MAT 3007 – Optimization

Branch-and-Bound

Lecture 20

July 23th

Andre Milzarek

SDS / CUHK-SZ

Repetition

Exercises:

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

Sample Final and Review:

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

LP Relaxation:

- Relax the integer constraints and solve a linear program.
- Find an integer solution near the optimal solution of the linear program.
- Solutions of the relaxed LP are bounds for the IP.
- If the optimal solution of the LP relaxation is an integer point, then is also optimal to the IP.

Total Unimodularity:

- If A is totally unimodular and b is an integer vector, then all BFS of the LP relaxation are integer points!
- Simplex method can recover the IP solution!
- The TU property is uncommon in practice.

Branch-and-Bound Method

Consider the following example:

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

- We solve the LP relaxation and get $x^* = (15/4, 9/4)$.
- x_1 is not an integer (we can not allow $15/4$ in the solution).

Solution: Consider two new subproblems:

- One with an additional constraint $x_1 \leq 3$.
- One with an additional constraint $x_1 \geq 4$.

The optimal solution to the IP must still be in one of the subproblems, but solutions with $3 < x_1 < 4$ are eliminated.

The General Process

First, we solve the LP relaxation of the IP:

- If the solution is an integer, then it is optimal to the IP.

If the optimal solution to the LP relaxation is x^* and $x_i^* \notin \mathbb{Z}$, then branch the problem into the following two:

1. One with an added constraint $x_i \leq \lfloor x_i^* \rfloor$, we call this (S1).
2. One with an added constraint $x_i \geq \lceil x_i^* \rceil$, we call this (S2).

Here $\lfloor \cdot \rfloor$ means rounding down, and $\lceil \cdot \rceil$ means rounding up:

- $\lfloor 3.75 \rfloor = 3$, $\lceil 3.75 \rceil = 4$.

What Happens after Branching?

We get two new IP's after branching:

- We then solve (S1) and (S2) and assume we can get the optimal solutions y_1^* and y_2^* with optimal values v_1^* and v_2^* (both (S1) and (S2) are still integer programs).

Claim

- If $v_2^* \leq v_1^*$, then y_1^* is the optimal solution to the original IP.
- If $v_1^* \leq v_2^*$, then y_2^* is the optimal solution to the original IP.

The claim is true because the union of the feasible regions in each branch equals the feasible region of the original problem.

Question: How to solve (S1) and (S2)?

$\rightsquigarrow$ Use the same idea (solve LP relaxation and further branch).

Bounding

For each branch, we can construct an upper bound and a lower bound for the problem (assume we are solving a max. problem):

- Upper bound: The LP relaxation solution will be an upper bound – the objective value of any integer solution from this node must be lower than the optimal value of the relaxed LP.
- Lower bound: The objective value of any feasible (integer) point is a lower bound for the optimal value – the optimal solution of the IP must be no less than the objective value achieved by any feasible point

Bounding Procedure:

- At a certain node, when the optimal value of the LP relaxation of this branch is even less than the current lower bound, then we can abandon this branch!
- These results will be the opposite if we are minimizing.

We will use the bounding steps to prune unnecessary computations (or in other words, remove unnecessary branches).

Example

Consider the earlier example

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

- We solve the LP relaxation and get $x^* = (15/4, 9/4)$.
- We first branch for x_1.

We solve two subproblems:

- One with an additional constraint $x_1 \leq 3 \rightsquigarrow$ problem (S1).
- One with an additional constraint $x_1 \geq 4 \rightsquigarrow$ problem (S2).

Example: Continued

Now we solve (S1) and (S2) respectively:

- We solve the LP relaxation of (S1). The optimal solution is $(3, 3)$. This is an integer solution, so we are done with this branch (the optimal value is 39).
- We solve the LP relaxation of (S2), the optimal solution is $(4, 1.8)$ (the optimal value is 41).

The solution to (S2) is not an integer. We have to do further branching:

- We add a constraint $x_2 \leq 1 \rightsquigarrow$ (S3).
- We add a constraint $x_2 \geq 2 \rightsquigarrow$ (S4).

Subproblem (S3):

$$
\begin{array}{rl}
\text{maximize} & 8x_1 + 5x_2 \\
\text{subject to} & 9x_1 + 5x_2 \le 45 \\
& x_1 + x_2 \le 6 \\
& x_1 \ge 4, x_2 \le 1 \quad x_1, x_2 \ge 0, \quad x_1, x_2 \in \mathbb{Z}
\end{array}
$$

Subproblem (S4):

$$
\begin{array}{rl}
\text{maximize} & 8x_1 + 5x_2 \\
\text{subject to} & 9x_1 + 5x_2 \le 45 \\
& x_1 + x_2 \le 6 \\
& x_1 \ge 4, x_2 \ge 2 \quad x_1, x_2 \ge 0, \quad x_1, x_2 \in \mathbb{Z}
\end{array}
$$

One can easily see that (S4) is not feasible. Thus we do not need to further consider this subproblem.

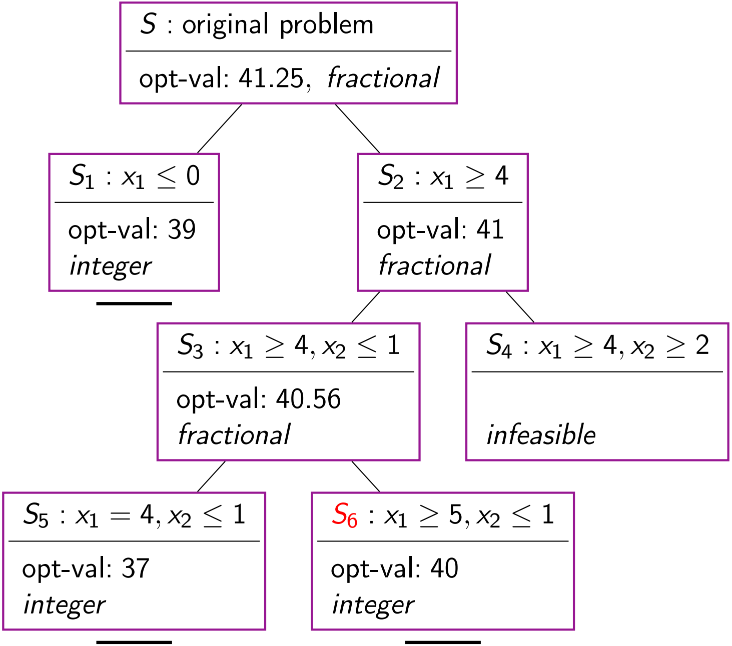
Solving (S3)

We solve (S3) and the optimal solution is $(40/9, 1)$, we have to do further branching:

- Add a constraint $x_1 \leq 4 \rightsquigarrow$ (S5).
- Add a constraint $x_1 \geq 5 \rightsquigarrow$ (S6).

For (S5), x_1 has to be 4 and the optimal solution is $(4, 1)$ with objective value 37 (already integer, so do not need to do further branching).

For (S6), the optimal solution is $(5, 0)$, the objective value is 40.


S : original problem
opt-val: 41.25, *fractional*
$S_1 : x_1 \le 0$
opt-val: 39
integer
$S_2 : x_1 \ge 4$
opt-val: 41
fractional
$S_3 : x_1 \ge 4, x_2 \le 1$
opt-val: 40.56
fractional
$S_4 : x_1 \ge 4, x_2 \ge 2$
infeasible
$S_5 : x_1 = 4, x_2 \le 1$
opt-val: 37
integer
$S_6 : x_1 \ge 5, x_2 \le 1$
opt-val: 40
integer

High-level idea:

- Branching: Divide the feasible region into smaller ones, solve each of them and combine them to find the optimal solution.
- Bounding: Use bounds (LP optimal value and feasible points) to reduce the number of branches we need to consider.

Branching Procedures:

1. Solve the LP relaxation.
   - If the optimal solution is integral, then it is optimal to IP.
   - Otherwise go to step 2.
2. If the optimal solution to the LP relaxation is x^* and x_i^* is fractional, then branch the problem into the following two:
   - One with an added constraint that $x_i \le \lfloor x_i^* \rfloor$.
   - One with an added constraint that $x_i \ge \lceil x_i^* \rceil$.
3. For each of the two problems, use the same method to solve them, and get optimal sol. y_1^* and y_2^* with optimal value v_1^* and v_2^*.
   - Compare to obtain the optimal solution.

Bounding procedures (for maximization):

- Any LP relaxation solution can provide an upper bound for each node in the branching process.
- Any feasible point to the IP can provide a lower bound for the entire problem.

When the optimal value of the LP relaxation of this branch is less than the current lower bound, then we can discard this branch.

- No better solution can be obtained from further exploring this branch.

Bounding is very important for branch-and-bound, it is the key to make it efficient (and practical).

Order of Computing the Branches

When we perform branch-and-bound, we may have two choices at each step:

- In the example, if we compute (S2) first, then we get a non-integer solution and thus two branches.
- Then we need to decide if we want to continue with one of the new branches or try (S1) next.

Basically, we need to decide if we want to go deep into one branch first or go wide to solve all problems on a given level.

Deep or Wide?

In the branch-and-bound algorithm, the best approach is to go deep into the tree, not to go wide:

- Most integer solutions lie deep in the tree. It is good to have integer feasible solutions early, so we can use it in the bounding procedure.
- It is also memory-efficient, since each LP is obtained from its parent by merely adding one constraint.
- It is also easier to code (recursion).

Branch-and-bound is essentially an enumeration method.

- In the worst case, branch-and-bound may need to go through each feasible integer point in the region, which is exponential in the problem size.
- Remember there is no polynomial-time algorithm for IP.

However, branch-and-bound does enumeration in a smart way and typically it only needs to visit a tiny fraction of all solutions.

- Much more efficient than explicitly enumerating the solutions.
- It is one of the most useful practical methods.

Branch-and-bound can also be used to solve binary linear programs

Consider the following knapsack problem:

$$\begin{aligned} \text{maximize} \quad & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14 \\ & x_j \in \{0,1\}, \quad j = 1, ..., 4. \end{aligned}$$

Example Continued

The LP relaxation for this IP is:

$$\begin{array}{rl} \text{maximize} & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & 0 \leq x_j \leq 1, \quad j = 1, ..., 4. \end{array}$$

In this case, the optimal solution is $x_1 = 1$, $x_2 = 1$, $x_3 = 0.5$, $x_4 = 0$. The optimal value is 22.

- We need to do branching for x_3.
- Consider two subproblems, one with $x_3 = 1$ ($\rightsquigarrow$ (S1)), the other with $x_3 = 0$ ($\rightsquigarrow$ (S2)).

Solving the LP relaxation (S1),

$$\begin{aligned} \text{maximize} \quad & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_3 = 1, \quad x_1, x_2, x_4 \in [0, 1]. \end{aligned}$$

we obtain the optimal solution $x_1 = 1$, $x_2 = 0.714$, $x_3 = 1$ and $x_4 = 0$. And the optimal value of the LP relaxation is 21.85.

- Still fractional. We need to do further branching.
  - $x_2 = 0 \rightsquigarrow$ (S3).
  - $x_2 = 1 \rightsquigarrow$ (S4).
- However, we obtained one important information: The optimal value of (S1) can not be better than 21.85.
- In fact, since the optimal value of (S1) must be an integer, therefore, it is at most 21.
- This is a trick often used in bounding.

Subproblem (S3): Consider the LP relaxation:

$$\begin{array}{rl} \text{maximize} & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} & 5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14 \\ & x_3 = 1, \quad x_2 = 0, \quad x_1, x_4 \in [0,1]. \end{array}$$

The optimal solution is $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, $x_4 = 1$. This is an integer solution with optimal value 18. Therefore, we are done with this branch!

- A lower bound of 18 is obtained.
- The optimal value of the original IP is at least 18.
- If we solve a later LP relaxation and the optimal value is less than 18, then we don't need to further consider that branch.

Example Continued

Subproblem (S4): Consider the LP relaxation:

$$\begin{array}{rl} \text{maximize} & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_3 = 1, \quad x_2 = 1, \quad x_1, x_4 \in [0,1]. \end{array}$$

The optimal solution is $x_1 = 0.6$, $x_2 = 1$, $x_3 = 1$, $x_4 = 0$ and the optimal value is 21.8.

- Still fractional. We need to do further branching.
  - $x_1 = 1 \rightsquigarrow$ (S5).
  - $x_1 = 0 \rightsquigarrow$ (S6).

Example Continued

Consider subproblem (S5):

$$\begin{array}{rl} \text{maximize} & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_3 = 1, \quad x_2 = 1, \quad x_1 = 1, \quad x_4 \in \{0, 1\}. \end{array}$$

It is easy to see that (S5) is infeasible. We do not need to further consider it.

Consider (S6):

$$\begin{array}{rl} \text{maximize} & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_3 = 1, \quad x_2 = 1, \quad x_1 = 0, \quad x_4 \in \{0, 1\}. \end{array}$$

The optimal solution is $x_1 = 0$, $x_2 = 1$, $x_3 = 1$ and x_4 =1. The optimal value is 21.

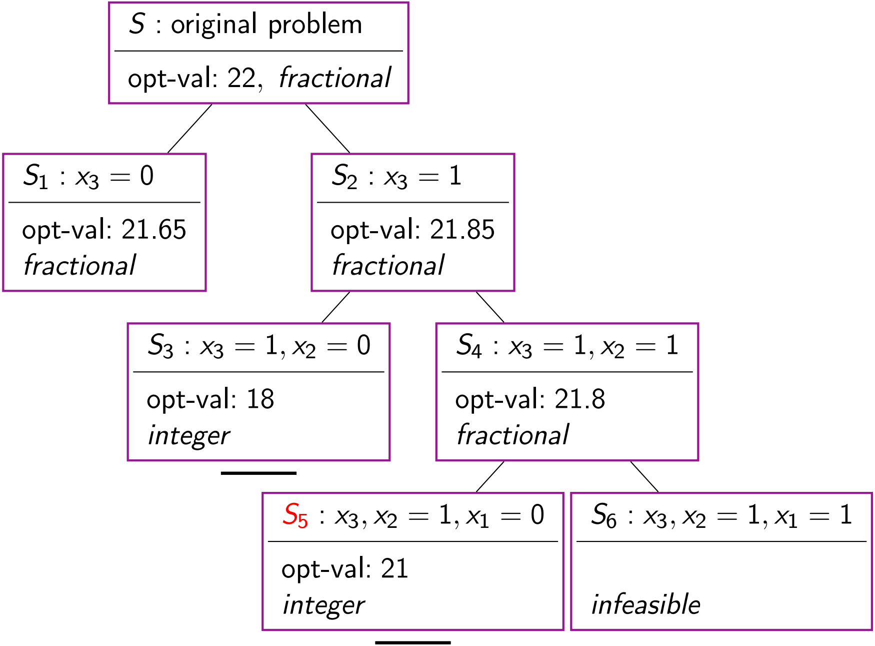
Therefore, we get 21 as the optimal value for the first branch (S1) with optimal solution $y^* = (0,1,1,1)^\top$.

Now consider (S2) and the LP relaxation:

$$\begin{aligned} \text{maximize} \quad & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14 \\ & x_3 = 0, \quad x_1, x_2, x_4 \in [0,1]. \end{aligned}$$

The optimal value is 21.65.

- 21.65 is an upper bound on this branch.
- Since the optimal value of (S2) must be integer, it means it can not be larger than 21.
- Therefore, no better solution can be obtained in this branch. We don't need to consider it!


S : original problem
opt-val: 22, *fractional*
$S_1 : x_3 = 0$
opt-val: 21.65
fractional
$S_2 : x_3 = 1$
opt-val: 21.85
fractional
$S_3 : x_3 = 1, x_2 = 0$
opt-val: 18
integer
$S_4 : x_3 = 1, x_2 = 1$
opt-val: 21.8
fractional
$S_5 : x_3, x_2 = 1, x_1 = 0$
opt-val: 21
integer
$S_6 : x_3, x_2 = 1, x_1 = 1$
infeasible

Example: Summary

There are 16 possible combinations in total, but we don't need to visit all of them.

- Bounding is very important, it can greatly reduce the search space.
- In the above example, we do not need to consider the $x_3 = 0$ branch because of bounding.

CVX can solve (mixed) integer programs with installed Gurobi solver (both are free for academic use):

- You can find the instructions online on the CVX website.
- You need to download Gurobi and obtain a Gurobi license.
- Install them properly (you need to follow the steps online).
- When using CVX, add the command "`cvx_solver gurobi`" at the top and the keyword "`integer`" or "`binary`" when declaring variables.

One can also use the MATLAB function: `intlinprog`.

We studied the most popular algorithm for solving integer programs – the branch-and-bound method.

- We can use this method to solve small-sized integer or binary programs.

Questions?