IE 511: Integer Programming, Spring 2021 **27 Apr, 2021**

Lecture 26: Lagrangian Dual

Lecturer: Karthik Chandrasekaran *Scribe: Karthik*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

26.1 Speeding Up Technique 1: Lagrangian Dual

To speed up solution techniques, we will try to derive good bounds for the IP. For maximization IP, our aim is to derive small upper bounds and for minimization IP, our aim is to derive large lower bounds. This will help in the branch and bound approach as it will enable the pruning of more nodes in the enumeration tree. Consider the IP

$$z = \max\{c^T x : Ax \le b, Dx \le d, x \in \mathbb{Z}^n\}.$$

In several applications, we can partition the constraints of the IP as $Ax \le b, Dx \le d$ where $Ax \le b$ is well-structured/easy, (i.e., IP with just these constraints are easy to solve or have good approximation algorithms) and $Dx \le d$ are complicating constraints. Examples of well-structured/easy constraints (that we have seen already) are flow constraints, matching constraints, TU constraint matrix with integral RHS vector, and matroid constraints.

Example: Minimum cost degree bounded spanning tree problem. This is a slightly more advanced version of the minimum cost spanning tree problem where we require bounds on the degrees of the vertices in the spanning tree. This problem also arises in connectivity and network design contexts.

Given: Graph $G = (V, E), c : E \to \mathbb{R}_+, b : V \to \mathbb{Z}_+$
Goal: $\min\{\sum_{e\in T} c_e : T \text{ is a spanning tree with } \deg_T(u) \le b_u \ \forall u \in V\}$

IP:

$$\min \sum_{e\in E} c_e x_e$$

$$Ax \le b \left[\begin{array}{l} \sum_{e\in F} x_f \le r(F) \ \forall\ F \subseteq E \\ \sum_{e\in E} x_e = |V| - 1 \end{array} \right.$$

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

$$Dx \le d \left[\ \sum_{e\in\delta(u)} x_e \le b_u \ \forall u \in V \right.$$

where $r(F)$ is the rank function of the graphic matroid corresponding to G. If we drop the complicating constraints then the IP can be solved easily—it is the minimum cost spanning tree problem, which we know can be solved efficiently by a greedy algorithm via the matroid connection.

Many discrete optimization problems have such a mix of well-structured and complicating constraints. Dropping these complicating constraints gives a relaxation (which is still an IP) and solving the resulting relaxation (as an IP) gives a bound on the optimum. However, the bound could be weak since some constraints are ignored. An alternative way to address this is by bringing these complicating constraints into the objective with a penalty term. This leads us to the Lagrangian relaxation.

Definition 1. Consider the IP $z = \max\{c^T x : x \in S, Dx \leq d\}$ where $D \in \mathbb{R}^{m \times n}, d \in \mathbb{R}^m$ and $S \subseteq \mathbb{Z}^n$. For $u \in \mathbb{R}^m, u \geq 0$, let IP$(u)$ be

$$z(u) := \max\{c^T x + u^T(d - Dx) : x \in S\}.$$

IP(u) is said to be the *Lagrangian relaxation* of IP *with parameter* u. Here, u is the *Lagrange multiplier* associated with the constraint system $Dx \leq d$.

Note that if S is a discrete set, then $z(u)$ is still an IP. IP(u) relaxes the complicating constraints by having them in the objective with a penalty term $u^T(d - Dx)$. We will study the power of this relaxation now.

Proposition 2. *For all $u \geq 0$, IP(u) is a relaxation of IP.*

Proof. We prove the properties needed of a relaxation (see Lecture 2).

1. Feasible region of IP(u) contains that of IP.
2. We have $u \geq 0$ and $Dx \leq d$ for all x in feasible region of IP. Therefore, $c^T x + u^T(d - Dx) \geq c^T x$ for all x in the feasible region of IP.

□

Proposition 2 implies that $z \leq z(u)$ for all $u \geq 0$.

Note that the Lagrangian relaxation is a relaxation for not only well-structured + complicating constraints but any collection of constraints and is hence significant for unstructured IPs in order to derive bounds while executing the Branch and Bound algorithm. So, Lagrangian relaxation is often used in Branch and Bound implementations.

Since the Lagrangian relaxation is indeed a relaxation for every non-negative u, the next natural question is, what choice of u should we use? Our goal is to find the tightest possible upper bound for the maximization problem. This leads to the following definition:

Definition 3. Let

$$w_{\text{LD}} := \min\{z(u) : u \geq 0\}.$$

The quantity w_{LD} is known as the *Lagrangian Dual Problem.*

We first see how to identify optimal solutions from the Lagrangian relaxation.

Optimality. Suppose we have a non-negative u such that the optimum solution to the Lagrangian Relaxation satisfies the complicating constraints and complementary slackness. Then the optimum solution to the Lagrangian Relaxation is also an optimum to the IP.

Lemma 3.1. *Suppose $u \geq 0$ and*

(i) $x(u)$ is an optimum solution to IP(u),

(ii) $Dx(u) \leq d$, and

(iii) $[Dx(u)]_i = d_i$ if $u_i > 0$ (complementarity).

Then $x(u)$ is an optimum solution to the IP.

Proof.

$$\begin{aligned}
w_{\text{LD}} &\leq z(u) && \text{(by definition of } w_{\text{LD}}\text{)} \\
&= c^T x(u) + u^T(d - Dx(u)) && \text{(by (i))} \\
&= c^T x(u) && \text{(by (iii))} \\
&\leq z && \text{(by (ii))} \\
&\leq w_{\text{LD}}. && \text{(by Proposition 2)}
\end{aligned}$$

Hence, we should have equality throughout which implies that $x(u)$ is an optimum to the IP. □

So, we can use this lemma to verify when the relaxation gives us an optimum.

Next, we will focus on the following questions:

1. How tight is w_{LD}?
2. Can we compute w_{LD}?

To start off, let us understand how far is the objective value of the Lagrangian dual from the IP value.

26.2 Strength of Lagrangian Dual

For simplicity, let S be finite, i.e., $S := \{x^1, \ldots, x^r\}$. Then,

$$\begin{aligned}
w_{\text{LD}} &= \min_{u \geq 0} z(u) \\
&= \min_{u \geq 0} \left\{ \max_{x \in S} c^T x + u^T(d - Dx) \right\} \\
&= \min_{u \geq 0} \left\{ \max_{i \in [r]} c^T x^i + u^T(d - Dx^i) \right\} \\
&= \min \left\{ \eta : \eta \geq c^T x^i + u^T(d - Dx^i) \forall i \in [r], \eta \in \mathbb{R}, u \in \mathbb{R}^m, u \geq 0 \right\}. \qquad (26.1)
\end{aligned}$$

In the formulation of Problem (26.1), we have introduced a variable η to represent the upper bound on the optimum value of the Lagrangian relaxation with parameter u. Observe that Problem (26.1) is an LP with variables η and u. The dual of Problem (26.1) is

$$\begin{aligned} w_{\text{LD}} = \max & \sum_{i=1}^{r} \mu_i (c^T x^i) \\ \text{subject to } & \sum_{i=1}^{r} \mu_i (Dx^i - d) \leq 0 \\ & \sum_{i=1}^{r} \mu_i = 1 \\ & \mu \in \mathbb{R}^r \\ & \mu \geq 0 \end{aligned}$$

Setting $x = \sum_{i=1}^{r} \mu_i x^i$ with $\sum_{i=1}^{r} \mu_i = 1, \mu \in \mathbb{R}^r_+$, we have write the dual of Problem (26.1) as

$$w_{\text{LD}} = \max\{c^T x : Dx \leq d, x \in \text{convex-hull}(S)\}. \tag{26.2}$$

Thus, we have derived the following results:

Theorem 4. *If $S := \{x \in \mathbb{Z}^n : Ax \leq b\}$, then*

$$w_{LD} = \max\{c^T x : Dx \leq d, x \in \textit{convex-hull}(S)\}.$$

Recall that $z := \max\{c^T x : x \in S, Dx \in d\}$. We derived Theorem 4 when S is a finite set. In fact, Theorem 4 holds for S of the form $\{x \in \mathbb{Z}^n : Ax \leq b\}$. This theorem tells us the strength (i.e., tightness) of the bound obtained from the Lagrangian Dual. Essentially, the Lagrangian Dual convexifies the feasible region and hence gives an LP.

Corollary 4.1. *If $S = \{x \in \mathbb{Z}^n : Ax \leq b\}$ and convex-hull$(S) = \{x \in \mathbb{R}^n : A'x \leq b'\}$, then $w_{LD} = \max\{c^T x : A'x \leq b', Dx \leq d, x \in \mathbb{R}^n\}$.*

In certain cases, the Lagrangian Dual ends up being the LP-relaxation. Note that if S is the set of incidence vectors of matchings in a bipartite graph or forests of a given graph then we can obtain convex-hull(S) by simply relaxing (i.e., dropping) the integrality constraints.

26.3 Solving the Lagrangian Dual

Consider the IP $z := \max\{c^T x : x \in S, Dx \in d\}$ and IP(u) given by $z(u) := \max\{c^T x + u^T(d - Dx) : x \in S\}$.

We know that $z \leq z(u)\ \forall u \geq 0$. Recall that the Lagrangian dual problem is

$$w_{\text{LD}} := \min_{u \geq 0} z(u).$$

We now see how to solve the Lagrangian Dual problem.

26.3.1 Approach 1: Solve Problem (26.2)

If convex-hull(S) needs lot of constraints, then use the cutting plane algorithm.

26.3.2 Approach 2: Subgradient Algorithm

The Lagrangian relaxation $z(u) = \max_{i\in[r]}\{u^T(d - Dx^i) + c^T x^i\}$ is a piecewise linear convex function and thus, the Lagrangian Dual problem $w_{\text{LD}} = \min_{u\geq 0} z(u)$ is equivalent to minimizing a piecewise linear convex function. We recall the geometry and algebraic form of piecewise linear convex function below.

Piecewise linear convex function:

- *Geometry:*

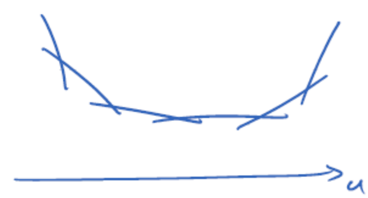

Figure 26.1: Piecewise linear convex function

- *Algebra:* $f(u) = \max_{i\in[r]}\{u^T a^i - b_i\}$

We immediately observe that the Lagrangian relaxation is a piecewise linear convex function and moreover, the Lagrangian Dual problem is equivalent to minimizing a piecewise linear convex function. The Subgradient Algorithm is designed to find a minimum of a piecewise linear convex function. It is similar to gradient descent for minimizing a convex function, but is applicable when the function is not differentiable. We emphasize that a piecewise linear convex function is not differentiable. A subgradient is a natural generalization of a gradient. It is defined below.

Definition 5. Let $f : \mathbb{R}^m \to \mathbb{R}$ be a convex function and $u \in \mathbb{R}^m$. *Subgradient of f at u* is a vector $\gamma(u) \in \mathbb{R}^m$ such that
$$f(v) \geq f(u) + \gamma(u)^T(v - u) \ \forall v \in \mathbb{R}^m.$$

Example: If f is a continuous differentiable function, then the gradient of f at $\bar{u}$ given by
$$\nabla f(\bar{u}) := \left(\frac{\partial f}{\partial u_1}, \ldots, \frac{\partial f}{\partial u_m}\right)\Bigg|_{u=\bar{u}}.$$

Exercise. If f is a continuous differentiable function, then the gradient of f at $\bar{u}$ is a subgradient of f at $\bar{u}$.

Just like how gradient can be used to recognize a minimizer of a continuous differentiable convex function, the subgradient can also be used to recognize a minimizer of a continuous convex function.

Proposition 6 (Exercise). *Let $f : \mathbb{R}^m \to \mathbb{R}$ be convex. A point $u \in \mathbb{R}^n$ is an optimum solution of* $\min\{f(u) : u \in \mathbb{R}^m\}$ *iff $\bar{0}$ is a subgradient of f at u.*

If the function is a piecewise linear convex function, then the subgradient of the function at any given point is easy to compute as illustrated by the following lemma.

Proposition 7 (**Exercise**). *Let* $\bar{u} \geq 0$ *and* $x(\bar{u})$ *be an optimum to the Lagrangian Relaxation* $IP(u)$*. Then* $d - Dx(\bar{u})$ *is a subgradient of* $z(u)$ *at* $\bar{u}$*.*

Subgradient Algorithm Outline. It is an iterative algorithm. In each iteration,

1. we choose an arbitrary subgradient and
2. move *opposite* to that direction by a small step.

With appropriate choice of subgradient and step size, the algorithm finds a point whose function value is close to that of the optimum. We state the algorithm below.

Algorithm 1: Subgradient Algorithm

Initialize $u^0 \in \mathbb{R}^n_+, k \leftarrow 0$;
repeat
  Solve the Lagrangian Relaxation IP(u^*) to obtain optimum solution $x(u^k)$;
  $\gamma^k \leftarrow d - Dx(u^k)$; // subgradient of $z(u)$ at u^k
  if $\gamma^k = 0$ **then**
    STOP and return $x(u^k)$
  $u^{k+1} \leftarrow \max\{u^k - \theta_k \gamma^k\}$ for step size $\theta_k > 0$;
  $k \leftarrow k + 1$

26.4 Choosing constraints to dualize in Lagrangian dual

Suppose we have an IP of the following form:

$$\begin{aligned} z = \max\{c^T x : A^1 x &\leq b^1 \\ A^2 x &\leq b^2 \\ x &\in \mathbb{Z}^n_+\} \end{aligned}$$

Then, we need to decide which constraints to dualize. We mention the trade-offs to keep in mind while deciding which constraints to dualize.

1. *Ability to solve Lagrangian Dual Problem* $w_{LD} = \min_{u \geq 0} z(u)$. This is typically difficult to estimate. The number of dual variables is a crude estimate for this.
2. *Ability to solve Lagrangian Relaxation IP*(u)*.* This is usually problem specific.
3. *Strength of the bound resulting from the Lagrangian Dual* w_{LD}*.* See Theorem 4.

26.5 Application: Set Cover Problem

Covering constrained discrete optimization problems arise in contexts where we have to minimize cost subject to producing at least as many items to meet a pre-specified demand. The set cover problem is a quintessential *covering constrained* discrete optimization problem.

Given:	A finite ground set N, $\mathcal{F} \subseteq 2^N$, costs $c : \mathcal{F} \to \mathbb{R}_+$
Goal:	$\min \left\{ \sum_{S \in \mathcal{S}} c_S : \bigcup_{S \in \mathcal{S}} S = N \right\}$

Let us summarize the data using the matrix $A \in \{0,1\}^{N \times \mathcal{F}}$ with entries given by

$$A_{iS} := \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise.} \end{cases}$$

for every $i \in N$ and $S \in \mathcal{F}$. With this matrix representation, we have the following IP formulation for the set cover problem:

$$\min \left\{ \begin{array}{ll} \sum_{S \in \mathcal{F}} c_S x_S : & \sum_{S \in \mathcal{F}} A_{iS} x_S \geq 1 \ \forall i \in N \quad \text{(covering constraints)} \\ & x_S \in \{0,1\} \ \forall S \in F \end{array} \right\}.$$

Lagrangian Relaxation. By dualizing all covering constraints, we obtain the following Lagrangian relaxation:

$$z(u) = \min \left\{ \sum_{S \in \mathcal{F}} c_S x_S + \sum_{i \in N} u_i (1 - \sum_{S \in \mathcal{F}} A_{iS} x_S) : x \in \{0,1\}^{\mathcal{F}} \right\}$$

i.e.,

$$z(u) = \min \left\{ \sum_{i \in N} u_i + \sum_{S \in \mathcal{F}} (c_S - \sum_{i \in N} u_i A_{iS}) x_S : x \in \{0,1\}^{\mathcal{F}} \right\} \tag{26.3}$$

For a fixed $u \geq 0$, IP(u) given in Problem (26.3) is easy to solve: set

$$[x(u)]_S := \begin{cases} 1 & \text{if } c_S - \sum_{i \in N} u_i A_{iS} < 0 \\ 0 & \text{otherwise.} \end{cases}$$

Exercise. $x(u)$ is an optimum to IP(u).

It means that, for a given $u \geq 0$, we can find an optimum solution $x(u)$ to the Lagrangian relaxation $z(u)$ quickly. Therefore, each iteration of the subgradient algorithm can be implemented to run very quickly. Keep in mind that the subgradient algorithm only obtains w_{LD} and does not solve the IP.