Integer Programming

Lecture 3

Alternative Formulations

- Recall our definition of a valid formulation from the last lecture.
- A key concept in the rest of the course will be that every mathematical model has many alternative formulations.
- Many of the key methodologies in integer programming are essentially automatic methods of reformulating a given model.
- The goal of the reformulation is to make the model easier to solve.
- There is a tradeoff between how difficult the reformulation itself is to perform and the effectiveness of the resulting simplification.
- Some reformulations may also dramatically increase the size of the problem description in their exact form.

Simple Example: Knapsack Problem

- We are given a set $N = \{1, \dots n\}$ of items and a capacity W.
- There is a profit p_i and a size w_i associated with each item $i \in N$.
- We want to choose the set of items that maximizes profit subject to the constraint that their total size does not exceed the capacity.
- The most straightforward formulation is to introduce a binary variable x_i associated with each item.
- x_i takes value 1 if item i is chosen and 0 otherwise.
- Then the formulation is

$$\begin{aligned} \max \sum_{j=1}^{n} p_j x_j & \\ \text{s.t.} \quad \sum_{j=1}^{n} w_j x_j \leq W & \\ x_i \in \{0, 1\} \qquad \forall i & \end{aligned}$$

- Is this formulation correct?

An Alternative Formulation

- Let us call a set $C \subseteq N$ a *cover* if $\sum_{i \in C} w_i > W$.
- Further, a cover C is *minimal* if $\sum_{i \in C \setminus \{j\}} w_i \leq W$ for all $j \in C$.
- Then we claim that the following is also a valid formulation of the original problem.

$$
\begin{aligned}
\max \; & \sum_{j=1}^{n} p_j x_j \\
\text{s.t.} \; & \sum_{j \in C} x_j \leq |C| - 1 && \text{for all minimal covers } C \\
& x_i \in \{0, 1\} && i \in N
\end{aligned}
$$

- Which formulation is "better"?

Compact Formulations

- A formulation is *compact* if the number of variables and constraints is polynomial in the "size" of the original problem description.
- This is only a rough definition, since the original problem may itself be described in multiple equivalent ways.
- To be more precise, we could say that the number of variables and constraints should be polynomial in the number of original "structural" variables.
- The second formulation for the knapsack problem is then not compact and this is a fundamental issue in solving MILPs in practice.
- Not all problems even have compact (linear) formulations.
- For example, we can prove that there is no compact formulation for optimization over the set of binary n-vectors with an even number of 1's.
- We will see other examples.

Back to the Facility Location Problem

- Recall our earlier formulation of this problem.
- Here is another formulation for the same problem:

$$\begin{aligned}
\min \sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} & \\
\text{s.t. } \sum_{j=1}^{n} x_{ij} = 1 & \qquad \forall i \\
x_{ij} \leq y_j & \qquad \forall i, j \\
x_{ij}, y_j \in \{0, 1\} & \qquad \forall i, j
\end{aligned}$$

- Notice that the set of integer solutions contained in each of the polyhedra is the same (why?).
- However, the second polyhedron is strictly included in the first one (how do we prove this?).
- Therefore, the second polyhedron will yield a better lower bound.
- The second polyhedron is a better approximation to the convex hull of integer solutions.

Formulation Strength and Ideal Formulations

- Consider two formulations A and B for the same MILP.
- Denote the feasible regions corresponding to their LP relaxations as $\mathcal{P}_A$ and $\mathcal{P}_B$.
- Formulation A is said to be *at least as strong as* (informally, we say "tighter than") formulation B if $\mathcal{P}_A \subseteq \mathcal{P}_B$.
- If the inclusion is strict, then A is *stronger than* B.
- If $\mathcal{S}$ is the set of all feasible integer solutions for the MILP, then we must have $conv(\mathcal{S}) \subseteq \mathcal{P}_A$ (why?).
- A is *ideal* if $conv(\mathcal{S}) = \mathcal{P}_A$.
- If we know an ideal formulation (of small enough size), we can solve the MILP (why?).
- How do our formulations of the knapsack problem compare by this measure?

Strengthening Formulations

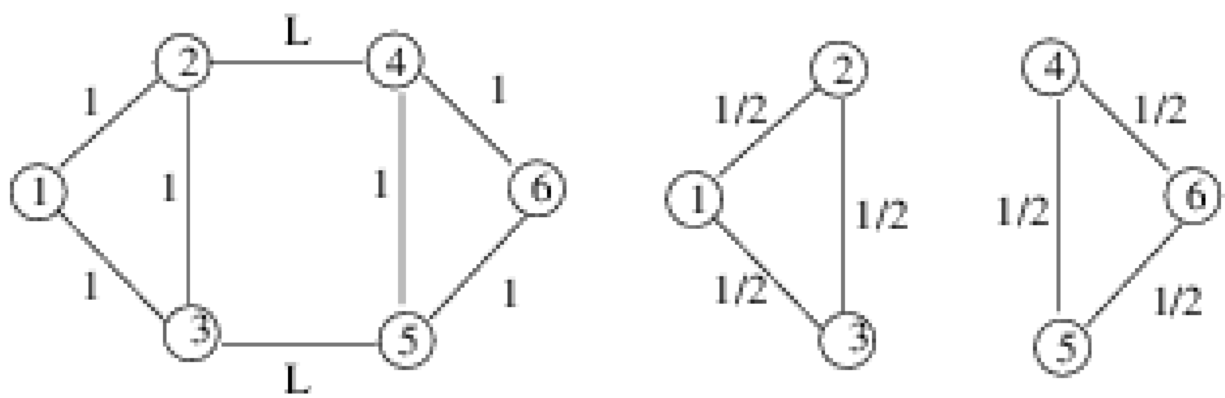
- Idea: Can we simply combine the two formulations for the knapsack problem to get the best of both worlds?
- Answer: Yes!
- Often, a given formulation can be strengthened with additional inequalities satisfied by all feasible integer solutions.
- We call these *valid inequalities* and will formally define the concept later in the course.
- As in the knapsack case, it is often easy to identify an exponential *class* of such inequalities.
- From a computational standpoint, the key is to only add the inequalities that are most "relevant."

Example

- Example: The Perfect Matching Problem
  - We are given a set of n people that need to be paired in teams of two.
  - Let c_{ij} represent the "cost" of the team formed by persons i and j.
  - We wish to minimize total cost of all assignment.
  - We can represent this problem on an undirected graph $G = (N, E)$.
  - The nodes represent the people and the edges represent pairings.
  - We have $x_e = 1$ if the endpoints of e are matched, $x_e = 0$ otherwise.

$$
\begin{aligned}
\min \quad & \sum_{e=\{i,j\}\in E} c_e x_e \\
\text{s.t.} \quad & \sum_{\{j \mid \{i,j\}\in E\}} x_{ij} = 1, \ \forall i \in N \\
& x_e \in \{0,1\}, \qquad\qquad \forall e = \{i,j\} \in E.
\end{aligned}
$$

Valid Inequalities for Matching



- Consider the graph on the left above.
- The optimal perfect matching has value $L + 2$.
- The optimal solution to the LP relaxation has value 3.
- This formulation can be extremely weak.
- Add the *valid inequality* $x_{24} + x_{35} \geq 1$.
- Every perfect matching satisfies this inequality.

The Odd Set Inequalities

- We can generalize the inequality from the last slide.
- Consider the cut S corresponding to any odd set of nodes.
- The *cutset* corresponding to S is

$$\delta(S) = \{\{i,j\} \in E | i \in S, j \notin S\}.$$

- An *odd cutset* is any $\delta(S)$ for which $|S|$ is odd.
- Note that every perfect matching contains at least one edge from every odd cutset.
- Hence, each odd cutset induces a possible valid inequality.

$$\sum_{e \in \delta(S)} x_e \geq 1, S \subset N, |S| \text{ odd}.$$

Using the New Formulation

- If we add all of the odd set inequalities, the new formulation is ideal.
- Hence, we can solve this LP and get a solution to the IP.
- However, the number of inequalities is exponential in size, so this is not really practical, i.e., the formulation is not compact.
- Recall that only a small number of these inequalities will be active at the optimal solution.
- Later, we will see how we can efficiently generate these inequalities on the fly to solve the IP.

Extended Formulations

- We have now seen two examples of strengthening formulations using additional constraints.
- However, changing the set of variables can also have a dramatic effect.
- We call a formulation with additional variables not appearing in the original model an "extended formulation."
- Example: A Lot-sizing Problem
  - We want to minimize the costs of production, storage, and set-up.
  - Data for period $t = 1, \ldots, T$:
    - d_t: total demand,
    - c_t: production set-up cost,
    - p_t: unit production cost,
    - h_t: unit storage cost.
  - Variables for period $t = 1, \ldots, T$:
    - y_t: prodution quantity
    - s_t: storage quantity
    - x_t: a binary variable, whether to produce

Lot-sizing: The "natural" formulation

- Here is the formulation based on the "natural" set of variables:

$$
\begin{aligned}
\min \sum_{t=1}^{T} & (p_t y_t + h_t s_t + c_t x_t) \\
\text{s.t. } & y_1 = d_1 + s_1, \\
& s_{t-1} + y_t = d_t + s_t, \qquad \text{for } t = 2, \ldots, T, \\
& y_t \leq \omega x_t, \qquad \text{for } t = 1, \ldots, T, \\
& s_T = 0, \\
& s, y \in \mathbb{R}_+^T, \\
& x \in \{0, 1\}^T.
\end{aligned}
$$

- Here, $\omega = \sum_{t=1}^{T} d_t$, an upper bound on y_t.

Lot-sizing: The "extended" formulation

- Suppose we split the production lot in period t into smaller pieces.
- Define the variables q_{it} to be the production in period i designated to satisfy demand in period $t \geq i$.
- Now, $y_i = \sum_{t=i}^{T} q_{it}$.
- With the new set of variables, we can impose the tighter constraint

$$q_{it} \leq d_t x_i \text{ for } i = 1, \ldots, T \text{ and } t = 1, \ldots, T.$$

- The additional variables strengthen the formulation.
- Again, this is contrary to conventional wisdom for formulating linear programs.

Strength of Formulation for Lot-sizing

- Although the formulation from the previous slide is much stronger than our original, it is still not ideal.
- Consider the following sample data.

```
# The demands for six periods
DEMAND = [6, 7, 4, 6, 3, 8]

# The production cost for six periods
PRODUCTION_COST = [3, 4, 3, 4, 4, 5]

# The storage cost for six periods
STORAGE_COST = [1, 1, 1, 1, 1, 1]

# The set up cost for six periods
SETUP_COST = [12, 15, 30, 23, 19, 45]

# Set of periods
PERIODS = range(len(DEMAND))
```

Strength of Formulation for Lot-sizing (cont'd)

```
Optimal Total Cost is:  171.42016761

Period  0 :  13 units produced,  7 units stored, 6 units sold
0.38235294 is the value of the fixed charge variable
Period  1 :  0 units produced,  0 units stored, 7 units sold
0.0 is the value of the fixed charge variable
Period  2 :  4 units produced,  0 units stored, 4 units sold
0.19047619 is the value of the fixed charge variable
Period  3 :  6 units produced,  0 units stored, 6 units sold
0.35294118 is the value of the fixed charge variable
Period  4 :  11 units produced, 8 units stored, 3 units sold
1.0 is the value of the fixed charge variable
Period  5 :  0 units produced,  0 units stored, 8 units sold
0.0 is the value of the fixed charge variable
```

- In period 0, it appears that we produced the full amount required to satisfy demand, but the fixed charge variable doesn't have value 1.
- What is happening here?

Strength of Formulation for Lot-sizing (cont'd)

Let's take a more detailed look:

```
production in period 0 for period 0 : 2.2941176
production in period 0 for period 1 : 2.6764706
production in period 0 for period 2 : 1.5294118
production in period 0 for period 3 : 2.2941176
production in period 0 for period 4 : 1.1470588
production in period 0 for period 5 : 3.0588235
```

What is the problem?

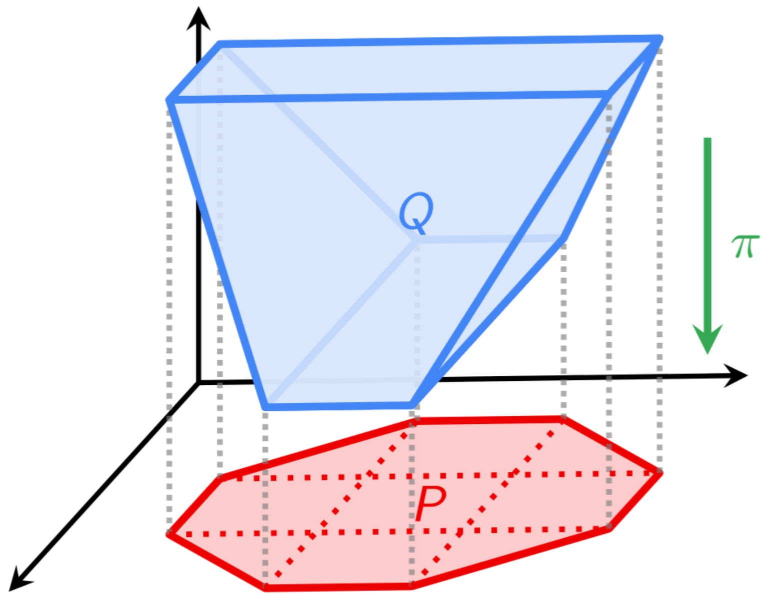
An Ideal Formulation for Lot-sizing

- We are only requiring that we have enough units on hand at time t to satisfy demand at time t.
- This was enough in the old formulation since units were not reserved for specific time periods.
- Now, some of the units we have on hand at time t may be reseved for sale in a future period.
- We can further strengthen the formulation by adding the constraint

$$\sum_{i=1}^{t} q_{it} \geq d_t \text{ for } t = 1, \ldots, T$$

- In fact, adding these additional constraints makes the formulation ideal.
- If we *project* into the original space, we will get the convex hull of solutions to the first formulation.
- How would we prove this?

Geometry of Extended Formulation



- By adding variables, we are "lifting" the formulation $\mathcal{P}$ into a higher-dimensional space to obtain $\mathcal{Q}$.
- When we project $\mathcal{Q}$ back into the original space, the resulting projected formulation is tighter, i.e., $\mathrm{proj}_x(\mathcal{Q}) \subset \mathcal{P}$.
- It is possible that the number of inequalities needed to describe $\mathcal{Q}$ is actually smaller than the number needed to describe $\mathcal{P}$.
- In some cases, the extended formulation is compact, whereas there is no compact formulation in the original space.

Contrast with Linear Programming

- In linear programming, the same problem can also have multiple formulations.
- In LP, however, conventional wisdom is that bigger formulations take longer to solve.
- In IP, this conventional wisdom does not hold.
- We have already seen two examples where it is not valid.
- Generally speaking, the size of the formulation does not determine how difficult the IP is.