The CPLEX Library: MIP Heuristics

Ed Rothberg, ILOG, Inc.

Motivation for Heuristics

Why not wait for branching?

- Produce feasible solutions as quickly as possible
  - Often satisfies user demands
  - Avoid exploring unproductive subtrees
  - Better reduced-cost fixing
- Avoid “tree pollution”
  - Good fixings in a heuristic are often not good branches
- Increase diversity of search
  - Strategies in heuristic may differ from strategies in branching

CPLEX Heuristics

Two classes

- Plunging heuristics:
  - Maintain linear feasibility
  - Try to achieve integer feasibility
- Local improvement heuristics:
  - Maintain integer feasibility
  - Try to achieve linear feasibility

3

Plunging Heuristic Structure

- Fix a set of integer infeasible variables
  - Usually by rounding
- Perform bound strengthening to propagate implications
- Solve LP relaxation
- Repeat

4

Bound Strengthening

Changing the rules of business™

Propagate new bounds through inequalities

- Given a constraint:
  - $\sum a_j x_j \leq b$
  - Split equalities into a pair of inequalities
- Consider a single x_k:
  - $a_k x_k + \inf (\sum_{j!=k} a_j x_j) \leq \sum a_j x_j \leq b$
  - $x_k \leq (b - \inf (\sum_{j!=k} a_j x_j)) / a_k$
    - Assuming $a_k \geq 0$
- Change in variable bound can produce changes in other bounds

5

Bound Strengthening Example

Changing the rules of business™

- $x + 2y + 3z \leq 3$
  - all variables binary
  - x=1
- $3 z \leq 3 - \inf (x + 2y) = 3 - 1 = 2$
- $z \leq 2/3$

6

Plunging Details

Changing the rules of business™

Important details

- How many variables to fix per round:
  - **All of them?**
    - Inexpensive; no need to solve LP relaxations
    - But ‘flying blind’ after a few fixings
      - Bound strengthening helps
  - **A few?**
    - More expensive
    - LP relaxation can guide later choices
      - (variable values, reduced costs, etc.)
- In what order are variables fixed?
  - **Variations useful for diversification**

7

Local Improvement Heuristics

Changing the rules of business™

High-level structure

- Choose integer values for all integer variables
  - **Produces linear infeasibility**
- Iterate over integer variables:
  - **Does adding/subtracting 1 reduce linear infeasibility?**
- Infeasibility metrics:
  - **Primary: number of violated constraints**
  - **Secondary: |b-Ax|**

8

Local Improvement Details

- What initial values to assign to integer variables?
  - Rounded relaxation values
  - 0
- Move acceptance criteria?
  - Greedy
- What to do when local improvement gets stuck?
  - Reverse infeasibility metrics

9

Local Improvement Details

Continuous variables

- What to do about continuous variables?
  - To what value should they be fixed?
  - What does the neighborhood look like?
- Our approach:
  - Don’t fix them
  - Constraint is satisfied if inf(LHS) <= RHS

10

General Heuristic Strategies

Apply 9 different variations

- Apply the least expensive heuristics after every round of root cutting planes
- Apply all heuristics before beginning the branch and bound search
- Apply them every 10 nodes in the MIP tree
- Decrease the frequency of a particular heuristic when it is not finding new feasible solutions

Sample CPLEX Output

First 1,000 nodes, default settings

	Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Node	ItCnt	Gap
	0	0	346.0000	536		346.0000	551	
*	0+	0		0	6262.0000	346.0000	551	94.47%
			560.0000	490	6262.0000	Cuts: 700	1131	91.06%
			560.0000	592	6262.0000	Cuts: 688	1561	91.06%
*	260+	256		0	3780.0000	560.0000	3566	85.19%
*	300+	296		0	2992.0000	560.0000	3703	81.28%
*	300+	296		0	2626.0000	560.0000	3703	78.67%
*	393	368		0	2590.0000	560.0000	4405	78.38%
*	680+	628		0	2576.0000	560.0000	6928	78.26%
*	690+	638		0	2538.0000	560.0000	6946	77.94%
*	710+	656		0	2478.0000	560.0000	7011	77.40%
*	720+	658		0	2448.0000	560.0000	7027	77.12%
*	720+	645		0	2402.0000	560.0000	7027	76.69%
*	730+	639		0	2360.0000	560.0000	7070	76.27%
*	820+	726		0	2340.0000	560.0000	8134	76.07%

…

Heuristic Results

Effectiveness

- Feasible solution found for most models before branch and bound begins
- Roughly 10% improvement in time to proven optimality (978 model test set)
- Often find solutions branching does not

13

Combining Local Search and MIP Heuristics to Solve Very Difficult MIP Models

14

Local Search

ILOG — Changing the rules of business™

Powerful optimization framework

- Local search is a very powerful heuristic approach to solving difficult combinatorial optimization problems
- Example local search methods:
  - Simulated annealing
  - Tabu search
  - Genetic algorithms
  - …

15

Local Search

ILOG — Changing the rules of business™

Three key ingredients

- *Neighborhood*:
  - A set of solutions that are in the vicinity of the current solution
- *Intensification*:
  - A temporary focus on a part of the solution space
- *Diversification*:
  - A mechanism for changing focus on occasion

16

Applying Local Search to MIP?

- Neighborhoods:
  - **Local search neighborhoods generally based on problem structure**
    - Example: Nodes and edges in a graph
  - **No high level structural information available in an arbitrary MIP model**
- Given an incumbent x*, can we generate and explore an interesting neighborhood?

17

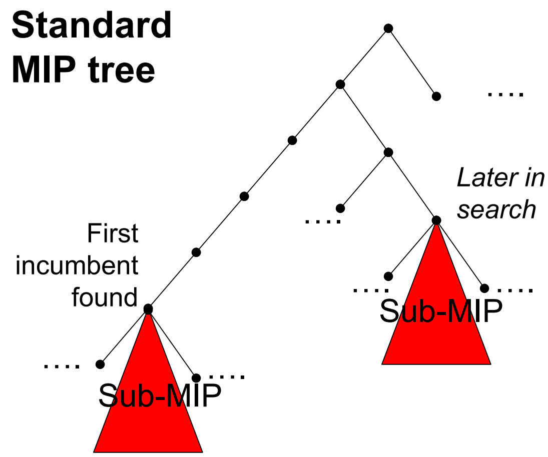
Two Recent Proposals

MIP neighborhood notions

- Local Branching [Fischetti and Lodi, 2002]
  - **Add a local branching constraint to MIP model:**
    - |x - x*| <= k
  - **Solve a (truncated) sub-MIP**
- Relaxation Induced Neighborhood Search (RINS) [Danna, Rothberg, and Le Pape, 2003]
  - **Fix all variables that agree in the current relaxation solution and x***
  - **Solve a sub-MIP on the variables that differ**

18

Intensification through sub-MIPs



19

Local Branching Details

Explore vicinity of incumbent

- Constrain sub-MIP to explore a small neighborhood of incumbent x*
  - |x - x*| <= k
  - k chosen to be ~20
- Apply whenever a new incumbent is found
  - Including those found by local branching
- A succession of improving, neighboring solutions

20

RINS Details

Explore portion where solutions differ

- Combine desirable properties of two solutions:
  - Incumbent: feasible
  - Relaxation: optimal
- Neighborhood contains both solutions
- Extend promising partial solution

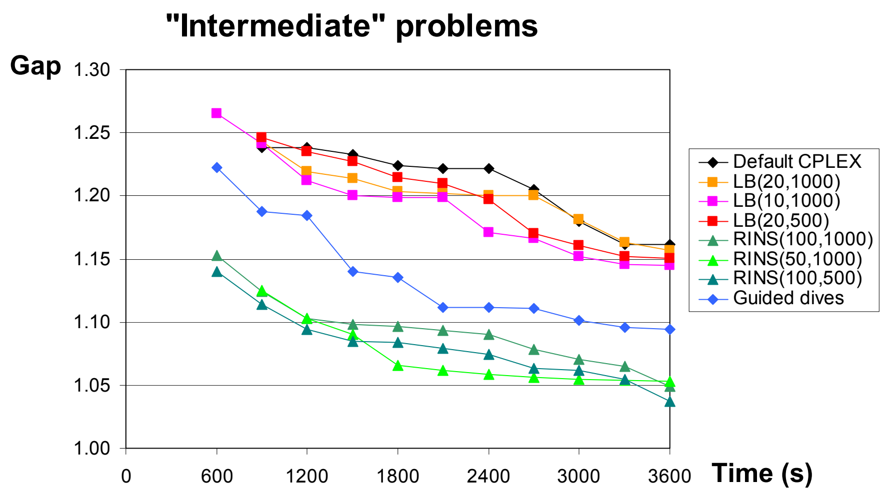
21

Local branching *vs.* RINS

Local branching	RINS
Explores a neighborhood of the incumbent	Explores a neighborhood of both the incumbent and the continuous relaxation
Can be called only each time a new incumbent is found	Can be called at each node of the branch-and-cut tree
Expensive sub-MIP: original model + dense constraint	Sub-MIP on a reduced number of variables
Not efficient on general integer variables	Can handle any type of variable

22

Results



23

Extensions

- **Other interesting neighborhoods?**
- **More efficient ways to explore neighborhoods?**

24