The CPLEX Library: Mixed Integer Programming

Ed Rothberg, ILOG, Inc.

The Diet Problem Revisited

Nutritional values

- **Bob considered the following foods:**

Food	Serving Size	Energy (kcal)	Protein (g)	Calcium (mg)	Price per serving	
Oatmeal	28 g	110	4	2	$0.30	x_1
Chicken	100 g	205	32	12	$2.40	x_2
Eggs	2 large	160	13	54	$1.30	x_3
Whole milk	237 cc	160	8	285	$0.90	x_4
Cherry pie	170 g	420	4	22	$2.00	x_5
Pork and beans	260 g	260	14	80	$1.90	x_6

- **Bob examines optimal solution and decides he needs more protein**
  - **Adds a protein constraint**
  - **Possible outcome:**
    - Eggs: $x_3 = 0.6203$ in optimal solution

Mixed Integer Programming (MIP)

$$\text{Minimize} \quad c^T x$$

$$\text{Subject to} \quad Ax = b$$

$$l \leq x \leq u$$

$$\text{Some } x_j \text{ are integer}$$

Integrality Restriction

3

The "Algorithm" for Solving a MIP

- Base algorithm: branch-and-bound
  - (Land and Doig 1960)
  - Linear programming as a subroutine
  - Provably exponential
- A "bag of tricks" to accelerate the search
  - Most tricks apply to only a subset of models
  - A "barrage" of algorithms

4

Branch and Bound

Solution Strategy: Branch & Bound

Key ingredients

- Split the solution space into disjoint subspaces
- Bound the objective value for all solutions in a subspace

Branching

Branch on integer variable

- Choose a *branching variable* x_j
  - Must be an integer variable
- Split the model into two sub-models
  - $x_j \leq i$ or $x_j \geq i+1$
- Binary variable special case:
  - $x_j=0$ or $x_j=1$

7

Bounding - Continuous Relaxation

Minimize $c^T x \ (= z_{lp})$

Subject to $Ax = b$

$l \leq x \leq u$

Some x are integer

Lower bound on MIP objective

Relax Integrality Restriction

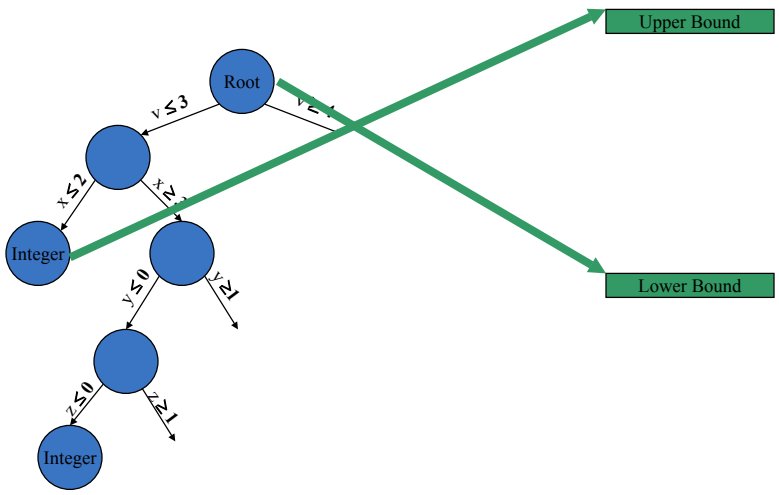
8

Nice Properties of Continuous Rel.

- If relaxation solution satisfies integrality restrictions:
  - No need to further explore subspace
- Natural branching candidates:
  - Integer variables that are fractional in relaxation

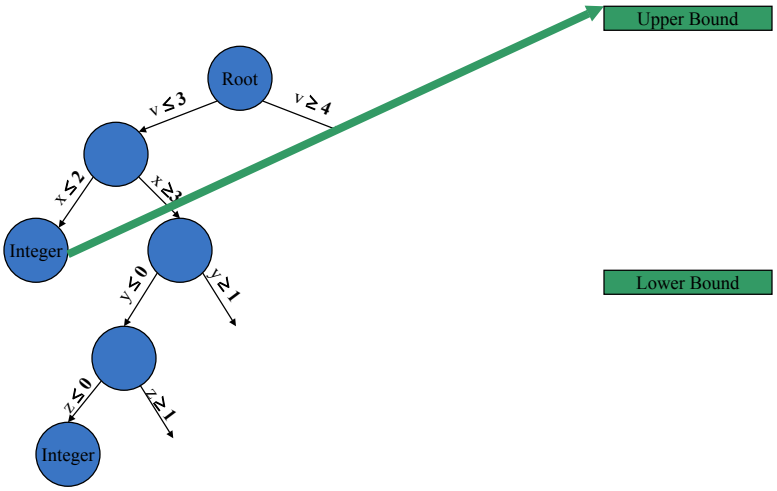
9

Branch and Bound for MIP



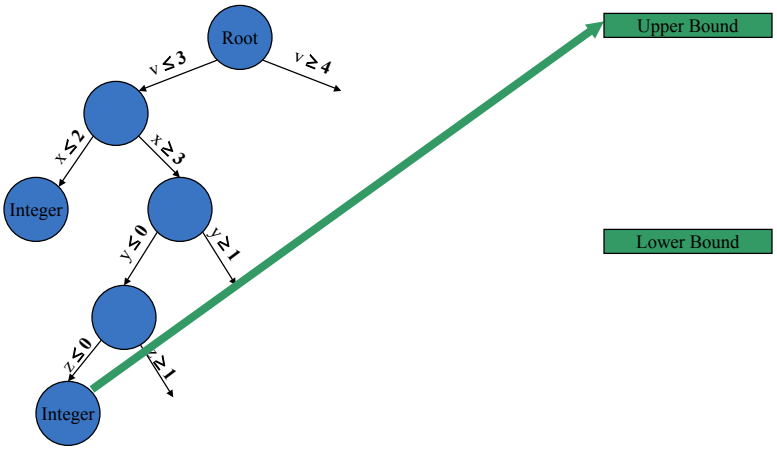
10

Branch and Bound for MIP



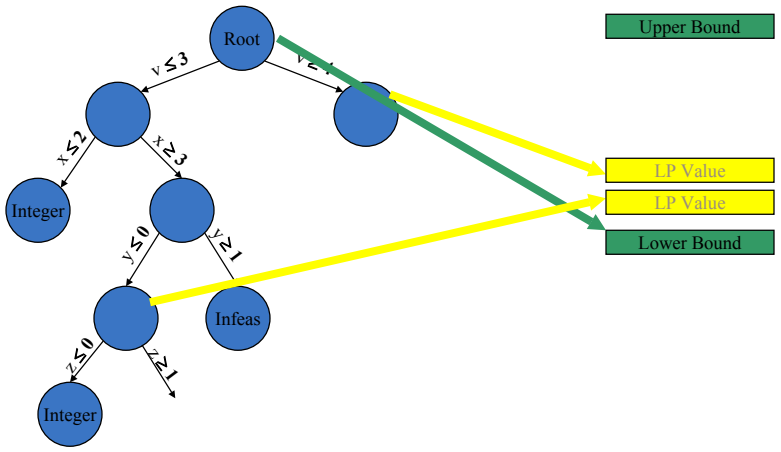
11

Branch and Bound for MIP



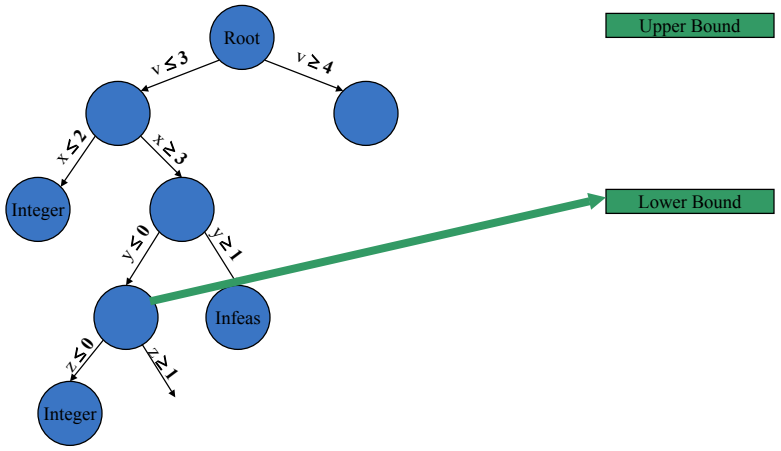
12

Branch and Bound for MIP



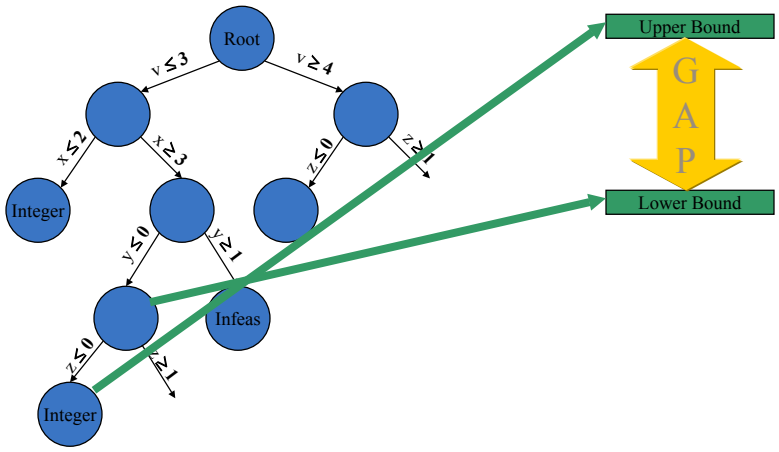
13

Branch and Bound for MIP



14

Branch and Bound for MIP



15

Important Steps

16

Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- Solve continuous relaxation
- Generate *cutting planes*
- Perform variable fixing
- Find integer feasible solutions that are "similar" to the relaxation solution
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

17

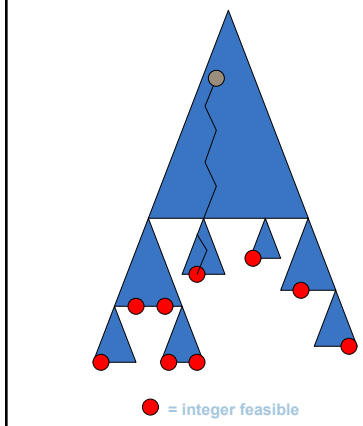
Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- Generate *cutting planes*
- Perform variable fixing
- Find integer feasible solutions that are "similar" to the relaxation solution
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

18

Node Selection



Tradeoff: feasibility versus optimality

- When exploring nodes deep in the search tree…
  - More likely to find integer feasible solutions
  - More likely to explore nodes that would be pruned by later feasible solutions

19

Node Selection Options

Options

- Depth first
- Breadth first
- Best first
- Limited discrepancy
- Best estimate

- Plunging (combined with above)
  - Always choose a child of the previously explored node

20

Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- **Solve relaxation**
- Generate *cutting planes*
- Perform variable fixing
- Find integer feasible solutions that are "similar" to the relaxation solution
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

21

Node Relaxation Solution

Ideally suited to dual simplex

- **Change from parent relaxation is small : a new bound on the branching variable**
  - **Previous basis remains dual feasible**
  - **Solution likely to be "close" to previous basis**
- **A few iterations of dual simplex typically suffice to restore optimality**
- **Cost per node quite low**

22

Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- **Generate *cutting planes***
- Perform variable fixing
- Find integer feasible solutions that are "similar" to the relaxation solution
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

23

Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- Generate *cutting planes*
- **Perform variable fixing**
- Find integer feasible solutions that are "similar" to the relaxation solution
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

24

Reduced Cost Fixing

Use reduced costs to fix variables

- Recall: reduced cost D_N is the marginal cost of moving a variable off of its bound
- If $z_{lp} + |D_j| \geq z^*$
  - z^* = objective of best known feasible solution (incumbent)
- Then x_j can be fixed to its current value in this subtree

25

Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- Generate *cutting planes*
- Perform variable fixing
- **Find integer feasible solutions that are "similar" to the relaxation solution**
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

26

Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- Generate *cutting planes*
- Perform variable fixing
- Find integer feasible solutions that are "similar" to the relaxation solution
- **Choose a variable on which to branch**
- Explore logical implications of branch
- Repeat

27

Variable Selection

Greatly affects search tree size

- **Guiding principles:**
  - **Make important decisions early**
  - **Both directions of branch should have an impact**
- **Example:**
  - **Decide whether or not to build a factory first**
  - **Decide how many lines to place in the factory later**

28

Variable Selection

Predicting impact

- **Question:**
  - **How to predict impact of a branch?**
- **Possible answers:**
  - **Find variables that are furthest from their bounds**
    - Maximum infeasibility
  - **Measure the impact for each branching candidate**
    - Strong branching [Applegate, Bixby, Chvatal, Cook]
  - **Use historical information**
    - Pseudo-costs

29

Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- Generate *cutting planes*
- Perform variable fixing
- Is the relaxation solution near-feasible?
- Choose a variable on which to branch
- **Explore logical implications of branch**
- Repeat

30

Logical Propagation

Propagate implications logically

- Simple example:
  - x + 2y + 3z ≤ 3, all variables binary
  - x = 1 (e.g., fixed during tree exploration)
  - z = 2/3 still feasible in LP relaxation
- Use *bound strengthening* to tighten variable bounds

31

Later Today

- Brief History of CPLEX MIP
- Heuristic details
- Cutting plane details

32