CS 524: Introduction to Optimization
Lecture 17 : Branch and Bound

Michael Ferris

Computer Sciences Department
University of Wisconsin-Madison

October 13, 2023

Discrete variables

- In many modeling situations some or all of the variables are constrained to take on only integer values.
- In addition to `positive` and `free variables`, GAMS has `integer variables` $x \in \mathbb{Z}$ and `binary variables` $x \in \{0,1\}$.
- Any model that involves discrete variables (and otherwise) linear constraints is a `mip` and not a linear program. The solve statement must be modified:

```
solve modname using mip minimizing obj;
```

- Such problems are much harder to solve as we now see.
- Note that replacing `mip` by `rmip` in the above solve statement relaxes the integer constraints so the variable are continuous and the resulting model is then a linear program (but is a "relaxation").

Semiint, semicont or sos (other discrete) variables

- Note that GAMS can use semiint or semicontinuous variables that are also discrete variables. See the "Variables" section of the user guide.
- Semicontinuous is a variable that is either zero, or else in a given range $[L, U]$. Define `semicont variable x;`, `x.lo = L; x.up = U;`
- Semiint is the same, except that there is also an integer restriction. It's a variable that is either zero or else an integer in the range $[L, U]$.
- Not all mip solvers allow these variable types so we often just model using binary and integer variables.
- Example: `dorian2.gms`
- There are also `sos1` and `sos2 variables` as we shall use later.

Integer Programs are hard!

- IP models can be *very* much more difficult to solve than LP models.

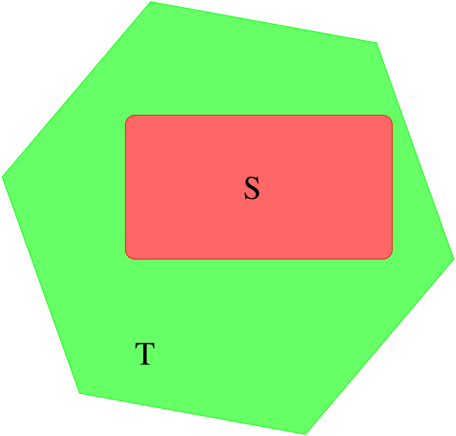
I'm not kidding

IP models can be *very* much more difficult to solve than LP models.

- It is important that you have a handle on...
  1. How to build a problem that is likely to be solved – Proper formulation is important!
  2. The general ideas of how integer programming problems are solved. Branch and Bound.

It's All About Relaxations

- Let $z_S = \max f(x) : x \in S$
- Let $z_T = \max f(x) : x \in T$



Question Time

- What can we say about z_S and z_T
- $z_S \leq z_T$?
- $z_S \geq z_T$?
- It depends?

Obvious, but Important Stuff

$$\boxed{z_T \geq z_S!!!}$$

- What if we get lucky?
- If x_T^* is an optimal solution to $\max f(x) : x \in T$
- And $x_T^* \in S$, then
- x_T^* is an optimal solution to $\max f(x) : x \in S$
- (If we were to replace max by min, then $z_T \leq z_S$. Everything is just flipped!)

LP relaxation

maximize

$$z_{IP} = c^T x$$

subject to

$$\begin{aligned} Ax &\geq b \\ x &\geq 0 \\ x_j &\in \mathbb{Z},\ j \in P \subseteq N. \end{aligned}$$

maximize

$$z_{LP} = c^T x$$

subject to

$$\begin{aligned} Ax &\geq b \\ x &\geq 0 \end{aligned}$$

- $z_{LP} \geq z_{IP}$
- If x^* solves the LP relaxation and x^* satisfies the integrality requirements ($x^* \in S$), then x^* solves IP

$x^* \notin S$?

- Then we branch!
- Partition the problem into smaller pieces.
- $x^* \notin S \Rightarrow \exists k \in P$ such that x_k^* is fractional.
- Create two new (restricted IP) problems...

1. In one problem, add the constraint $x_k \leq \lfloor x_k^* \rfloor$
2. In the other problem, add the constraint $x_k \geq \lceil x_k^* \rceil$

What Does All That Fancy Notation Mean?

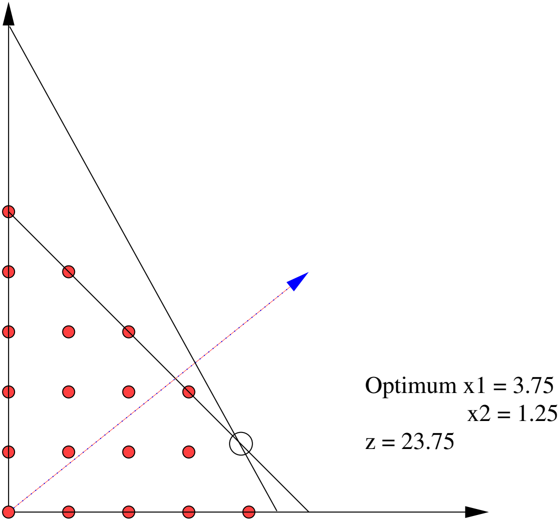
maximize

$$z = 5x_1 + 4x_2$$

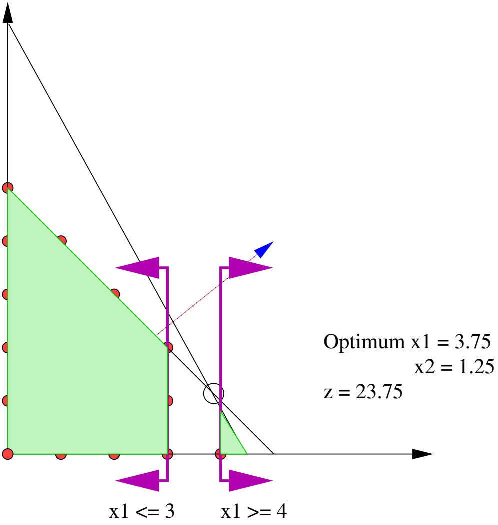
subject to

$$\begin{aligned} x_1 + x_2 &\leq 5 \\ 10x_1 + 6x_2 &\leq 45 \\ x_1, x_2 &\geq 0 \\ x_1, x_2 &\in \mathbb{Z} \text{ (must be integer valued)} \end{aligned}$$

The Feasible Region



Branching



The Branch and Bound Algorithm

- All of the following assumes a *maximization* problem.
- It works equally well for minimization, but you need to replace "lower" by "upper" (and vice verse) everywhere
- Let z_L be a lower bound on optimal objective value. Originally $z_L = -\infty$. The initial list of restricted IP problems to solve consists of just the entire problem.

Branch and Bound (1 of 2)

1. Select a restricted IP problem from the list of open problems.
2. Solve the LP relaxation of the restricted IP problem. If the LP relaxation is infeasible, Go to 1.
3. Otherwise, let x^* be the optimal solution to the LP relaxation and let z^* be its objective value.
4. If x^* satisfies the integer restrictions, then z^* is the optimal value for the restricted IP problem. If $z^* > z_L$, then $z_L := z^*$. (Also keep track of x^* as a candidate solution). Go to 1.

Branch and Bound (2 of 2)

5. Otherwise, x^* does not satisfy the integer restrictions. z^* is an upper bound on the optimal value of the restricted IP problem.
6. If $z^* \leq z_L$, (Fathom) Go to 1. (There is no way that a better optimal solution can be in this restricted region).
7. Otherwise, Divide. Choose some index k such that x_k^* does not satsify the integer restriction. Add two new restricted IP problems to the list:
    1. Include constraint $x_k \leq \lfloor x_k^* \rfloor$
    2. Include constraint $x_k \geq \lceil x_k^* \rceil$
8. Go to 1.

Bounds and performance guarantees

- We saw in the description of the algorithm how to get a lower bound z_L on the optimal objective function value
- We can also get an upper bound on the optimal objective value. z_U is the maximum of all of the upper bounds of all of the remaining restriced problems that must be evaluated
- If we only want to solve a problem to within $\alpha\%$ of optimality, we can!!!
- This is really what is great about optimization. Not only do you get a solution, but you can tell your boss that if there is a better solution it can be at most $\alpha\%$ better.

GAMS and Tolerances

- `option optcr = 0.0`
- `option optca = 0.0`

- `optcr`: Stop if $(z_U - z_L)/\max\{|z_L|, 1\} \leq$ `optcr`
- `optca`: Stop if $(z_U - z_L) \leq$ `optca`

Some Good CPLEX parameters

- mipemphasis (1 for solution finding, 2 for optimality)
- varsel (strong branching 3)
- cuts (-1 no, 5 aggressive),
- probe (-1 no, 3 full),
- lbheur (local branching heuristic),
- preslvnd,
- repeatepresolve,
- subalg,
- nodesel (0 dfs, 2 best estimate),
- heurfreq (-1 for node heuristic off)
- mipstart 1 (use an existing integer solution to start)
- numericalemphasis (1 for extreme caution)

Some Good Gurobi parameters

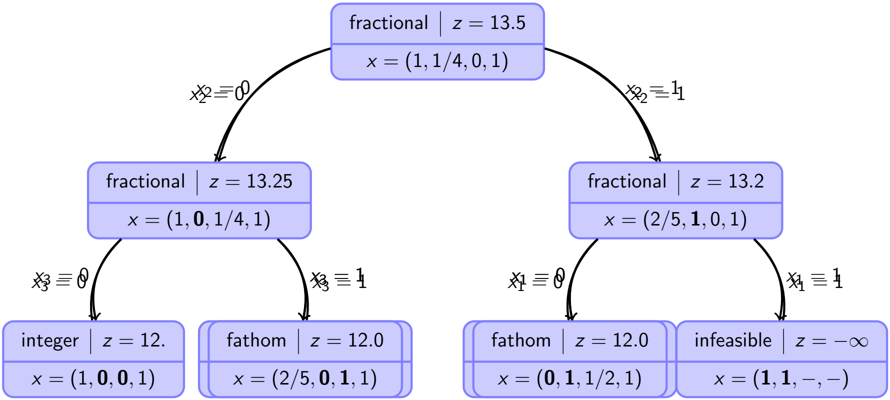
- cuts -1: auto, 0:off, 3:aggresive
- heuristics (fraction) % of time spent trying to find feasible solutions
- mipfocus: 1 for solution finding, 2 for optimality
- mipstart 1

For further details on options for CPLEX (and Gurobi): Refer to [Klotz and Newman(2013a)] for MIP and [Klotz and Newman(2013b)] for LP.

Simple knapsack example (b_and_b.gms)

$$\begin{gathered}\max_x 8x_1 + 6x_2 + 5x_3 + 4x_4 \\ 5x_1 + 4x_2 + 4x_3 + 2x_4 \leq 8 \\ x_1, x_2, x_3, x_4 \in \{0, 1\}\end{gathered}$$

Branch and Bound Process (`op.txt`)



- $z_L = -\infty 12.0$, $z_U = 13.513.212.0$
- Can round down objective since must be integer (all entries integer)

E. Klotz and A. M. Newman.
Practical guidelines for solving difficult mixed integer linear programs.
Surveys in Operations Research and Management Science, 18(1-2): 18–32, Oct. 2013a.
doi: 10.1016/j.sorms.2012.12.001.

E. Klotz and A. M. Newman.
Practical guidelines for solving difficult linear programs.
Surveys in Operations Research and Management Science, 18(1-2): 1–17, Oct. 2013b.
doi: 10.1016/j.sorms.2012.11.001.