Lecture 27: Column Generation, Knapsack Cover Inequalities

Lecturer: Karthik Chandrasekaran *Scribe: Karthik*

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

27.1 Speeding-up Technique 2: Column Generation

A recurring idea in algorithm design is that of decomposition. We will see another use of it today.

Suppose we have an IP of the following form:

$$\begin{aligned}
\max c^1x^1 + c^2x^2 + \cdots + c^kx^k & & \\
A^1x^1 + A^2x^2 + \cdots + A^kx^k &= b & (0)\\
D^1x^1 \qquad\qquad\qquad &\le d_1 & (1)\\
D^2x^2 \qquad\qquad &\le d_2 & (2)\\
\ddots \qquad & & \\
D^kx^k &\le d_k & (k)\\
x^1 \in \mathbb{Z}_+^{n_1}, \ldots, x^k \in \mathbb{Z}_+^{n_k} & &
\end{aligned}$$

We notice that constraints $(1), ..., (k)$ are over disjoint sets of variables while constraint (0) involves all variables. Let

$$S_j := \{x^j \in \mathbb{Z}_+^{n_j} : D^jx^j \le d_j\}.$$

Then S^j and S^i are independent for distinct $i, j \in [k]$. It is only the joint constraint $\sum_{j=1}^k A^jx^j = b$ that links the disjoint set of variables. We have already seen two approaches to benefit from such structures:

1. Cut generation: Generate valid inequalities for each subset S^j and use disjunctive cuts.
2. Lagrangian Relaxation: Dualize the joint constraint.

Let us see another way to exploit such structures. Our IP can be restated in the following convenient form:

$$z = \max\left\{\sum_{i=1}^k c^ix^i : \sum_{i=1}^k A^ix^i = b, x^i \in S^i\ \forall i \in [k]\right\} \tag{27.1}$$

For simplicity, suppose that S^i is bounded for every $i \in [k]$. So, $S^i = \{x^{i,1}, \ldots, x^{i,T_i}\}$. It implies that

$$S^i = \left\{\begin{array}{ll} x^i \in \mathbb{R}^{n_i}: & x^i = \sum_{t=1}^{T_i} \lambda_{i,t}x^{i,t} \\ & \sum_{t=1}^{T_i} \lambda_{i,t} = 1 \\ & \lambda_{i,t} \in \{0,1\}\ \forall t \in [T_i] \end{array}\right\} \tag{27.2}$$

Substituting (27.2) back into IP (27.1), we get the *IP master problem*:

$$z^{\text{IP-Master}} = \max \left\{ \begin{array}{ll} \sum_{i=1}^{k} \sum_{t=1}^{T_i} (c^i x^{i,t}) \lambda_{i,t} : & \sum_{i=1}^{k} \sum_{t=1}^{T_i} (A^i x^{i,t}) \lambda_{i,t} = b \\ & \sum_{t=1}^{T_i} \lambda_{i,t} = 1 \ \ \forall i \in [k] \\ & \lambda_{i,t} \in \{0,1\} \ \forall t \in [T_i], \forall i \in [k] \end{array} \right\}. \tag{27.3}$$

Problem (27.3) is a reformation of IP (27.1) with $\lambda_{i,t}$ as variables. We observe the following about the reformulation:

- *Con:* The number of variables $\lambda_{i,t}$ could be very large.
- *Pro:* LP-relaxation may be solvable faster which may lead to faster bounds for the IP.

Solving LP-relaxation of the IP master problem. LPs of the form obtained by relaxing the IP master problem are best solved by the primal simplex method. Primal simplex only needs the basis to be stored. So, we can run the primal simplex by *generating columns* only if needed.

Strength of the LP-relaxation of the IP master problem. We now discuss the strength of the LP-relaxation of the IP master problem.

Theorem 1. *[Strength of Master LP] Let z^{LPM} be the optimum value of LP-relaxation of the IP master problem. Then,*

$$z^{LPM} = \max \left\{ \sum_{i=1}^{k} c^i x^i : \sum_{i=1}^{k} A^i x^i = b, x^i \in \textit{convex-hull}(S^i) \ \forall i \in [k] \right\}.$$

Proof. The LP-relaxation of the IP master problem is obtained by substituting

$$x^i = \sum_{t=1}^{T_i} \lambda_{i,t} x^{i,t}, \quad \sum_{t=1}^{T_i} \lambda_{i,t} = 1, \quad \lambda_{i,t} \geq 0 \quad \forall t \in [T_i]$$

in Problem (27.3) and dropping the binary constraints. This is equivalent to saying that $x^i \in$ convex-hull(S^i). □

Note that the form of the RHS in Theorem 1 resembles the form of the result that we showed for the strength of the Lagrangian Dual. We make this connection explicit now. The Lagrangian Dual obtained by dualizing the linking constraints is

$$w_{\text{LD}} = \min_{u \in \mathbb{R}^m} z(u)$$

where

$$\begin{aligned} z(u) &= \max \left\{ \sum_{i=1}^{k} c^i x^i + u^T \left(b - \sum_{i=1}^{k} A^i x^i \right) : x^i \in S^i \ \forall i \in [k] \right\} \\ &= u^T b + \sum_{i=1}^{k} \max \left\{ (c^i - u^T A^i) x^i : x^i \in S^i \right\}. \end{aligned}$$

Note that calculating $z(u)$ breaks into k subproblems. From the result on Lagrangian Dual, we know, w_{LD} is the RHS of Theorem 1. So, we have the following corollary:

Corollary 1.1. $z^{LPM} = w_{LD}$.

So, column generation gives another way to solve the Lagrangian Dual problem for IPs with such decomposable structure.

27.2 Speeding up Technique 3: Better Cuts

There are two natural criteria for cuts in the cutting plane algorithm:

1. Cut deep: we would like to cut as much of the polyhedron as possible.
2. Cut fast: we would like cut generation to be efficient.

We have the following way to compare cuts.

Definition 2 (Comparing cuts)**.** Let $c^T x \le \delta, w^T x \le d$ be two valid inequalities for $P \subseteq \mathbb{R}^n_+$. $c^T x \le \delta$ *dominates* $w^T x \le d$ if there exists $\lambda > 0$ such that $c \ge \lambda w, \delta \le \lambda d$ and $(c, \delta) \ne (\lambda w, \lambda d)$.

Observation. If $c^T x \le \delta$ dominates $w^T x \le d$ then

$$\left\{x \in \mathbb{R}^n_+ : c^T x \le \delta\right\} \subseteq \left\{x \in \mathbb{R}^n_+ : w^T x \le d\right\}.$$

27.2.1 Application: 0-1 knapsack

We will see how to generate better cuts for the 0-1 knapsack IP. The 0-1 knapsack IP is a quintessential BIP. So, solving and speeding-up techniques for the knapsack IP help in addressing general BIPs. The 0-1 knapsack IP is of the following form: $\max\left\{c^T x : \sum_{i\in N} a_i x_i \le b, x \in \{0,1\}^N\right\}$. Let

$$P(N, a, b) := \left\{x \in \mathbb{R}^N : \sum_{i\in N} a_i x_i \le b, 0 \le x_j \le 1\ \forall j \in N\right\}$$

and

$$P_I(N, a, b) := \text{convex-hull}\left\{x \in \{0,1\}^N : \sum_{i\in N} a_i x_i \le b\right\}.$$

We will call the constraint $\sum_{i\in N} a_i x_i \le b$ as the *knapsack constraint.* Note that knapsack constraint occurs as a constraint in general IPs. For this reason, it is important/useful to know how to generate valid inequalities for $P_I(N, a, b)$.

We will assume that $0 \le a_j \le b$ for all $j \in b$. This assumption is without loss of generality for the following reasons.

- If $a_j > b$, then we can fix $x_j^* = 0$ and consequently, remove x_j itself.
- If $a_j < 0$, there replace x_j by $1 - x_j$ to satisfy the assumption.

With this assumption, our ideal goal is to obtain facet-defining inequalities for $P_I(N, a, b)$.

A running example. We will consider the following running example:

$$P := \left\{x \in \mathbb{R}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19, 0 \leq x_i \leq 1 \ \forall x \in [7]\right\}$$

Observation 1. The inequality

$$x_1 + x_2 + x_6 \leq 2 \tag{27.4}$$

is valid for P_I (because $11 + 6 + 4 = 21 > 19$ and hence we cannot pick all three of $\{1, 2, 6\}$).

This observation is generalized by the notion of covers.

Definition 3. A set $C \subseteq N$ is a *cover* if $\sum_{j \in C} a_j > b$.

Back to the running example. The sets C_1, C_2, C_3 below are covers. The covers immediately lead to valid inequalities for P_I as shown below:

$$\begin{aligned} &C_1 = \{1, 2, 6\} \\ &C_2 = \{3, 4, 5, 6\} \implies x_3 + x_4 + x_5 + x_6 \leq 3 \text{ is valid for } P_I. \\ &C_3 = \{1, 2, 5, 6\} \implies x_1 + x_2 + x_5 + x_6 \leq 3 \text{ is valid for } P_I. \end{aligned} \tag{27.5}$$

Theorem 4. *Let $C \subseteq N$ be a cover. Then $\sum_{j \in C} x_j \leq |C| - 1$ is valid for $P_I(N, a, b)$ and is known as a* cover inequality*.*

Next, we see a simple way to strengthen the cover inequality:

Back to the running example. We have seen that $C = \{3, 4, 5, 6\}$ is a cover, therefore,

$$x_3 + x_4 + x_5 + x_6 \leq 3$$

is valid for P_I.

Observation 2. The inequality

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3 \tag{27.6}$$

is also valid for P_I. This is because the coefficients of x_1 and x_2 are at least as large as the coefficient of x_3 in the knapsack constraint. This leads to the following definition of an extended cover:

Definition 5. Let C be a cover. The *extended cover* of C is

$$E(C) := C \cup \{j \in N : a_j \geq a_i \ \forall i \in C\}.$$

Theorem 6. *The* extended cover inequality $\sum_{j \in E(C)} x_j \leq |C| - 1$ *is valid for $P_I(N, a, b)$.*

Note that the family of extended cover inequalities dominates the family of cover inequalities.

Can we strengthen extended cover inequalities further?

Back to the running example. We obtain another valid inequality for P_I that dominates the extended cover inequality (27.6).

Observation 3. The inequality $2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$ is valid for P_I. (Note that this dominates the extended cover inequality (27.6).)

Proof. We ignore x_2 and explain why the choice of 2 as the coefficient of x_1 in the above inequality gives a valid inequality for P_I.

The inequality $x_3 + x_4 + x_5 + x_6 \leq 3$ is valid for $\left\{x \in \{0,1\}^4 : 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\right\}$. We would like to determine α_1 such that

$$\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3 \tag{27.7}$$

is valid for $\left\{x \in \{0,1\}^5 : 11x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\right\}$.

- If $x_1 = 0$, then (27.7) is valid for $\left\{x \in \{0,1\}^5 : 11x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\right\}$ for all α .
- If $x_1 = 1$, then (27.7) is valid for $\left\{x \in \{0,1\}^5 : 11x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\right\}$ iff $\alpha_1 + x_3 + x_4 + x_5 + x_6 \leq 3$ is valid for

  $$\left\{x \in \{0,1\}^4 : 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 8\right\}$$

  iff $\alpha_1 \leq 3 - z$ where

  $$z = \max\left\{x_3 + x_4 + x_5 + x_6 : 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 8, x \in \{0,1\}^4\right\}.$$

We note that the last problem z above is in fact a knapsack problem again. The solution $(0, 0, 0, 1)$ is a feasible solution for this new knapsack problem so $z \geq 1$. Also, no two items can fit in the knapsack so $z \leq 1$. Therefore, the optimum value is $z = 1$ and hence $\alpha_1 \leq 2$. Taking $\alpha_1 = 2$ gives the strongest inequality. □

The above technique is called *lifting*. We lifted an inequality with few variables into an inequality with larger number of variables. Such lifting is possible by understanding local structures in the IP. More generally, we can lift a cover inequality into all variables. For this, we would like to find best possible values α_j for all $j \in N \setminus C$ such that $\sum_{j \in N \setminus C} \alpha_j x_j + \sum_{j \in C} x_j \leq |C| - 1$ is valid for $P_I(N, a, b)$. Generalizing the above proof technique, we obtain the following lifting algorithm:

Algorithm 1: Lifting Algorithm

find an ordering $j_1, \ldots, j_r$ of $j \in N \setminus C$;
for $t = 1, \ldots, r$ **do**
  solve $z = \max\{\sum_{i=1}^{t-1} \alpha_j x_j + \sum_{j \in C} x_j : \sum_{i=1}^{t-1} a_j x_j + \sum_{j \in C} a_j x_j \leq b - a_{j_t}\}$;
  set $\alpha_{j_t} = |C| - 1 - z$

In Algorithm 1, at each round t, we suppose $\sum_{i=1}^{t-1} \alpha_j x_j + \sum_{j \in C} x_j \leq |C| - 1$ has been obtained so far. Then, in order to find α_{j_t} such that $\alpha_{j_t} x_{j_t} + \sum_{i=1}^{t-1} \alpha_j x_j + \sum_{j \in C} x_j \leq |C|_1$ is valid for $P_I(N, a, b)$, we solve a knapsack problem. Note that the algorithm is sensitive to the ordering of processing the variables.

The lifting technique is applicable in more general settings and at times leads to facet-defining inequalities. We describe a case under which it leads to facet-defining inequalities for $P_I(N, a, b)$.

Definition 7. A cover C is *minimal* if $C \setminus \{j\}$ is not a cover for all $j \in C$.

Theorem 8. *Let C be a* minimal *cover for $P_I(N, a, b)$. Then Algorithm 1 gives a facet-defining inequality for P_I.*

How fast can we generate a violated cover/extended cover inequality?

- Separation problem for the family of cover inequalities requires solving a knapsack type problem on fewer variables. This is typically solved using DP or heuristics.
- Separation problem for the family of extended cover inequalities requires solving a sequence of knapsack type problems. Again, this is done using DP or heuristics.

27.3 Conclusion

In this course, we touched various aspects of solving IPs. We saw well-structured IPs that enable efficient solutions—as special cases, we saw the linear equations integer feasibility problem, total unimodularity, and total dual integrality. We also saw efficient solving techniques for well-structured IPs—primal, primal-dual, and dynamic programming. Next, we saw solving techniques for unstructured IPs—branch and bound and cutting plane. We saw various theoretical aspects of cutting planes including cut generation, closures, and rank.

Active directions for research include cut generation, closures, rank, *lift and project*, *hierarchy*, branch and bound, and lattice optimization problems. Understanding some of these aspects for well-structured problems arising from combinatorial optimization and exploiting these approaches to design exact/approximation algorithms are also of interest.