Mixed-Integer Linear Programming (MILP): Branch-and-Bound Search

Benoît Chachuat <benoit@mcmaster.ca>

McMaster University
Department of Chemical Engineering

ChE 4G03: Optimization in Chemical Engineering

Solving Discrete Optimization Problems

Standard Mixed-Integer Linear Programming (MILP) Formulation:

$$\min_{\mathbf{x},\mathbf{y}} \quad z \triangleq \mathbf{c}^\mathsf{T}\mathbf{x} + \mathbf{d}^\mathsf{T}\mathbf{y}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{y} \left\{ \begin{array}{c} \leq \\ = \\ \geq \end{array} \right\} \mathbf{b}$$

$$\mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}, \quad \mathbf{y} \in \{0,1\}^{n_y}$$

- We seek a rigorous solution
- The concept of local derivative information (or gradient) does not exist for discrete variables!
- The basic numerical optimization paradigm (improving search) applies only when we know/assume the values of **all** integer variables
- **We need a new approach for problems with integer variables!**

Solving Discrete Optimization Problems

- That discrete optimization models are more difficult to solve than continuous models may appear counter-intuitive
- After all, a discrete model only has a finite number of choices for decision variables!

Total Enumeration:

Solve a discrete optimization by trying all possible combinations and keep whichever is best

Class Exercise: Solve the following discrete optimization model by total enumeration:

$$\max_{\mathbf{y}} \quad 7y_1 + 4y_2 + 19y_3$$

$$\text{s.t.} \quad y_1 + y_2 \leq 2$$

$$y_2 + y_3 \leq 1$$

$$y_1, y_2, y_3 \in \{0,1\}$$

Case	Objective	Case	Objective

Solving Discrete Optimization Problems

Exponential Growth with Total Enumeration:

- For n binary variables,
  - $n = 10$:
  - $n = 20$:
  - $n = 30$:

1. With no more than a few discrete variables, total enumeration is often the most effective solution method
2. But, exponential growth makes total enumeration impractical with models having more than a handful of discrete decision variables

Back to the Drawing Board!

Solving Discrete Optimization Problems

New Paradigm for Discrete Optimization Search

Construct a sequence of related, simpler subproblems, the solutions of which converges (finitely) to the original solution

- Use relaxations to define the subproblems
  - E.g., relax the feasible region (LP relaxations) and/or the objective function (Lagrangian relaxations)
- The subproblems should be easier to solve than the original (since many may have to be solved)
- Each subproblem should yield a bound on the original optimal solution value:
  - Lower bound for a minimize problem
  - Upper bound for a maximize problem
- The number of subproblems solved should be much smaller than with the complete enumeration method

Outline

For additional details, see Rardin (1998), Chapter 12

Relaxations of Discrete Optimization Models

Constraint Relaxation

Model (R) is said to be a constraint relaxation of model (P) if:

- every feasible solution to (P) is also feasible in (R)
- (P) and (R) have the same objective function

Original MILP Model:

$$\begin{aligned} \min_{x,y} \quad & 7x_1 + x_2 + 3y_1 + 6y_2 \\ \text{s.t.} \quad & x_1 + 10x_2 + 2y_1 + y_2 \geq 100 \\ & y_1 + y_2 \leq 1 \\ & x_1, x_2 \geq 0, \quad y_1, y_2 \in \{0,1\} \end{aligned}$$

Relax. #1: Drop constraint

$$\begin{aligned} \min_{x,y} \quad & 7x_1 + x_2 + 3y_1 + 6y_2 \\ \text{s.t.} \quad & x_1 + 10x_2 + 2y_1 + y_2 \geq 100 \\ & x_1, x_2 \geq 0, \quad y_1, y_2 \in \{0,1\} \end{aligned}$$

Relax. #2: Relax constraint RHS

$$\begin{aligned} \min_{x,y} \quad & 7x_1 + x_2 + 3y_1 + 6y_2 \\ \text{s.t.} \quad & x_1 + 10x_2 + 2y_1 + y_2 \geq 50 \\ & y_1 + y_2 \leq 1 \\ & x_1, x_2 \geq 0, \quad y_1, y_2 \in \{0,1\} \end{aligned}$$

Relax. #3: Remove integrality

$$\begin{aligned} \min_{x,y} \quad & 7x_1 + x_2 + 3y_1 + 6y_2 \\ \text{s.t.} \quad & x_1 + 10x_2 + 2y_1 + y_2 \geq 100 \\ & y_1 + y_2 \leq 1 \\ & x_1, x_2 \geq 0, \quad 0 \leq y_1, y_2 \leq 1 \end{aligned}$$

Linear Programming Relaxations

LP Relaxations

LP relaxations of a MILP model are formed by treating any discrete variables as continuous, while retaining all other constraints:

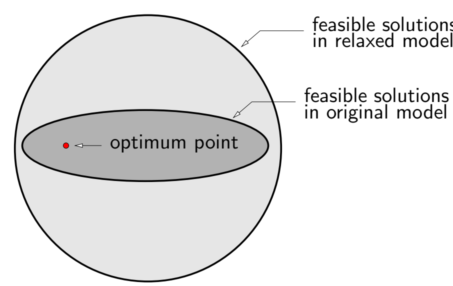
$$y_i \in \{0,1\} \quad \Longrightarrow \quad 0 \leq y_i \leq 1$$

Motivations:

- Bring all the power of LP to bear on analysis of discrete models
- By far the most used relaxation forms

Properties:

- Are LP relaxations guaranteed to yield valid relaxations?
  -



Properties of LP Relaxations (cont'd)

Proving Infeasibility with Relaxations

If an LP relaxation is infeasible, so is the MILP model it relaxes

Question: Can we conclude anything regarding the feasibility of the ILP model if the LP relaxation has a feasible solution?

-

Class Exercise: Use LP relaxations to help establish infeasibility of the following ILP models:

$$\begin{aligned} \min_{\mathbf{y}} \quad & 8y_1 + 2y_2 \\ \text{s.t.} \quad & y_1 - y_2 \geq 2 \\ & -y_1 + y_2 \geq -1 \\ & y_1, y_2, \in \{0, 1, 2, \ldots\} \end{aligned}$$

$$\begin{aligned} \min_{\mathbf{y}} \quad & y_1 + 2y_2 \\ \text{s.t.} \quad & 4y_1 + 2y_2 \geq 1 \\ & 4y_1 + 4y_2 \leq 3 \\ & y_1, y_2, \in \{0, 1\} \end{aligned}$$

Properties of LP Relaxations (cont'd)

Solution Bounds from LP Relaxations

- The optimal value of an LP relaxation of a maximize MILP model yields an upper bound on the optimal value of that model
- The optimal value of an LP relaxation of a minimize MILP model yields a lower bound

Class Exercise: Consider the following ILP model:

$$\begin{aligned} \max_{\mathbf{y}} \quad & y_1 + y_2 + y_3 \\ \text{s.t.} \quad & y_1 + y_2 \leq 1 \\ & y_1 + y_3 \leq 1 \\ & y_2 + y_3 \leq 1 \\ & y_1, y_2, y_3 \in \{0, 1\} \end{aligned}$$

1. Formulate and solve an LP relaxation of the ILP model (by inspection)
2. Solve the ILP model (by inspection) and compare the optimal values

Properties of LP Relaxations (cont'd)

Optimal Solutions from Relaxations

If an optimal solution to an LP relaxation is feasible in the MILP model it relaxes, the solution is optimal in that model too

Class Exercise: Solve LP relaxations for the following ILP models:

$$\begin{aligned} \max_{\mathbf{y}} \quad & y_1 + y_2 + y_3 \\ \text{s.t.} \quad & y_1 + y_2 \leq 1 \\ & y_1 + y_3 \leq 1 \\ & y_2 + y_3 \leq 1 \\ & y_1, y_2, y_3 \in \{0, 1\} \end{aligned}$$

$$\begin{aligned} \min_{\mathbf{y}} \quad & 20y_1 + 8y_2 + 3y_3 \\ \text{s.t.} \quad & y_1 + y_2 + y_3 \leq 1 \\ & y_1, y_2, y_3 \in \{0, 1\} \end{aligned}$$

Is the relaxation optimum also optimal in the MILP model?

Heuristics: LP relaxations may produce optimal solutions that are easily "rounded" to good feasible solution for the corresponding MILP model

Branch-and-Bound Search

Branch-and-Bound algorithms combine partial enumeration strategy with relaxation techniques:

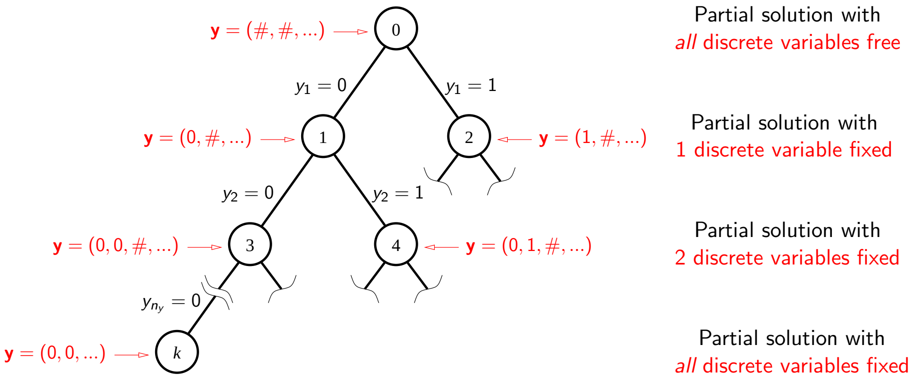
- Classes of solutions are formed and investigated to determine whether they can or cannot contain optimal solutions
- This search is conducted by analyzing associated relaxations
- Only promising classes are searched in further details

Partial Solutions and Completions

- A partial solution has some discrete decision variables fixed, while other left free (denoted by $\#$)
- The completions of a partial solution are the possible full solutions agreeing with the partial solution on all fixed variables

Example: $\mathbf{y} = (1, \#, 0, \#)$ is a partial solution with $y_1 = 1$ and $y_3 = 0$, while y_2 and y_4 are free; its completions are $(1, 0, 0, 0)$, $(1, 1, 0, 0)$, $(1, 0, 0, 1)$, and $(1, 1, 0, 1)$

Branch-and-Bound Tree

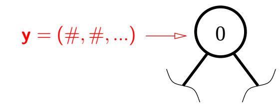


- Nodes of the B&B tree represent partial solution
  - Numbers indicate the sequence in which nodes are investigated
  - Total number of nodes is: $1 + 2 + 2^2 + \cdots + 2^{n_y} = \sum_{i=0}^{n_y} 2^i > 2^{n_y}$!
- Edges of the B&B tree specify how variables are fixed: *branch* part

Getting Started: The Root Node

Root Node

B&B search begins at initial or root partial solution $\mathbf{y}^{(0)} \triangleq (\#, \#, \ldots)$ with all discrete variables free



$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & z \triangleq \mathbf{c}^\mathsf{T}\mathbf{x} + \mathbf{d}^\mathsf{T}\mathbf{y} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{y} \left\{ \begin{matrix} \leq \\ = \\ \geq \end{matrix} \right\} \mathbf{b} \\ & \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \\ & \mathbf{y} \in \{0,1\}^{n_y} \end{aligned}$$

LP $\Longrightarrow$ relaxation

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & z \triangleq \mathbf{c}^\mathsf{T}\mathbf{x} + \mathbf{d}^\mathsf{T}\mathbf{y} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{y} \left\{ \begin{matrix} \leq \\ = \\ \geq \end{matrix} \right\} \mathbf{b} \\ & \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \\ & \mathbf{0} \leq \mathbf{y} \leq \mathbf{1} \end{aligned}$$

Solution of the LP relaxation at the root node provides:

- A lower bound on the MILP (global) optimum for a minimize problem
- An upper bound for a maximize problem

Outcomes from LP Relaxation Solution at the Root Node

1. No feasible solution

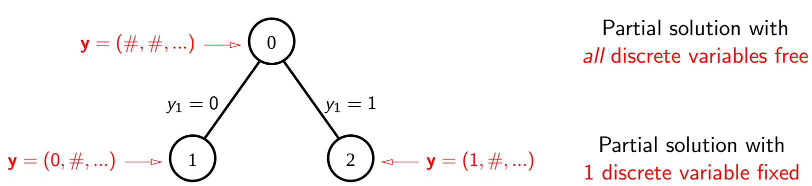
   Action: Terminate by Infeasibility — The MILP problem is itself infeasible

2. All relaxed binary variables are either 0 or 1 at the optimum

   Action: Terminate by Completion — We have found an optimum for the MILP problem (We are *very* lucky!)

3. Some relaxed binary variables have fractional value at the optimum

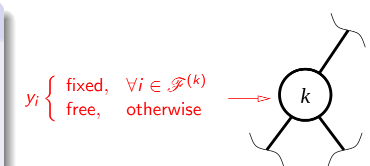
   Action: Branch — Choose one of the relaxed variables, e.g., y_1, and create two new nodes:



Intermediate Nodes

Candidate Problems

The **candidate problem** associated with a partial solution to an MILP model is the restricted model obtained by fixing the discrete variables as in the partial solution



$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & z \triangleq \mathbf{c}^\mathsf{T}\mathbf{x} + \mathbf{d}^\mathsf{T}\mathbf{y} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{y} \left\{ \begin{matrix} \leq \\ = \\ \geq \end{matrix} \right\} \mathbf{b} \\ & \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \\ & \mathbf{y} \in \{0,1\}^{n_y} \end{aligned}$$

Partial solution $\Longrightarrow$ $\mathscr{F}^{(k)} \triangleq$ fixed set

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & z \triangleq \mathbf{c}^\mathsf{T}\mathbf{x} + \mathbf{d}^\mathsf{T}\mathbf{y} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{y} \left\{ \begin{matrix} \leq \\ = \\ \geq \end{matrix} \right\} \mathbf{b} \\ & \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \\ & y_i \left\{ \begin{matrix} \text{fixed}, & \forall i \in \mathscr{F}^{(k)} \\ \in \{0,1\}, & \text{otherwise} \end{matrix} \right. \end{aligned}$$

- Formulate and solve the LP relaxation of a candidate model

Intermediate Nodes (cont'd)

Incumbent Solutions

The **incumbent solution** at any stage in a B&B search is the best (in terms of objective value) feasible solution known so far

- The incumbent solution may have been discovered as the search evolved, or derive from experience prior to the search
- Any incumbent solution provides:
  - an upper bound on the MILP global optimum for a minimize problem
  - a lower bound for a maximize problem

The B&B search is efficient when many partial solution can be terminated at an early stage:

- Exploit LP relaxations of candidate models
- Exploit incumbent solutions

Terminating Partial Solutions

1. The candidate problem has an Infeasible LP relaxation

   Action: Terminate by infeasibility — The candidate problem is itself infeasible

2. The candidate problem has a LP relaxation whose optimal value is no better than the current incumbent solution value

   Action: Terminate by value dominance — No feasible completion of the candidate model can improve on the incumbent

3. The candidate problem has a LP relaxation whose optimal solution with all relaxed binary variables equal to 0 or 1

   Action 1: Terminate by Completion — This is an optimum for the candidate problem
   Action 2: Update incumbent (if applicable)

In any other case, branch!

Terminating Branch-and-Bound Search

- B&B search stops when *every* partial solution in the tree has been either branched or terminated

The final incumbent is a global optimum, if one exists
The model is infeasible, otherwise

- One might also decide the stop B&B search when sufficiently close to the optimum:

$$\frac{z_{\rm rel}^{(k)} - z_{\rm inc}^{(k)}}{\frac{1}{2}\left|z_{\rm rel}^{(k)} + z_{\rm inc}^{(k)}\right|} < \epsilon_r$$

  with ϵ_r a user tolerance

Branch-and-Bound Heuristics

Heuristics for Branching Variable Selection:

- Consider only those discrete variables having fractional values in the associated candidate problem
- If several, branch by fixing the fractional discrete variable closest to 0 or 1 — Accounting on experience can be pretty useful too!

Heuristics for Branching Node Selection:

- Depth-first search selects an active partial solution with the most component fixed — i.e., one deepest in the search tree
- Best-first search selects an active partial solution with best parent bounds
- Depth-forward best-back search selects a deepest active partial solution after branching a node, but one with best parent bound after a termination

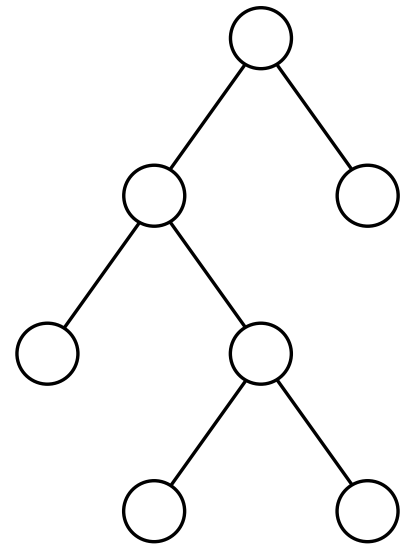
Applying Branch-and-Bound Search

Class Exercise: The following table shows candidate problem LP relaxation optima for all possible combinations of fixed and free values in a maximizing MILP problem over $x \geq 0$ and $(y_1, y_2, y_3) \in \{0, 1\}$

Partial	$\mathbf{y}^{(k)}_{\rm rel}$	$x^{(k)}_{\rm rel}$	$z^{(k)}_{\rm rel}$	Partial	$\mathbf{y}^{(k)}_{\rm rel}$	$x^{(k)}_{\rm rel}$	$z^{(k)}_{\rm rel}$
(#,#,#)	(0.2,1,0)	0	82.80	(0,0,1)	Infeasible		
(#,#,0)	(0.2,1,0)	0	82.80	(0,1,#)	(0,1,0.67)	0	80.67
(#,#,1)	(0,0.8,1)	0	79.40	(0,1,0)	(0,1,0)	2	28.00
(#,0,#)	(0.7,0,0)	0	81.80	(0,1,1)	(0,1,1)	0.5	77.00
(#,0,0)	(0.7,0,0)	0	81.80	(1,#,#)	(1,0,0)	0	74.00
(#,0,1)	(0.4,0,1)	0	78.60	(1,#,0)	(1,0,0)	0	74.00
(#,1,#)	(0.2,1,0)	0	82.80	(1,#,1)	(1,0,1)	0	63.00
(#,1,0)	(0.2,1,0)	0	82.80	(1,0,#)	(1,0,0)	0	74.00
(#,1,1)	(0,1,1)	0.5	77.00	(1,0,0)	(1,0,0)	0	74.00
(0,#,#)	(0,1,0.67)	0	80.67	(1,0,1)	(1,0,1)	0	63.00
(0,#,0)	(0,1,0)	2	28.00	(1,1,#)	(1,1,0)	0	62.00
(0,#,1)	(0,0.8,1)	0	79.40	(1,1,0)	(1,1,0)	0	62.00
(0,0,#)	Infeasible			(1,1,1)	(1,1,1)	0	51.00
(0,0,0)	Infeasible						

1. Solve the model by B&B search, using depth-first search

Applying Branch-and-Bound Search



Mixed-Integer Programming — The Final Words

- Mixed-integer programs are very common!
- In general, MILPs require the solving of many (perhaps a huge number) of LPs
- B&B search offers the potential for a sizeable reduction in computation — Though not for *all* MILP problems!
- Experience with a problem can lead to great computational savings
- Sensitivity information is lacking at an optimal solution due to binary/integer variables
- GAMS™ with the CPLEX solver provides good performance for MILP solution