MIP Heuristics

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**

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**

Plunging Heuristic Structure

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

Plunging Fixings

- **Methods for choosing fixings:**
  - **Non-basic variables**
    - Sorted in order of increasing dj
  - **Fractional variables**
    - Sorted in order of increasing distance to an integer in relaxation

Plunging Details

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**

Local Improvement Heuristics

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|

Local Improvement Details

- What initial values to assign to integer variables?
  - Rounded relaxation values
  - Bounds
- What to do when local improvement gets stuck?
  - Reverse infeasibility metrics

General Heuristic Strategies

Apply 11 different variations

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

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 finds solutions branching does not

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

Local Search for MIP

Relaxation Induced Neighborhood Search (RINS)

- **New local search heuristic in CPLEX**
- **Local search uses *neighborhoods* to improve a given solution**
  - **Neighborhoods generally based on problem structure**
    - Example: Nodes and edges in a graph
  - **No high level structural information available in an arbitrary MIP model**

Local Search for MIP

Basic Approach

- Form *sub-MIP* from variables whose values differ in incumbent and relaxation
- Combine desirable properties of two solutions:
  - Incumbent: feasible
  - Relaxation: optimal
- Neighborhood contains both solutions
- Extend promising partial solution

Local Search for MIP

Example: RINS = off

```
   Nodes                                                 Cuts/
   Node  Left     Objective  IInf  Best Integer     Best Node    ItCnt     Gap
      0     0    6.5256e+08   191                  6.5256e+08     1425
*     0+    0                   0    6.8841e+08    6.5256e+08     1425    5.21%
                 6.5343e+08   208    6.8841e+08    Cuts:  118     1716    5.08%
                 6.5380e+08   184    6.8841e+08     Cuts:  87     2042    5.03%
                 6.5381e+08   185    6.8841e+08   GUBcuts:  3     2083    5.03%
*  2320+ 2317                  0    6.7979e+08    6.5383e+08    28773    3.82%
   4000  3961    6.7069e+08    22    6.7979e+08    6.5385e+08    48444    3.82%
Elapsed time = 478.14 sec. (tree size = 20.75 MB)
```

Local Search for MIP

Example: RINS = 100

```
   Nodes                                                Cuts/
   Node  Left     Objective  IInf  Best Integer     Best Node    ItCnt     Gap

      0     0    6.5256e+08   191                 6.5256e+08     1425
*     0+    0                   0    6.8841e+08    6.5256e+08     1425    5.21%
                 6.5343e+08   208    6.8841e+08    Cuts:  118     1716    5.08%
                 6.5380e+08   184    6.8841e+08     Cuts:  87     2042    5.03%
                 6.5381e+08   185    6.8841e+08   GUBcuts:  3     2083    5.03%
*    100+  100                 0    6.7642e+08    6.5383e+08     5059    3.34%
*    400+  400                 0    6.7190e+08    6.5383e+08    10388    2.69%
*    500+  500                 0    6.7153e+08    6.5383e+08    11797    2.64%
*    600+  600                 0    6.7138e+08    6.5383e+08    13408    2.61%
*   1200+ 1188                 0    6.7137e+08    6.5385e+08    23258    2.61%
*   1300+ 1286                 0    6.7046e+08    6.5385e+08    25503    2.48%
*   2100+ 1936                 0    6.6916e+08    6.5392e+08    40180    2.28%
    3000  2817    6.6233e+08   135    6.6916e+08    6.5392e+08    61274    2.28%
Elapsed time = 456.37 sec. (tree size = 14.94 MB)
```