Lagrangian Relaxation: An overview

Sources for this lecture:

- D. Bertsimas and J. Tsitsiklis: Introduction to Linear Optimization, Athena Scientific, 1997
- M. L. Fisher. An applications oriented guide to Lagrangian relaxation. *Interfaces*, 15:1021, 1985.

General idea

Lagrangian relaxation is a technique well suited for problems where the constraints can be divided into two sets:

- "good" constraints, with which the problem is solvable very easily
- "bad" constraints that make it very hard to solve.

The main idea is to relax the problem by removing the "bad" constraints and putting them into the objective function, assigned with weights (the *Lagrangian multiplier*). Each weight represents a penalty which is added to a solution that does not satisfy the particular constraint.

We are given the following *integer linear problem*:

$$Z_{\text{IP}} := \min c^T x \tag{4.1}$$
$$Ax \geq b \tag{4.2}$$
$$Dx \geq d \tag{4.3}$$
$$x \text{ integer} \tag{4.4}$$

with A, D, b, c, d having integer entries. Let

$$X := \{x \text{ integral } \mid Dx \geq d\} \ .$$

We assume that optimizing over the set X can be done very easily, whereas adding the "bad" constraints $Ax \geq b$ makes the problem intractable.

Therefore, we introduce a dual variable for *every* constraint of $Ax \geq b$. The vector $\lambda \geq 0$ is the vector of dual variables (the *Lagrangian multipliers*) that has the same dimension as vector b. For a fixed $\lambda \geq 0$, consider the *relaxed problem*

$$\begin{aligned} Z(\lambda) := \min\ & c^T x + \lambda^T (b - Ax) \\ & Dx \geq d \\ & x \text{ integer} \ . \end{aligned}$$

By assumption, we can efficiently compute the optimal value for the relaxed problem with a fixed vector λ.

Lemma (Weak duality).
$Z(\lambda)$ provides a lower bound on Z_{IP}.

Proof. Blackboard.

Of particular interest is the tightest of all bounds, that is

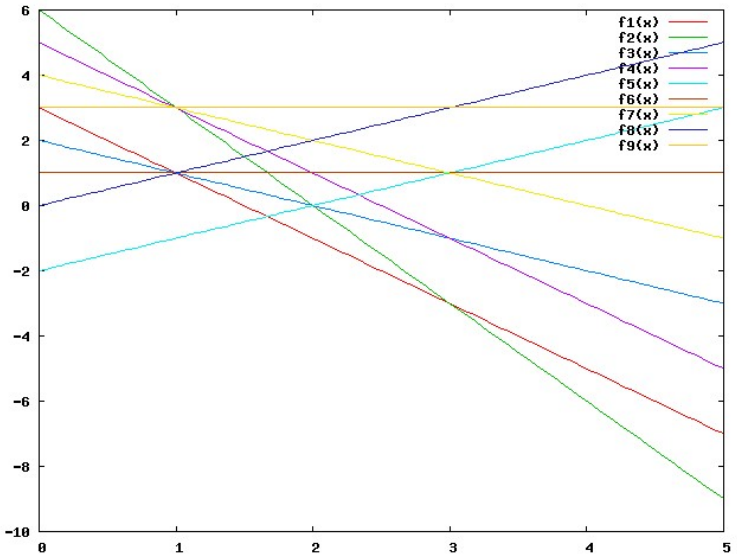
$$Z_D := \max_{\lambda \geq 0} Z(\lambda) \ .$$

The problem above is called the *Lagrangian dual*.

If X is a finite set, say $X = \{x^1, \ldots, x^m\}$, then $Z(\lambda)$ can also be written as

$$Z(\lambda) = \min_{i=1,\ldots,m} \{c^T x^i + \lambda^T (b - Ax^i)\} \ .$$

The function $Z(\lambda)$ is the minimum of a finite set of linear functions of λ and therefore it is concave and piecewise linear.



It is important to note, however, that—unlike in linear programming—integer linear programming does not have strong duality theory. This implies that the optimal value of the Lagrangian dual does not have to be the same as the optimal value of the original (primal) problem. Instead of

$$Z_D = Z_{IP}$$

the following holds:

$$Z_D \leq Z_{IP} \ .$$

In general we have $Z_{LP} \leq Z_D \leq Z_{IP}$, where Z_{LP} is the solution value of the LP relaxation.

We can illustrate this nicely using the following theorem which characterizes the Lagrangean dual as a *linear program* !

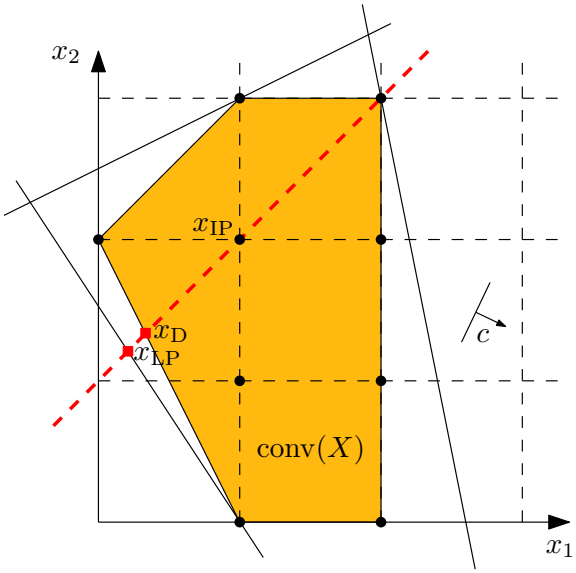
Theorem
The optimal value Z_D of the langrangean dual is equal to the optimal cost of the following LP:

$$\begin{aligned} \min\ & c^T x \\ & Ax \geq b \\ & x \in CH(X) \ . \end{aligned}$$

(proof is in Bertsimas).

A good time to see an example... (blackboard)

Figure Example Lagrange



Solving the Lagrangian Dual

How should the λ be set, such that the gap between $Z(\lambda)$ and Z_{IP} is as small as possible (or zero in the best case)?

For sake of simplicity, we assume that X is finite and can be written as $X = \{x^1, \ldots, x^m\}$. Then, as described above, $Z(\lambda)$ can be written as

$$Z(\lambda) = \min_{i=1,\ldots,m} \{c^T x^i + \lambda^T (b - Ax^i)\}$$

with $f_i = b - Ax^i$ and $h_i = c^T x^i$ this can be rewritten as

$$Z(\lambda) = \min_{i=1,\ldots,m} \{h_i + \lambda f_i^T\} \ ,$$

a piecewise linear and concave function.

If $Z(\lambda)$ was differentiable then, the classical approach of maximizing the function would be the *steepest ascent method*, that is computing a sequence of iterations with

$$\lambda^{t+1} = \lambda^t + \gamma^t \nabla Z(\lambda^t) \ .$$

We are following the gradient at the current position, with a specified stepsize γ, to reach points with a higher function value.

In our case, this evaluates to

$$\lambda^{t+1} = \lambda^t + \gamma^t (b - Ax^t) \ .$$

Unfortunately, this procedure is no longer valid for our function, since it is not differentiable everywhere.

Therefore, we adapt the method at points where the function is non-differentiable $\rightarrow$ *subgradient optimization*.

We use the following lemma:

Lemma 1. *A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is concave iff for any $x^* \in \mathbb{R}^n$, there exists a vector $s \in \Re^n$ such that*

$$f(x) \leq f(x^*) + s^t (x - x^*)$$

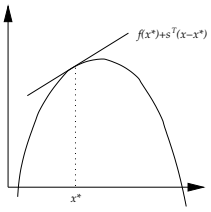
holds $\forall x \in \Re^n$.

Using the alternative notion of concavity functions, the notion of a *gradient* can be extended as follows:

Definition 2. Let f be a concave function. A vector s such that

$$f(x) \leq f(x^*) + s^t(x - x^*)$$

for all $x \in \Re^n$, is called a *subgradient* of function f at point x^*. The set of all *subgradients* of f at x^* is denoted by $\partial f(x^*)$ and is called the *subdifferential* of f at x^*.



As soon as $\mathbf{0} \in \partial f(x^*)$, we are done, as the following lemma states:

Lemma 3. *Let $f : \Re^n \to \Re$ be a concave function. A vector x^* maximizes f over $\Re^n$ iff $0 \in \partial f(x^*)$.*

Lemma **??** follows directly from Lemma **??**. If $(s = 0) \in \partial f(x^*)$, it implies that

$$f(x) \leq f(x^*)$$

which describes the maximum of function f.

The algorithm can be written as:

Subgradient optimization method.

1. Choose a starting point λ^0, e. g., $\lambda^0 = 0$; $t = 0$.
2. Choose a subgradient $s^t = b - Ax^t$ of the function Z at λ^t. If $s^t = \mathbf{0} \to$ **STOP**, because the optimal value has been reached.
3. Compute $\lambda^{t+1} = \max\{0, \lambda^t + \gamma^t s^t\}$, where γ^t denotes the stepsize[1].
4. Increment t and go to 2.

The definition of stepsize γ is of crucial importance, since the speed of convergence depends heavily on the stepsize. One can prove that the process converges to Z_D—assuming it is finite—for any stepsize γ^t with

$$\sum_{t=0}^{\infty} \gamma^t = \infty$$

and

$$\lim_{t \to \infty} \gamma^t = 0 \ .$$

One example for such a sequence is $\gamma^t = \frac{1}{t+1}$. In practice this does not always lead to quick convergence, hence other stepsize sequences are chosen.

Held and Karp proposed the following formula for adapting the stepsize:

$$\gamma^t = \mu^t \frac{Z^* - Z(\lambda^t)}{\sum_{i=1}^{m} (b_i - \sum_{j=1}^{n} a_{ij} x^t)^2} \ ,$$

where

- Z^* is the value of the best solution for the original problem found so far

[1] We need to cut off at zero because the multipliers have to remain non-negative.

- μ^t is a decreasing adaption parameter with $0 < \mu^0 \leq 2$ and

$$\mu^{t+1} = \begin{cases} \alpha\mu^t & Z_D \text{ did not increase in the last } T \text{ iterations} \\ \mu^t & \text{otherwise} \end{cases}$$

  with parameters $0 < \alpha < 1$ and $T > 1$.
- the denominator is the square of the length of the subgradient vector $b - Ax^t$.