MAT 3007 – Optimization

Integer Programming

Lecture 19

July 21th

Andre Milzarek

SDS / CUHK-SZ

Agenda

Announcements

Exercises:

- Exercise sheet 6 is online since Saturday; it's due on Sunday 26th, 11:00 am.
- One final exercise sheet (mainly on integer programming) will be uploaded on Thursday/Friday.

Final Exam:

- On-site: August 24 – September 5. (Or Wed., July 29th).
- Total Time: 3 hours.
- Each student can bring two self-made and handwritten sheets of A4 paper (with arbitrary notes on both sides of it). Copies and text that is created or generated by software or other tools are not allowed.

- Will post a sample final and review slides this week!

Agenda

So far we have only studied continuous optimization problems (both LP and NLP).

- Decision variables can take continuous values (any fractional values).

In the remaining two lectures, we are going to study discrete optimization problems.

- Decision variables can only take discrete values.
- We mainly consider integer linear programs.

Integer Programming

Integer Linear Program

An integer linear program (IP) is a linear program with the additional constraint that all variables must be integers:

$$\begin{aligned} \text{minimize}_x \quad & c^\top x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \\ & x \in \mathbb{Z}^n \end{aligned}$$

Here, we use $\mathbb{Z}$ to denote the set of integers.

- One may also encounter mixed integer programs (MIP), in which one set of variables must be integer and the rest are allowed to be continuous.

A Special Case: Binary IP

One subclass of IP is the binary integer program, in which each decision variable has to be either 0 or 1:

$$\begin{aligned} \text{minimize}_x \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & x_i \in \{0,1\} \quad \forall\ i. \end{aligned}$$

- The variables in a binary IP are also called indicator variables.

Integer Programs

In many real applications, the decision variables should actually be integers (or binary variables):

- Production planning.
- Shortest path.
- Generating mosaics.

In the previous discussions, we ignored those integral constraints for the ease of analysis.

However, it is important to study such problems since they capture many practical situations.

Examples & Modeling

Example: Knapsack Problem

John is planning a trip. There are n items he would like to bring.

- The ith item has value v_i.
- The weight of ith item is a_i.
- His bag has a maximum allowable weight C.
- He wants to bring as much value as possible.

Decision Variables:

- x_i: whether to bring ith item or not: $x_i \in \{0, 1\}$.

Optimization Problem:

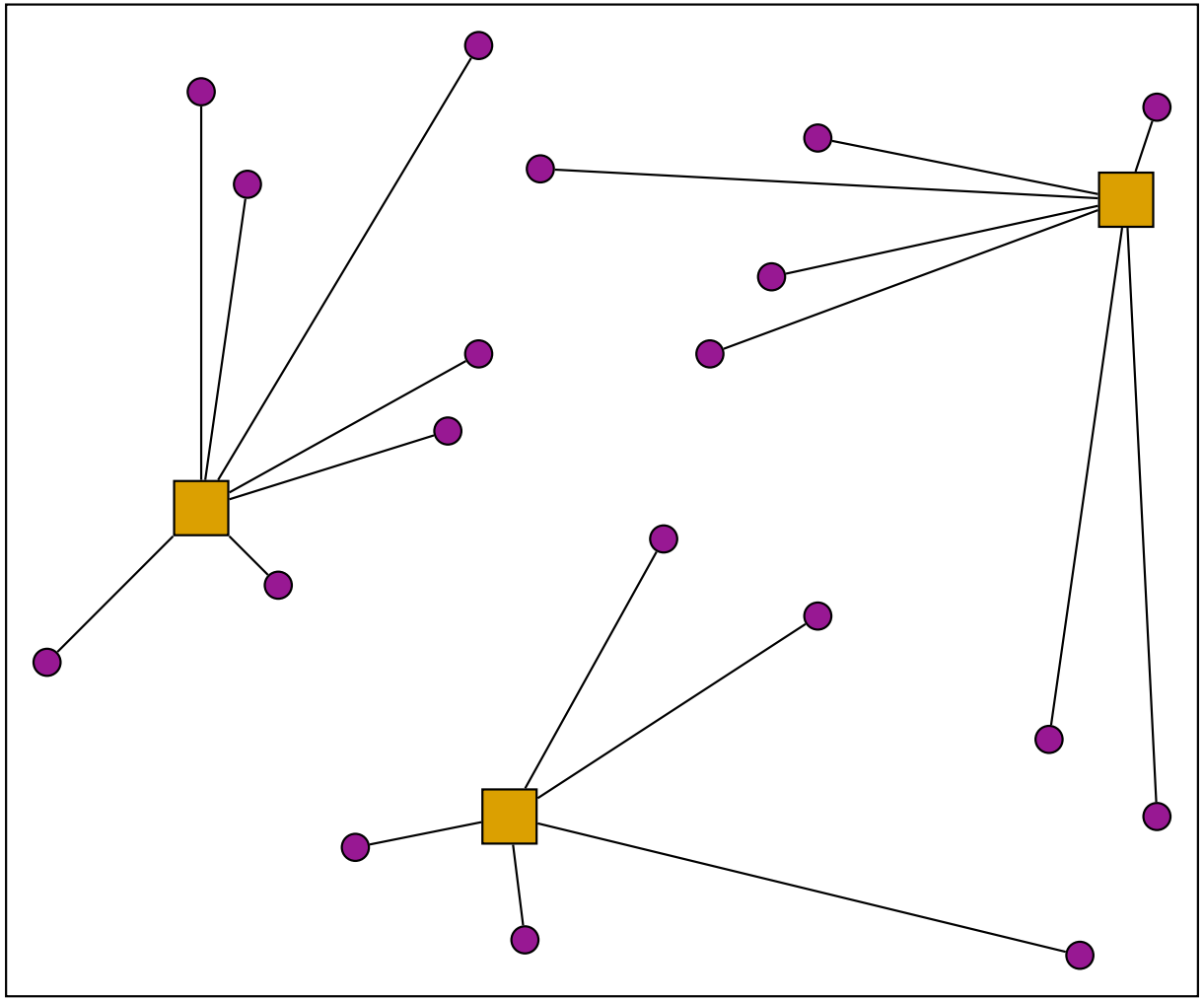
$$\begin{aligned} \text{maximize}_x \quad & \sum_{i=1}^n v_i x_i \\ \text{subject to} \quad & \sum_{i=1}^n a_i x_i \leq C \\ & x_i \in \{0, 1\} \quad \forall\ i. \end{aligned}$$

$\rightsquigarrow$ This is a binary optimization problem.

Example: Warehouse/Facility Location Problem

- There are n warehouses available for use.
- Opening warehouse i has a fixed operating costs f_i.
- There are m customers.
- Customer j has demand d_j. It costs c_{ij} to ship one unit of product from warehouse i to customer j.
- The objective is to satisfy all customers' demands, while minimizing the total costs (operating + shipment).

Facility Location Problem



Formulation

Decision Variables:

- y_i: whether to open warehouse i or not: $y_i \in \{0, 1\}$.
- x_{ij}: how many units to ship from warehouse i to customer j.

Objective function:

$$\text{minimize}_{x,y} \quad \sum_{i=1}^{n} f_i y_i + \sum_{i,j} c_{ij} x_{ij}$$

Constraints:

- Need to satisfy every customer's demand: $\sum_{i=1}^{n} x_{ij} \geq d_j$.
- A location can provide supply only if it is opened.
- To capture this, we can define constraints (why?):

$$x_{ij} \leq d_j y_i \quad \text{and} \quad x_{ij} \geq 0.$$

Full Formulation

We can formulate the warehouse/facility location problem as follows:

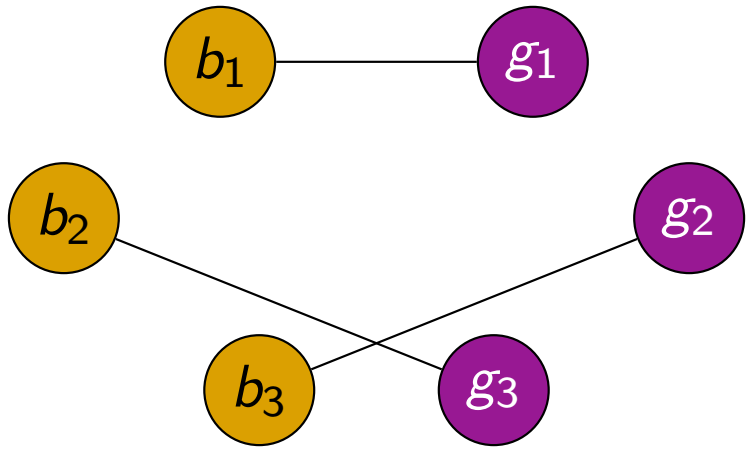
$$
\begin{aligned}
\text{minimize}_{x,y} \quad & \textstyle\sum_{i=1}^{n} f_i y_i + \sum_{i,j} c_{ij} x_{ij} & \\
\text{subject to} \quad & \textstyle\sum_{i=1}^{n} x_{ij} \geq d_j, & \text{for all } j \\
& x_{ij} \leq d_j y_i, & \text{for all } i, j \\
& x_{ij} \geq 0 & \\
& y_i \in \{0, 1\} &
\end{aligned}
$$

$\rightsquigarrow$ This is a mixed integer program.

Example: Matching Problem

Setup: Assume there are n girls and n boys' information in a dating website.

- Each girl i has rated boy j with a score v_{ij}.
- The website wants to match one girl with one boy so as to maximize the total score of the matching.



Matching Problem

Decision Variables:

- x_{ij}: whether girl i is matched with boy j.

Objective Function:

$$\text{maximize} \sum_{i=1}^{n} \sum_{j=1}^{n} v_{ij} x_{ij}$$

Constraints:

- Each girl is matched to exactly one boy: $\sum_{j=1}^{n} x_{ij} = 1$.
- Each boy is matched to exactly one girl: $\sum_{i=1}^{n} x_{ij} = 1$.
- The assignment must be integers: $x_{ij} \in \{0, 1\}$.

Matching Problem: Full Formulation

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} v_{ij}x_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} = 1 \quad \forall\, j, \quad \sum_{j=1}^{n} x_{ij} = 1 \quad \forall\, i \\ & x_{ij} \in \{0,1\} \end{aligned}$$

It is a binary optimization problem.

- It is also used in e.g., search engine adwords allocation problem, object tracking, and many other useful problems.
- Mosaic generation.

Solving Integer Programs

Our Objective

We have seen that integer optimization problems arise naturally from many applications.

We want to answer two questions in our study:

- How different is it from linear optimization problems?
- How to solve IPs? Are there any interesting properties?

We can give a quick answer to the first question:

- It is very different!
- Remember we have defined the complexity class NP-hard to denote a class of hard problems. Integer problems are NP-hard. It is unlikely to have polynomial-time algorithms.
- Even to verify if a set (polyhedron) contains an integral point is very hard.

An Intuitive Thought: Relaxation

General Idea:

Maybe we can first relax the integer constraints, solve the linear program (which is easy) and find some integer solution near the optimal solution of the linear program.

- We call the corresponding linear program the LP relaxation of the IP.
- Definitely a plausible method.
- May work well in some case.
- But may fail in other cases.

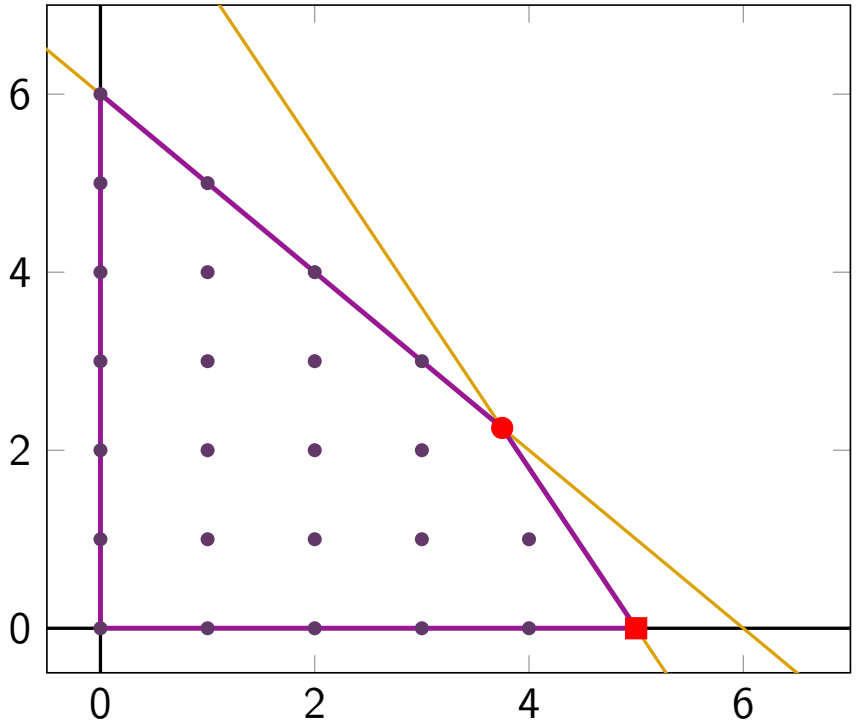
Example: Relaxation

Consider the following integer program (a production planning problem):

$$\begin{aligned} \text{maximize} \quad & 8x_1 + 5x_2 \\ \text{subject to} \quad & 9x_1 + 5x_2 \leq 45 \\ & x_1 + x_2 \leq 6 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in \mathbb{Z} \end{aligned}$$

- If we consider the LP relaxation, the optimal solution is $(x_1, x_2) = (15/4, 9/4)$.

Example: Relaxation



- If we try to round it to the nearest integer solution, we get $(4, 2)$ which is not even feasible for the integer program.
- If we consider the nearest feasible integer point, it will be $(3, 2)$, the objective value is 34.
- The optimal solution to the integer program is actually $(5, 0)$, the objective value is 40.

Another Example: Knapsack Problem

There are three items:

- Item A: 3kg with value 300.
- Item B: 2kg with value 180.
- Item C: 2kg with value 170.

The capacity of the bag is 4kg.

- The relaxation of a binary constraint $x \in \{0, 1\}$ is $x \in [0, 1]$.
- Optimal solution to LP relaxation: (1, 1/2, 0) with optimal value 390.
- Nearest feasible integer solution: (1, 0, 0) with value 300.
- Optimal integer solution: (0, 1, 1) with value 350.

Summary

The optimal solution to the integer program might not be close to the optimal solution to the LP relaxation.

- Directly rounding the LP solution to integers may not yield a good solution ($\rightsquigarrow$ might even be infeasible).
- The reason is the very combinatorial nature of the IP problem (no continuity in the solution space).

But does the LP relaxation provide any useful information?

LP Relaxation as a Bound for IP

Theorem: LP Relaxation as a Bound for IP

1. For a maximization problem, the optimal value of the LP relaxation provides an upper bound for the optimal value of the IP.
2. For a minimization problem, the optimal value of the LP relaxation provides a lower bound for the optimal value of the IP.

The difference between the optimal value of the LP and the IP is called the integrality gap.

- For maximization problems, the integrality gap is $v^{LP} - v^{IP}$.
- For minimization problems, the integrality gap is $v^{IP} - v^{LP}$.

Use LP Relaxation as a Bound

Maximization Problem: Suppose we solved the LP relaxation and obtain the optimal value is v^{LP} (with some fractional solution).

$\rightsquigarrow$ Then we round the solution to a feasible integer point and find the objective value is $v^{Rounding}$.

Then the difference between the rounded solution and the optimal integer solution (whose objective value is v^{IP}) satisfies:

$$0 \leq v^{IP} - v^{Rounding} \leq v^{LP} - v^{Rounding}$$

That is, one can use the LP relaxation solution to construct a bound on how good a certain feasible integer point is.

The Ideal Case

There are some cases where solving the LP relaxation already solves the integer problem.

Theorem

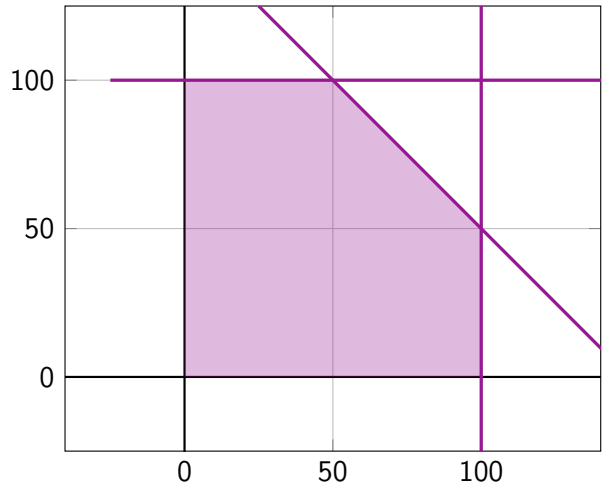
If the optimal solution to the LP relaxation is integer, then the solution must be optimal to the IP problem.

LPs and Integer Solutions

Recap: For LPs, there must exist an optimal solution that is a basic feasible solution. (If the problem has a solution)

- If we can guarantee that every BFS is an integer point, then the LP must have an integer optimal solution.

For example, in the following production planning problem:



All the extreme points are integers. Therefore, there must exist an optimal solution that is an integer point.

- This criterion has nothing to do with the objective function.

Total Unimodularity

We introduce a condition under which all BFS must be integers – total unimodularity.

Definition: Total Unimodulartiy

A matrix A is said to be totally unimodular if the determinant of each submatrix of A is either 0, 1, or -1.

Theorem: Total Unimodularity and Integer Solutions

If the constraint matrix A is totally unimodular and b is an integer vector, then all the BFS are integers and the LP relaxation must have an optimal solution that is an integer solution.

Total Unimodularity (TU):

- In order for A to be TU, it can only have 0, 1 or -1 entries.
- But it requires much more!

One Sufficient Condition for TU

Theorem

Let A be an $m \times n$ matrix. Then the following conditions together are sufficient for A to be totally unimodular:

1. Every column of A contains at most two non-zero entries;
2. Every entry in A is 0, $+1$, or -1;
3. The rows of A can be partitioned into two disjoint sets B and C such that:
   - If two non-zero entries in a column of A have the same sign, then the row of one entry is in B, and the row of the other in C;
   - If two non-zero entries in a column of A have opposite signs, then the rows are both in B, or both in C.

Example: Matching Problem

Recall the matching problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} v_{ij} x_{ij} & \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} = 1 & \forall\, j \\ & \sum_{j=1}^{n} x_{ij} = 1 & \forall\, i \\ & x_{ij} \in \{0,1\} & \end{aligned}$$

The constraint matrix is TU.

An example when $n = 2$ (in the order of $x_{11}, x_{12}, x_{21}, x_{22}$):

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Choosing $B = \{1, 2\}$, $C = \{3, 4\}$, the condition holds and A is TU.

Matching Problem: Observation

Therefore, to solve the matching problem, we can safely relax the binary constraints to $0 \leq x_{ij} \leq 1$.

It is guaranteed that there exists an integer optimal solution of the LP relaxation (and the simplex method can find it).

- This result also applies to transportation problems and assignment problems which have the same constraint matrix as the matching problem.
- There are some network flow problems which also have the TU property.

Example: Generating Mosaics

- The mosaic problem is an assignment problem and the constraints have the TU property.

Another Example

Consider the following matrix A:

$$A = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 \end{bmatrix}$$

Then we can make the partition to be $B = \{1, 2, 3, 4\}$, $C = \emptyset$. The condition holds and A is TU.

Discussion

The TU property is uncommon in practice.

$\rightsquigarrow$ If one solves the LP relaxation and if the optimal BFS is not integral, it means that the TU property does not hold.

- We need to have methods to solve IP systematically (and efficiently).
- The most commonly applied technique is the branch-and-bound method.

Questions?