Integer Programming
ISE 418

Lecture 7 (Part 2)

Dr. Ted Ralphs

Reading for This Lecture

- Nemhauser and Wolsey Sections II.3.1, II.3.6, II.4.1, II.4.2, II.5.4
- Wolsey Chapter 7
- CCZ Chapter 1
- "Constraint Integer Programming," Achterberg, Chapter II

Interpreting: Two Conceptual Reformulations

- From what we have seen so far, we have two conceptual reformulations of a given integer optimization problem.
- The first is in terms of *disjunction*:

$$\max \left\{ c^\top x \mid x \in \left(\bigcup_{i=1}^{k} \mathcal{P}_i + \mathrm{intcone}\{r^1, \ldots, r^t\} \right) \right\} \qquad \text{(DIS)}$$

- The second is in terms of *valid inequalities*:

$$\max \left\{ c^\top x \mid x \in \mathrm{conv}(\mathcal{S}) \right\} \qquad \text{(CP)}$$

  where $\mathcal{S}$ is the feasible region.

- In principle, if we had a method for generating either of these reformulations, this would lead to a practical method of solution.
- Unfortunately, these reformulations are necessarily of exponential size in general, so there can be no way of generating them efficiently.

Another Generic Algorithmic Framework

- Many algorithms in optimization consist of the iterative solution of a certain relaxation or "dual".
- The relaxation or dual is improved dynamically until an optimality criterion is achieved.
- A simple algorithm for solving MILPs is to start by solving the LP relaxation to obtain

$$\hat{x} \in \operatorname{argmax}_{x \in \mathcal{P}} c^\top x$$

  and the upper bound $U = c^\top \hat{x} \geq z_{\mathsf{IP}}$
- Then determine either a valid disjunction or a valid inequality that is *violated* by $\hat{x}$ and "add" it to the relaxation.
- Re-solve the strengthened relaxation and continue this process until $U = z_{\mathsf{IP}}$ (the solution to the relaxation is in $\mathcal{S}$).
- This vague algorithm is, at a high level, how we solve MILPs and we will see that branch-and-bound fits into this framework.
- The condition that $U = z_{\mathsf{IP}}$ is the basic optimality condition used in a wide range of optimization algorithms.

Optimality Conditions

- Let us now consider an MILP (A, b, c, p) with feasible set $\mathcal{S} = \mathcal{P} \cap (\mathbb{Z}^p_+ \times \mathbb{R}^{n-p}_+)$.
- Further, let $\{X_i\}_{i=1}^k$ be a linear disjunction valid for this MILP so that $X_i \cap \mathcal{P} \subseteq \mathbb{R}^n$ is polyhedral.
- Then $\max_{X_i \cap \mathcal{S}} c^\top x$ is an MILP for all $i \in 1, \ldots, k$.
- For each $i = 1, \ldots, k$, let $\mathcal{P}_i$ be a polyhedron such that $X_i \cap \mathcal{S} \subseteq \mathcal{P}_i \subseteq \mathcal{P} \cap X_i\}$.
- In other words, $\mathcal{P}_i$ is a valid formulation for subproblem i, possibly strengthened by additional valid inequalities.
- Note that $\{\mathcal{P}_i\}_{i=1}^k$ is itself a valid linear disjunction.
- We will see why there is a distinction between X_i and $\mathcal{P}_i$ later on.
- Conceptually, we are combining and relaxing the formulations (CP) and (DIS).

Optimality Conditions (cont'd)

- From the disjunction on the previous slide, we obtain a relaxation of a general MILP.
- This relaxation yields a practical set of optimality conditions.
- In particular,

$$\max_{i \in 1,\ldots,k} \max_{x \in \mathcal{P}_i \cap \mathbb{R}^n_+} c^\top x \geq z_{\mathsf{IP}}, \tag{1}$$

which implies that if we have $x^* \in \mathcal{S}$ such that

$$\max_{i \in 1,\ldots,k} \max_{x \in \mathcal{P}_i \cap \mathbb{R}^n_+} c^\top x = c^\top x^*, \tag{OPT}$$

then x^* must be optimal.

More on Optimality Conditions

- Although it is not obvious, these optimality conditions can be seen as a generalization of those from LP.
- They are also the optimality conditions implicitly underlying many advanced algorithms.
- There is an associated duality theory that we will see later.
- By parameterizing (1), we obtain a “dual function” that is the solution to a dual that generalizes the LP dual.

Termination Conditions

- Note that although we use multiple disjunctions to branch during the algorithm, the tree can still be seen as encoding a single disjunction.
- To see this, consider the set $\mathcal{T}$ of subproblems associated with the leaf nodes in the tree.
  - Provided that we use admissible disjunctions for branching, the feasible regions of these subproblems are a partition of $\mathcal{S}$.
  - Furthermore, we will see that there exists a collection of polyhedra $\{\mathcal{P}_i\}_{i\in\mathcal{T}}$, where
    - $\mathcal{P}_i$ is a formulation for subproblem i; and
    - $\{\mathcal{P}_i\}_{i=1}^k$ is admissible with respect to $\mathcal{S}$.
- When this disjunction, along with the best solution found so far satisfies the optimality conditions (OPT), the algorithm terminates.
- We will revisit this more formally as we further develop the supporting theory.

Ensuring Finite Convergence

- For LP-based branch and bound, ensuring convergence requires a convergent branching method.
- Roughly speaking, a convergent branching method is one which will
  - produce a violated admissible disjunction whenever the solution to the bounding problem is infeasible; and
  - if applied recursively, guarantee that at some finite depth, any resulting bounding problem will either
    - produce a feasible solution (to the original MILP); or
    - be proven infeasible; or
    - be pruned by bound.
- Typically, we achieve this by ensuring that at some finite depth, we have a complete description of the convex hull of solutions to the subproblem.
- This is always the case if we branch on variable disjunctions and $\mathcal{S}$ is bounded.
- We will also revisit this result more formally as we develop the supporting theory.