How Does Excel Solve Integer Optimization?

- Unlike linear optimization, discrete optimization is typically quite challenging to solve.

- In general, to find the best solution, a solver may just need to enumerate over all possible solutions.
  - In our simple course-selection problem, there are $2^{10} = 1024$ possible course selections.
  - In real-world sized problems with millions of variables, there would be way too many solutions to enumerate.

- However, solvers are still routinely and efficient at finding **very good solutions, good enough for real-world use-cases.** *How do solvers do it?*

First, ignore the integrality constraints

Maximize: $10\ y_A + 2\ y_B + 4\ y_C + 2\ y_D + 5\ y_E$
$+ 4\ y_F + 8\ y_G + 7\ y_H + 6\ y_I + 6\ y_J$

over variables: $y_A, y_B, \ldots, y_J$

Subject To:

(binary)	$y_A, y_B, \ldots, y_J \geq 0, \leq 1$ ~~and integral~~
(points budget)	$200\ y_A + 50\ y_B + \ldots + 100\ y_J \leq 1000$
(max credits)	$12\ y_A + 9\ y_B + \ldots + 6\ y_J \leq 54$
(min credits)	$12\ y_A + 9\ y_B + \ldots + 6\ y_J \geq 36$
(MW H3 load)	$y_A + y_B + y_E + y_G \leq 3$
(MW H4 load)	$y_A + y_B + y_G + y_J \leq 3$
(TR H3 load)	$y_C + y_D + y_I \leq 3$
(TR H4 load)	$y_C + y_D + y_F + y_H + y_I \leq 3$
(A B conflict)	$y_A + y_B \leq 1$
(B or C required)	$y_B + y_C \geq 1$
(E pre-req to H)	$y_H \leq y_E$

Solve in Excel and check if the solution is integral

Maximize:	$10\, y_A + 2\, y_B + 4\, y_C + 2\, y_D + 5\, y_E$ $+ 4\, y_F + 8\, y_G + 7\, y_H + 6\, y_I + 6\, y_J$
over variables:	$y_A, y_B, \ldots, y_J$
Subject To:	
(binary)	$y_A, y_B, \ldots, y_J \geq 0, \leq 1$ ~~and integral~~
(points budget)	$200\, y_A + 50\, y_B + \ldots + 100\, y_J \leq 1000$
(max credits)	$12\, y_A + 9\, y_B + \ldots + 6\, y_J \leq 54$
(min credits)	$12\, y_A + 9\, y_B + \ldots + 6\, y_J \geq 36$
(MW H3 load)	$y_A + y_B + y_E + y_G \leq 3$
(MW H4 load)	$y_A + y_B + y_G + y_J \leq 3$
(TR H3 load)	$y_C + y_D + y_I \leq 3$
(TR H4 load)	$y_C + y_D + y_F + y_H + y_I \leq 3$
(A B conflict)	$y_A + y_B \leq 1$
(B or C required)	$y_B + y_C \geq 1$
(E pre-req to H)	$y_H \leq y_E$

Solve in Excel / Solver

Optimal Solution

$y_A = 1.0$
$y_B = 0.0$
$y_C = 1.0$
$y_D = 0.0$
$y_E = 1.0$
$y_F = 0.0$
$y_G = 1.0$
$y_H = 1.0$
$y_I = 0.0$
$y_J = 1.0$

If the solution has all integers, then we are done! Why?

Take one of the fractional variables and …

Maximize: $10\ y_A + 2\ y_B + 4\ y_C + 2\ y_D + 5\ y_E + 4\ y_F + 8\ y_G + 7\ y_H + 6\ y_I + 6\ y_J$

over variables: $y_A, y_B, \ldots, y_J$

Subject To:

(binary)	$y_A, y_B, \ldots, y_J \geq 0, \leq 1$ ~~and integral~~
(points budget)	$200\ y_A + 50\ y_B + \ldots + 100\ y_J \leq 1000$
(max credits)	$12\ y_A + 9\ y_B + \ldots + 6\ y_J \leq 54$
(min credits)	$12\ y_A + 9\ y_B + \ldots + 6\ y_J \geq 36$
(MW H3 load)	$y_A + y_B + y_E + y_G \leq 3$
(MW H4 load)	$y_A + y_B + y_G + y_J \leq 3$
(TR H3 load)	$y_C + y_D + y_I \leq 3$
(TR H4 load)	$y_C + y_D + y_F + y_H + y_I \leq 3$
(A B conflict)	$y_A + y_B \leq 1$
(B or C required)	$y_B + y_C \geq 1$
(E pre-req to H)	$y_H \leq y_E$

Solve in Excel / Solver

Optimal Solution

$y_A = 0.7$
$y_B = 0.0$
$y_C = 0.7$
$y_D = 0.0$
$y_E = 1.0$
$y_F = 0.3$
$y_G = 1.0$
$y_H = 1.0$
$y_I = 0.0$
$y_J = 1.0$

Branch into two optimization problems

Maximize: $10\ y_A + 2\ y_B + 4\ y_C + 2\ y_D + 5\ y_E + 4\ y_F + 8\ y_G + 7\ y_H + 6\ y_I + 6\ y_J$

over variables: $y_A, y_B, \ldots, y_J$

Subject To:

(binary) $y_A, y_B, \ldots, y_J \geq 0, \leq 1$ and <u>integral</u>

…

(E pre-req to H) $y_H \leq y_E$

Maximize: $10\ y_A + 2\ y_B + 4\ y_C + 2\ y_D + 5\ y_E + 4\ y_F + 8\ y_G + 7\ y_H + 6\ y_I + 6\ y_J$

over variables: $y_A, y_B, \ldots, y_J$

Subject To:

(binary) $y_A, y_B, \ldots, y_J \geq 0, \leq 1$ and <u>integral</u>

…

(E pre-req to H) $y_H \leq y_E$

(Fix $y_A = 1$) $y_A = 1$

Maximize: $10\ y_A + 2\ y_B + 4\ y_C + 2\ y_D + 5\ y_E + 4\ y_F + 8\ y_G + 7\ y_H + 6\ y_I + 6\ y_J$

over variables: $y_A, y_B, \ldots, y_J$

Subject To:

(binary) $y_A, y_B, \ldots, y_J \geq 0, \leq 1$ and <u>integral</u>

…

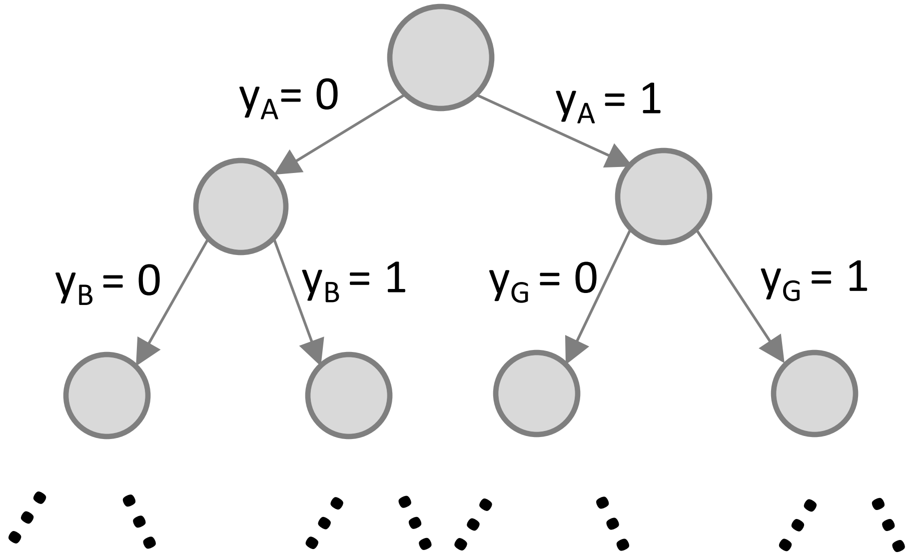
(E pre-req to H) $y_H \leq y_E$

(Fix $y_A = 0$) $y_A = 0$

We have turned 1 problem with 10 variables into 2 problems with 9 variables.

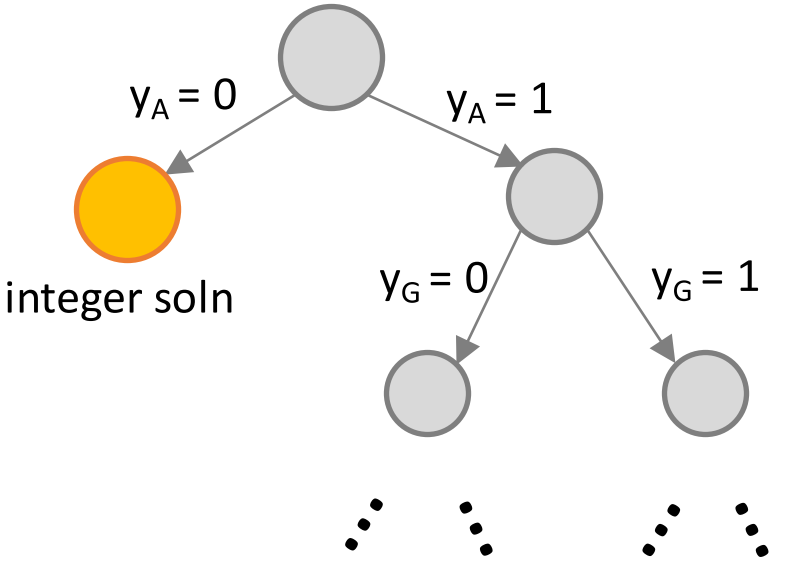
Take the *better* of the two optimal solutions.

Branching doubles the number of optimization problems we must solve.



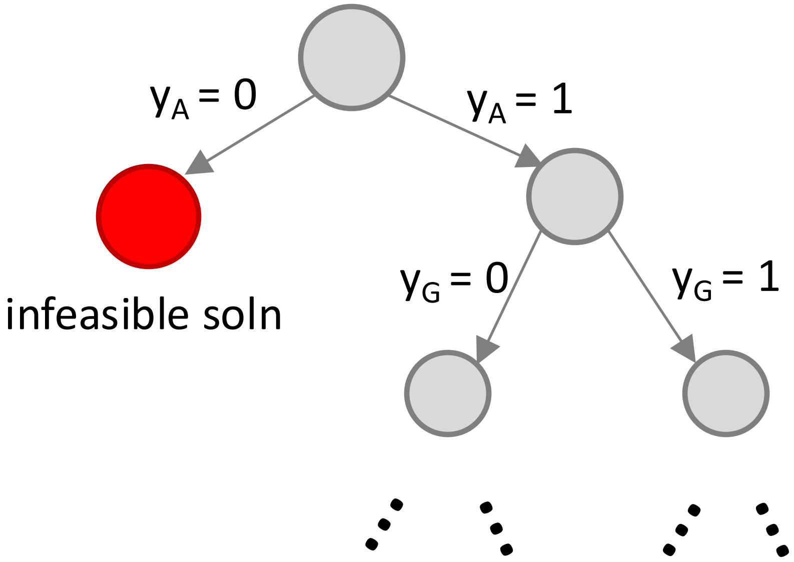
- We continue to branch, each time *doubling* the number of Linear Programs that we must solve. Yikes!

Ways that this enumeration is "smarter"



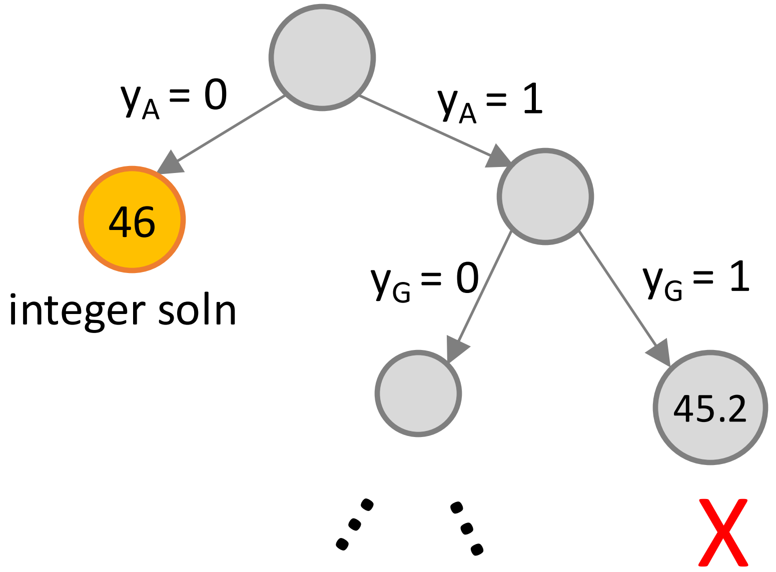
- Every time we hit an integer solution, we can stop branching.

Ways that this enumeration is "smarter"



- Every time we hit an integer solution, we can stop branching.
- If the linear optimization relaxation is *infeasible*, then we can also stop branching.

Ways that this enumeration is “smarter”



- Every time we hit an integer solution, we can stop branching.
- If the linear optimization relaxation is *infeasible*, then we can also stop branching.
- If the linear optimization relaxation has an objective value worse than an integer solution, then we can also stop branching.

This algorithm is called "Branch and Bound".