Chapter 12

Lagrangian Relaxation

This chapter is mostly inspired by Chapter 16 of [1].

In the previous chapters, we have succeeded to find efficient algorithms to solve several important problems such as SHORTHEST PATHS, NETWORK FLOWS. But, as we have seen, most of practical graph or network problems are $\mathcal{NP}$-complete and hard to solve. In such a case, it may be interesting to solve a simplified problem to obtain approximations or bounds on the initial hardest problem. Consider the following optimization problem where $f : \mathbb{R}^n \to \mathbb{R}$ and $S \subseteq \mathbb{R}^n$:

$$\begin{array}{ll} \text{Minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in S \end{array}$$

A relaxation of the above problem has the following form:

$$\begin{array}{ll} \text{Minimize} & f_R(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in S_R \end{array}$$

where $f_R : \mathbb{R}^n \to \mathbb{R}$ is such that $f_R(\mathbf{x}) \leq f(\mathbf{x})$ for any $\mathbf{x} \in S$ and $S \subseteq S_R$. It is clear that the optimal solution f_R^* of the relaxation is a lower bound of the optimal solution of the initial problem. In previous section, the considered problems are such that $S = X \cap \{0,1\}^n$ where $X \subseteq \mathbb{R}^n$ (or $X \subseteq \mathbb{Q}^n$) and the fractional relaxation corresponds to consider $f_R = f$ and $S_R = X$.

A large number of these problems have an underlying network structure. The idea of the *Lagrangian Relaxation* is to try to use the underlying network structure of these problems in order to use these efficient algorithms. The Lagrangian Relaxation is a method of *decomposition*: the constraints $S = S_1 \cup S_2$ of the problems are separated into two groups, namely the'easy' constraints S_1 and the'hard' constraints S_2. The hard constraints are then removed, i.e., $S_R = S_1$ and transferred into the objective function, i.e., f_R depends on f and S_2.

Since S_R is a set of'easy' constraints, it will be possible to solve the relaxation problem. Moreover, the interest of the Lagrangian relaxation is that, in some cases, the optimal solution of the relaxed problem actually gives the optimal solution of the initial problem.

We first illustrate the method on a classical example.

12.1 Constrained Shortest Path.

Consider a digraph $D = (V, E)$ with a source s and a sink t. To each arc $e \in E$, we associate two values: c_e the cost of the arc and and t_e the time necessary to take the arc. See Figure 12.1-(a) where $s = v_1$ and $t = v_6$.. The CONSTRAINED SHORTEST PATH problem consists in finding an (s,t)-path of minimum cost, under the additional constraint that the path requires at most T units of time to traverse. Such problems arises frequently in practice since in many contexts a company (e.g. a package delivery firm) wants to provide its services at the lowest possible cost and yet ensure a certain level of service to its customers (as embodied in the time restriction).

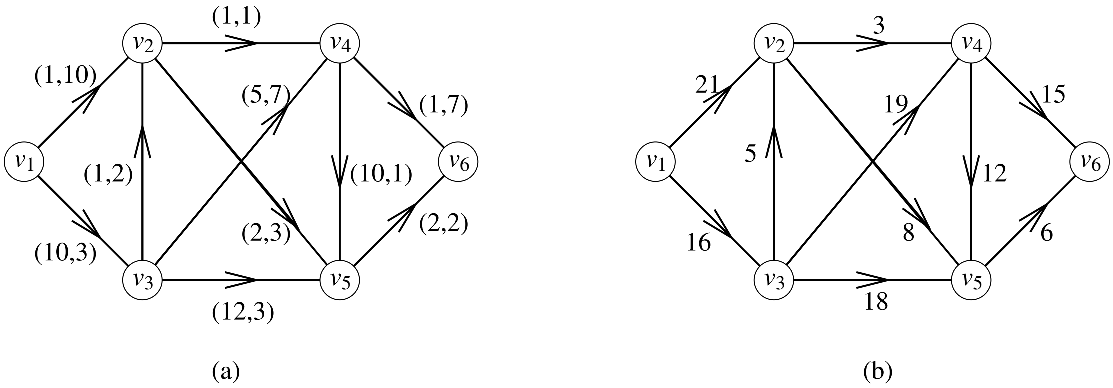

Figure 12.1: (a) An instance of CONSTRAINED SHORTEST PATH, and (b) its Lagrangian relaxation for $\mu = 2$. Next to each arc e appears the pair (c_e, t_e) in (a) and the value $c_e + \mu t_e$ in (b).

CONSTRAINED SHORTEST PATH can be formulated as the following integer linear programme, with $\mathbf{x}$ the indicator vector of the path.

$$
\begin{array}{ll}
\text{Minimize} & \sum_{e \in E} c_e x_e \\
\text{subject to} & \\
& \displaystyle\sum_{e \text{ entering } v} x_e - \sum_{e \text{ leaving } v} x_e = \begin{cases} -1 & \text{if } v = s \\ +1 & \text{if } v = t \\ 0 & \text{for all } v \in V \setminus \{s,t\} \end{cases} \\
& \displaystyle\sum_{e \in E} t_e x_e \le T \\
& x_e \in \{0,1\} \qquad \forall e \in E
\end{array}
\tag{12.1}
$$

(12.1) can clearly be decomposed into a classical (and easily solvable) shortest path problem plus an extra 'time' constraint. The idea of the Lagrangian Relaxation is to include this extra constraint as a penalty in the objective function.

More precisely, let $\mu > 0$ and consider the following integer linear programme.

$$\begin{array}{ll} \text{Minimize} & \sum_{e\in E} c_e x_e - \mu(T - \sum_{e\in E} t_e x_e) \\ \text{subject to} & \end{array}$$

$$\sum_{e \text{ entering } v} x_e - \sum_{e \text{ leaving } v} x_e = \begin{cases} -1 & \text{if } v = s \\ +1 & \text{if } v = t \\ 0 & \text{for all } v \in V \setminus \{s,t\} \end{cases} \tag{12.2}$$

$$x_e \in \{0,1\} \qquad \forall e \in E$$

The programme (12.2) is equivalent to finding a shortest (s,t)-path in D with the modified cost function $\mathbf{c}_\mu = \mathbf{c} + \mathbf{t}$. See Figure 12.1-(b).

$$\begin{array}{ll} \text{Minimize} & \sum_{e\in E} (c_e + \mu \cdot t_e) x_e \\ \text{subject to} & \end{array}$$

$$\sum_{e \text{ entering } v} x_e - \sum_{e \text{ leaving } v} x_e = \begin{cases} -1 & \text{if } v = s \\ +1 & \text{if } v = t \\ 0 & \text{for all } v \in V \setminus \{s,t\} \end{cases}$$

$$x_e \in \{0,1\} \qquad \forall e \in E$$

Therefore, $\mu > 0$ being fixed, (12.2) can be easily solved. Let opt^*_μ be the value of an optimal solution of (12.2) and let P^*_μ be a path that achieves this solution. Now, let us informally describe how we can obtain the value opt^* of an optimal solution of (12.1).

First, note that any feasible solution of (12.1) is an (s,t)-path P such that $\sum_{e\in E(P)} t_e \leq T$. Therefore, because $\mu > 0$, any feasible solution P has a cost $c_\mu(P)$ in (12.2) that is no larger than its cost $c(P)$ in (12.1). In particular,

$$opt^*_\mu \leq c_\mu(P^*) \leq c(P^*) = opt^*$$

for any $\mu > 0$, where P^* is a feasible solution of (12.1) achieving the optimal value opt^*. That is, the optimal solution of (12.2) provides a lower bound on the optimal value of (12.1).

Let us consider the example of Figure 12.1-(a) for $T = 10$.

First, let us set $\mu = 0$. P^*_0 is the path (v_1, v_2, v_4, v_6) with cost $opt^*_0 = c(P^*_0) = 3$. However, this path is not feasible in the initial problem. Therefore, the relaxation with $\mu = 0$ only provides that $3 \leq opt^*$.

For $\mu = 1$, we get $P^*_1 = (v_1, v_2, v_5, v_6)$ and $opt^*_1 = c(P^*_1) = 20 - \mu \cdot T = 10$. Again, this path is not feasible in the initial problem. However, we got a better lower bound: $10 \leq opt^*$.

For $\mu = 2$ (see Figure 12.1-(b)),two paths achieve the optimal value $opt^*_2 = 15 \leq opt^*$. But, one of them, namely $P^*_2 = (v_1, v_3, v_2, v_5, v_6)$ is such that $\sum_{e\in E(P^*_2)} t_e = T$. Therefore, P^*_2 is a feasible solution of (12.1). Moreover, $15 = c(P^*_2) \geq opt^* \geq c_2(P^*_2) = 15$. Hence, we have solved (12.1).

In this example, we have shown that choosing the 'good' value of μ allows to obtain the optimal solution of the initial problem. In the sequels, we show how to choose a value of μ that will provide a 'good' lower bound for the initial problem and we show that, in some cases, it actually leads to the optimal solution.

12.2 The Lagrangian Relaxation Technique

In this section, we formally define the Lagrangian dual of an optimization problem and show that the solution of the Lagrangian dual provides a lower (resp., upper) bound of the initial minimization (resp., maximization) problem. Moreover, in the case of (convex) linear programmes, the optimal solution of the Lagrangian dual coincides with the optimal solution of the initial problem. Also, the bound obtained thanks to the Lagrangian relaxation is at least as good as the one obtained from fractional relaxation.

12.2.1 Lagrangian dual

Consider the following integer linear programme:

$$\begin{array}{ll} \text{Minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in X \end{array} \tag{12.3}$$

The Lagrangian relaxation procedure uses the idea of relaxing the explicit linear constraints by bringing them into the objective function with associated vector μ called the *Lagrange multiplier*. We refer to the resulting problem

$$\begin{array}{ll} \text{Minimize} & \mathbf{c}^T\mathbf{x} + \mu^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \\ \text{subject to} & \\ & \mathbf{x} \in X \end{array} \tag{12.4}$$

as the *Lagrangian relaxation* or *Lagrangian subproblem* or the original problem (12.3), and we refer to the function

$$L(\mu) = \min\{\mathbf{c}^T\mathbf{x} + \mu^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \mid \mathbf{x} \in X\},$$

as the Lagrangian function.

Lemma 12.1 (Lagrangian Bounding Principle)**.** *For any Lagrangian multiplier μ, the value $L(\mu)$ of the Lagrangian function is a lower bound on the optimal objective function value z^* of the original problem (12.3).*

Proof. Since $\mathbf{A}\mathbf{x} = \mathbf{b}$ for every feasible solution $\mathbf{x}$ of 12.3, for any Lagrangian multiplier μ, $z^* = \min\{\mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in X\} = \min\{\mathbf{c}^T\mathbf{x} + \mu^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in X\}$. Since removing the constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$ from the second formulation cannot lead to an increase in the value of the objective function (the value might decrease), $z^* \geq \min\{\mathbf{c}^T\mathbf{x} + \mu^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \mid \mathbf{x} \in X\} = L(\mu)$. □

12.2.2 Bounds and optimality certificates

To obtain the sharpest possible lower bound, we would need to solve the following optimization problem

$$L^* = \max_{\mu} L(\mu) \tag{12.5}$$

which we refer to as the *Lagrangian Dual* problem associated with the original optimization problem (12.3). The Lagrangian Bounding Principle has the following immediate implication.

Property 12.2 (Weak duality). *The optimal solution L^* of the Lagrangian dual (12.5) is a lower bound on the value z^* of an optimal solution of (12.3), i.e., $L^* \leq z^*$.*

Proof. For any Lagrangian multiplier μ and for any feasible solution $\mathbf{x}$ of Problem 12.3, we have

$$L(\mu) \leq L^* \leq \mathbf{c}^T\mathbf{x}.$$

□

Corollary 12.3 (Optimality test). *Let μ be a Lagrangian multiplier.*
If $\mathbf{x}$ is a feasible solution of (12.3) satisfying $L(\mu) = \mathbf{c}^T\mathbf{x}$. Then

- *$L(\mu)$ is the optimal value of the Lagrangian dual, i.e., $L^* = L(\mu)$, and*
- *$\mathbf{x}$ is an optimal solution of the primal (12.3).*

As indicated by the previous property and its corollary, the Lagrangian Bounding Principle immediately implies one advantage of the Lagrangian relaxation approach. It provides a *certificate* for guaranteeing that a given solution to the primal (12.3) is optimal. Indeed, in the next subsection, we describe a method to compute the optimal solution L^* of the Lagrangian dual.

Even if $L(\mu) < \mathbf{c}^T\mathbf{x}$, having the lower bound permits us to state a bound on how far a given solution is from optimality: If $(\mathbf{c}^T\mathbf{x} - L(\mu))/L(\mu) \leq 0.05$, for example, we know that the value of the feasible solution $\mathbf{x}$ is no more than 5% from optimality. This type of bound is very useful in practice. It permits us to assess the degree of sub-optimality of given solutions and it permits us to terminate our search for an optimal solution when we have a solution which is close enough to optimality for our purposes.

Inequality constraints

In (12.3), the'hard' constraints are all equalities. In practice, problems are described using inequalities. Consider the following optimization problem:

$$\begin{array}{ll} \text{Minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \in X \end{array} \tag{12.6}$$

In that case, we consider only Lagrangian multipliers with positive coefficients. The Lagrangian dual is

$$L^* = \max_{\mu \geq 0} L(\mu) \tag{12.7}$$

In this setting, the Lagrangian Bounding Principle and the weak duality property (12.2) are still valid. However, a vector $\mathbf{x}$ may not be an optimal solution of the primal problem even if

$\mathbf{x}$ is feasible for the primal problem and if $\mathbf{x}$ achieves the optimal solution of the Lagrangian dual $L^* = L(\mu)$ for some $\mu \geq 0$. The optimality test (Corollary) may however be adapted in the following way:

Property 12.4. *If $L(\mu)$ is achieved by a vector $\mathbf{x}$ such that*

- *$\mathbf{x}$ is a feasible solution of (12.6), and moreover*
- *$\mathbf{x}$ satisfies the complementary slackness condition $\mu^T(\mathbf{Ax} - \mathbf{b}) = 0$,*

then, $L(\mu)$ is the optimal value of the Lagrangian dual (12.7) and $\mathbf{x}$ is an optimal solution of the primal (12.6).

12.2.3 Linear Programming

All results presented above do not depend on the kind of considered optimization problem. More precisely, previously, the set X defining the'easy' constraints is arbitrary. We know focus on the linear programming case. More precisely, let us consider the following problem:

$$\begin{array}{ll} \text{Minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \\ & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{Dx} \geq \mathbf{q} \\ & \mathbf{x} \geq 0 \end{array} \tag{12.8}$$

Recall that the corresponding dual problem is (see Section 9.3.2):

$$\begin{array}{ll} \text{Maximize} & \mathbf{b}^T\mathbf{y} + \mathbf{q}^T\mathbf{z} \\ \text{subject to} & \\ & \mathbf{A}^T\mathbf{y} + \mathbf{D}^T\mathbf{z} \leq \mathbf{c} \\ & \mathbf{y}_2 \geq \mathbf{0} \end{array} \tag{12.9}$$

For any vector μ, we set the Lagrangian function as

$$L(\mu) = \min\{\mathbf{c}^T\mathbf{x} + \mu^T(\mathbf{Ax} - \mathbf{b}) \mid \mathbf{Dx} \geq \mathbf{q}, \mathbf{x} \geq 0\}. \tag{12.10}$$

Theorem 12.5. *Let (P) be any linear programme such as Problem 12.8. The optimal value $L^* = \max_\mu L(\mu)$ of the Lagrangian dual of (P) coincides with the optimal value opt^* of (P).*

Proof. Let $\mathbf{x}^*$ be a vector achieving the optimal value opt^* of (12.8) and let $(\mathbf{y}^*, \mathbf{z}^*)$ be an optimal solution of the dual 12.9. Then $\mathbf{A}^T\mathbf{y}^* + \mathbf{D}^T\mathbf{z}^* - \mathbf{c} \geq 0$. Moreover, by the complementary slackness conditions (Theorem 9.11):

- $(\mathbf{A}^T\mathbf{y}^* + \mathbf{D}^T\mathbf{z}^* - \mathbf{c})\mathbf{x}^* = 0$
- $\binom{\mathbf{y}^*}{\mathbf{z}^*}\left(\binom{\mathbf{A}}{\mathbf{D}}\mathbf{x}^* - \binom{b}{q}\right) = \mathbf{z}^*(\mathbf{D}x^* - \mathbf{q}) = 0$

Set $\mu = -\mathbf{y}^*$, we have $L(-\mathbf{y}^*) = \min\{\mathbf{c}^T\mathbf{x} - (\mathbf{y}^*)^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \mid \mathbf{D}\mathbf{x} \geq \mathbf{q}, \mathbf{x} \geq 0\}$. That is, any vector achieves $L(-\mathbf{y}^*)$ if and only if it achieves the optimal value of the following linear programme:

$$\text{Minimize} \quad (\mathbf{c}^T - (\mathbf{y}^*)^T\mathbf{A})\mathbf{x} \quad \text{subject to} \quad \mathbf{D}\mathbf{x} \geq \mathbf{q} \text{ and } \mathbf{x} \geq 0. \tag{12.11}$$

The corresponding dual problem is

$$\text{Maximize} \quad \mathbf{q}^T\mathbf{z} \quad \text{subject to} \quad \mathbf{D}^T\mathbf{z} \leq \mathbf{c} - \mathbf{A}^T\mathbf{y}^* \text{ and } \mathbf{z} \geq 0. \tag{12.12}$$

Therefore, the complementary slackness conditions implies that if there is a feasible solution $\tilde{\mathbf{x}}$ to (12.11) and a feasible solution $\tilde{\mathbf{z}}$ feasible to (12.12) such that

- $\tilde{\mathbf{z}}(\mathbf{D}\tilde{x} - \mathbf{q}) = 0$, and
- $(\mathbf{D}^T\tilde{\mathbf{z}} - \mathbf{c} + \mathbf{A}^T\mathbf{y}^*)\tilde{\mathbf{x}} = 0$,

then $\tilde{\mathbf{x}}$ is an optimal solution of (12.11).

Since setting $\tilde{\mathbf{x}} = \mathbf{x}^*$ and $\tilde{\mathbf{z}} = \mathbf{z}^*$ satisfies the complementary slackness conditions, x^* is an optimal solution to (12.11).

Thus, $L(-\mathbf{y}^*) = \mathbf{c}^T\mathbf{x}^* + \mu^T(\mathbf{A}\mathbf{x}^* - \mathbf{b}) = \mathbf{c}^T\mathbf{x}^* = opt^*$. The result follows Corollary 12.2.2. □

Theorem 12.6. *Let X be a finite set in $\mathbb{R}^n$ and let $\mathcal{H}(X)$ its convex hull. Then, the Lagrangian dual of*

$$\textit{Minimize} \quad \mathbf{c}^T\mathbf{x} \quad \textit{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \textit{ and } \mathbf{x} \in X\}$$

has the same optimal value as

$$\textit{Minimize} \quad \mathbf{c}^T\mathbf{x} \quad \textit{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \textit{ and } \mathbf{x} \in \mathcal{H}(X)\}.$$

Proof. Let $L(\mu) = \min\{\mathbf{c}^T\mathbf{x} + \mu(\mathbf{A}\mathbf{x} - \mathbf{b}) \mid \mathbf{x} \in X\}$. It is equivalent to $L(\mu) = \min\{\mathbf{c}^T\mathbf{x} + \mu(\mathbf{A}\mathbf{x} - \mathbf{b}) \mid \mathbf{x} \in \mathcal{H}(X)\}$ because the optimal values of the second formulation are reached at some extreme points of the polytope $\mathcal{H}(X)$ and that any vertex of $\mathcal{H}(X)$ belongs to X.

Therefore, the solution of the Lagrangian dual of the initial problem equals the solution of the Lagrangian dual of

$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \in \mathcal{H}(X)\}.$$

The convex hull of a finite number of points can be defined as the intersection of a finite family of half-spaces, i.e., by a finite number of inequalities. Therefore, applying Theorem 12.5 to

$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \in \mathcal{H}(X)\},$$

we get that its optimal value and the one of its Lagrangian dual coincide. □

Theorem 12.7. *Let (ILP) be an integer linear programme. Then the bound achieved by a Lagrangian relaxation of (ILP) is at least as good as the optimal value of its fractional relaxation.*

Proof. Consider the integer linear programme (ILP):

$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \in X \cap \mathbb{Z}^n$$

where X is convex since it is defined by linear inequalities. By Theorem 12.6, its Lagrangian relaxation has the same solution as (LR):

$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \in \mathcal{H}(X \cap \mathbb{Z}^n).$$

Since the convex hull $H = \mathcal{H}(X \cap \mathbb{Z}^n)$ of $X \cap \mathbb{Z}^n$ is such that $H \subseteq X$, we get that the solution of (LR) is not better than the one of (LP):

$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \in X,$$

the fractional relaxation of (ILP). Hence, $opt(LP) \leq opt(LR) \leq opt(ILP)$. □

12.2.4 Solving the Lagrangian dual

In this subsection, we describe a method to approximate the Lagrangian dual

$$L^* = \max_{\mu} L(\mu) \tag{12.13}$$

of the Lagrangian function which relaxes the constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$, that is

$$L(\mu) = \min\{\mathbf{c}^T\mathbf{x} + \mu^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \mid \mathbf{x} \in X\}.$$

Recall that the principle of the Lagrangian relaxation is to include the‘hard’ constraints into the objective function. In other words, optimizing a linear function over X is assumed to be‘easy’. Therefore, μ being fixed, we can compute the value of $L(\mu)$ and an corresponding optimal solution $\mathbf{x} \in X$.

Note that the Lagrangian function is the lower envelope of the set of hyperplanes $\{ \mathbf{c}^T\mathbf{x} + \mu^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \mid \mathbf{x} \in X \}$. Therefore, $L(\mu)$ is a concave function. Say differently, it is equivalent to solve the following linear programme:

$$\begin{array}{ll} \text{Maximize} & w \\ \text{subject to} & \\ & w \leq \mathbf{c}^T\mathbf{x} + \mu^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \quad \mathbf{x} \in X,\ \mu \in \mathbb{R}^n \end{array}$$

Generally the number of constraints of such a programme is exponential, so we use a gradient descent method to compute a value as close as desired to the Lagrangian dual. More precisely, given a concave function $f : \mathbb{R}^n \to \mathbb{R}$, a vector $\mathbf{g}$ is a *subgradient* of f at $\mathbf{x}$ if, for any $\mathbf{y} \in \mathbb{R}^n$, $f(\mathbf{y}) \leq f(\mathbf{x}) + \mathbf{g}^T(\mathbf{y} - \mathbf{x})$. The function f is *differentiable* in $\mathbf{x}$ if and only if f admits a unique subgradient in $\mathbf{x}$. If L were differentiable, we would use the gradient descent method to converge toward the optimal value. However, in our case, L is not everywhere differentiable because it is a piecewise linear function. So we compute a sequence of $(\mu_k)_{k \in \mathbb{N}}$ such that $L(\mu_k)$ converges to the optimal solution, using the following subgradient method.

Algorithm 12.1 (Subgradient method)**.**

0. Set $k = 0$ and choose $\mu_0 \in \mathbb{R}^n$;
1. Compute $L(\mu_k)$ and a vector $\mathbf{x}_k \in X$ where it is achieved;
2. Choose a subgradient $\mathbf{g}_k = \mathbf{A}\mathbf{x}_k - \mathbf{b}$ of the function L at μ_k;
3. If $\mathbf{g}_k = 0$, then stop, the optimal solution is $L(\mu_k)$
4. Compute $\mu_{k+1} = \mu_k + \theta_k^T.\mathbf{g}_k$ where θ_k is the stepsize at this step.
5. Increment k and go to Step 2.

In practice, the formula to calculate the stepsize is $\theta_k = \frac{UB - L(\mu_k)}{||\mathbf{A}\mathbf{x}_k - \mathbf{b}||^2}$ where UB is an upper bound on the optimal solution we want to compute.

In the case when the constraints of the initial programme that we relax were $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, then the method must be slightly modified such that $\mu_k \geq 0$ for any $k \geq 0$. For this purpose, at Step 4., the ith coordinate of μ_{k+1} is taken as the maximum between 0 and the ith coordinate of $\mu_k + \theta_k^T.\mathbf{g}_k$.

In all cases, the number k of iterations depends on the desired accuracy of the result.

12.3 Applications

12.3.1 Travelling Salesman Problem

Let $D = (V, E)$ be an arc-weighted complete digraph, i.e., for any $u, v \in V$ there are both the arcs (u, v) and (v, u) and there is a cost function $c : E \to \mathbb{R}$. The cost of a (directed) cycle is the sum of the cost of its arcs. The TRAVELLING SALESMAN problem is to find a minimum-cost (directed) cycle passing exactly once through all vertices. This problem is a well known $\mathcal{NP}$-hard problem.

A first way to formulate this problem as an integer linear programme is the following, where the variable $x_e \in \{0, 1\}$ indicates if the arc e belongs to the cycle or not.

$$
\begin{array}{lll}
\text{Minimize} & \sum_{e \in E} c(e) x_e & \\
\text{subject to} & & \\
& \sum_{e \text{ entering } v} x_e = 1 & \text{for all } v \in V \\
& \sum_{e \text{ leaving } v} x_e = 1 & \text{for all } v \in V \\
& \sum_{e \in E(S)} x_e \le |S| - 1 & \text{for all } S \subseteq V \\
& x_e \in \{0,1\} & \text{for all } e \in E
\end{array}
\tag{12.14}
$$

The first two sets of constraints and the integrity constraint ensure that the solution is a disjoint union of cycles covering all vertices exactly once, thats is a *cycle factor*. So, if the third set of constraints is omitted, this programme solves the minimum-cost cycle-factor problem. The minimum cycle-factor problem consists in finding a cycle factor of a digraph $D = (V,E)$ with minimum cost. This problem can be solved in polynomial time since it is equivalent to solve a minimum-cost perfect matching problem. Indeed, consider the bipartite graph G with vertex set $\{v^-, v^+ \mid v \in V\}$ in which there is an edge v^-u^+ of cost $c(vu)$ for each arc $(v,u) \in E$. There is a one-to-one correspondence between the perfect matchings of G and the cycle-factors of D.

A possibility would be to relax the third set of constraints and include it in the objective funcction, but the resulting objective function would be a sum of an exponential number of terms, which is not convenient. We shall now see how to replace the third set of constraints by some other, so that Lagraangian relaxation is practically possible.

The third set of constraints ensures we have only one cycle. In other words, it ensures the connectivity of the solution. We now give a new linear programming formulation of TRAVELLING SALESMAN in which this set of constraints is replaced by two others guaranteeing the connectivity of any feasible solutions. The first of this two new sets of constraints ensures that there is a flow from some vertex $s \in V$ to all other vertices. The second forces this flow to use the arcs of the solution, so that any feasible solution must be connected.

$$
\begin{array}{lll}
\text{Minimize} & \sum_{e\in E} c(e)x_e & \\
\text{subject to} & & \\
& \sum_{e \text{ entering } v} x_e = 1 & \text{for all } v \in V \\
& \sum_{e \text{ leaving } v} x_e = 1 & \text{for all } v \in V \\
& \sum_{e \text{ entering } s} f_e - \sum_{e \text{ leaving } s} f_e = -(n-1) & \\
& \sum_{e \text{ entering } s} f_e - \sum_{e \text{ leaving } s} f_e = 1 & \text{for all } v \in V \setminus \{s\} \\
& f_e \leq (n-1)x_e & \forall e \in E \\
& x_e \in \{0,1\} & \text{for all } e \in E
\end{array}
$$

Using this formulation and relaxing the constraint $f_e \leq (n-1)x_e$, the problem is equivalent to solving separately a flow problem and a minimum cycle-factor problem.

Bibliography

[1] R. K. Ahuja, T. L. Magnanti and J. B. Orlin. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.