangean relaxation:

- Which constraints to relax?
- How to find Lagrangean multipliers?

- Ideally the optimal value of the Lagrangean dual program is equal to the optimal value of the original integer program.
- If the two programs do not have optimal values which are equal then a **duality gap** is said to exist, the size of which is measured by the relative difference between the two optimal values.
- Eg. in the case of weak duality there might be a gap between the two solutions.

- The size of the gap is a good indicator of the difficulty of a problem.
- As a rule of thumb problems with a gap of more than 5-10% are too difficult to solve in practice.
- Note that in most cases we only have an estimate of the gap as we do not know the exact value of the optimal solution.

Lagrangean decomposition

Consider the following problem:

$$\begin{aligned} \min \quad & cx \\ & Ax \leq b \\ & Dx \leq d \\ & x \in B \end{aligned}$$

Now we introduce a set of variables y and set them equal to x. We can now use them in our second set of constraints and get.

$$\begin{aligned} \min \quad & cx \\ & x = y \\ & Ax \leq b \\ & Dy \leq d \\ & x \in B \\ & y \in B \end{aligned}$$

The original problem and the transformed problem are equivalent. NOW let us relax the constraints linking x and y together by introducing a Lagrangean multiplier vector λ. We then get

$$
\begin{aligned}
\min \quad & cx + \lambda(x - y) \\
& Ax \leq b \\
& Dy \leq d \\
& x \in B \\
& y \in B
\end{aligned}
$$

and now our problem is separable into the sum of the two programs:

$$\begin{array}{llcll} \min & (c+\lambda)x & \text{and} & \min & -\lambda y \\ & Ax \le b & & & Dy \le d \\ & x \in B & & & y \in B \end{array}$$

The sum of the solutions to these two programs provides a lower bound on the optimal solution to the original problem.

Solving the Lagrangean Dual

For simplicity assume that the set X contains a very large but finite number of points $\{x^1, x^2, \ldots, x^T\}$.

$$
\begin{aligned}
w_{LD} &= \min_{u \leq 0} z(u) \\
&= \min_{u \leq 0} \{\max_{x \in X} [cx + u(d - Dx)]\} \\
&= \min_{u \leq 0} \{\max_{t=1,2,\ldots,T} [cx^t + u(d - Dx^t)]\} \\
&= \min \eta \\
& \quad \eta \geq cx^t + u(d - Dx^t) \text{ for all } t \\
& \quad u \in R^m_+, \eta \in R^1
\end{aligned}
$$

The latter problem is a linear programming problem. Taking its dual gives:

$$
\begin{aligned}
w_{LD} = \quad \max \quad & \textstyle\sum_{t=1}^{T} \mu_t (cx^t) \\
& \textstyle\sum_{t=1}^{T} \mu_t (Dx^t - d) \leq 0 \\
& \textstyle\sum_{t=1}^{T} \mu_t = 1 \\
& \mu \in R_+^T
\end{aligned}
$$

Now if we set $x = \sum_{t=1}^{T} \mu_t x^t$ we get:

$$\begin{aligned} w_{LD} = \quad \max \quad & cx \\ & Dx \leq d \\ & x \in \mathsf{conv}(X) \end{aligned}$$

This result can also be shown in the more general case where X is the feasibe region of any integer program.

Theorem: $w_{LD} = \max\{cx : Dx \leq d, x \in \mathsf{conv}(X)\}$.

Structure of LD

- Minimize piecewise linear convex function – non-differential
- Subgradient of convex function: $f : R^m \rightarrow R$
  - subgradient at u:

$$\begin{aligned} \gamma(u) &\in R^m \\ f(v) &\geq f(u) + \gamma(u)^T(v - u) \end{aligned}$$

  - Note: if f is differentiable, then only one subgradient exists: *the gradient*.

Subgradient Algorithm

1. Choose initial Lagrange multipliers u^0, set $t = 0$
2. Solve the Lagrangean subproblem IP(u^t)
3. Calculate the current violation of the complicated constraints $s = d - Dx(u^t)$
4. $u^{t+1} = u^t + \mu^t \frac{s}{\|s\|}$, μ^t is the step size
5. $t := t + 1$

The algorithm is guaranteed to converge to the optimal solution as long as $\{\mu^t\}_{t=0}^{\infty} \rightarrow 0$ and $\sum_{t=0}^{\infty} \mu^t \rightarrow \infty$.

Subgradient Algorithm - specialization

1. Choose initial Lagrange multiplies u^0, set $t = 0$
2. Define $0 < \pi \leq 2$
3. Solve the Lagrangean subproblem IP(u^t)
4. Calculate the current violation of the complicated constraints $s = d - Dx(u^t)$
5. Calculate $T = \frac{\pi(z_{UB} - z(u))}{\sum s_i^2}$
6. $u_i^{t+1} = max\{0, u_i^t + Ts_i\}$
7. $t := t + 1$

Subgradient applied to Setcover

Let us take our set covering example from earlier.

- Set $\pi = 2$, $u = (4, 4, 3, 3)$.
- Solve IP(u): $x_1 = x_2 = 1, x_3 = x_4 = x_5 = x_6 = 0$ and $z(u) = 12$
- Compute s: $s_1 = 1 - x_1 - x_3 - x_6 = 0$, $s_2 = 1 - x_2 - x_4 - x_5 = 0$, $s_3 = 1 - x_1 - x_2 - x_3 = -1$, $s_4 = 1 - x_3 - x_5 = 1$
- $T = \frac{2(z_{UB} - z_{LB})}{\sum s_i} = \frac{2(22-12)}{2} = 10$

- Update the u's: $u_1 = max\{0, 4 + 10 \cdot 0\} = 4$,
  $u_2 = max\{0, 4 + 10 \cdot 0\} = 4$,
  $u_3 = max\{0, 3 + 10 \cdot (-1)\} = 0$,
  $u_1 = max\{0, 3 + 10 \cdot 1\} = 13$,

If we recompute $z(u)$ now with the updated u's we see that we get 6, which is a worse lower bound. The subgradient does namely not promise improvement in **every** step.