Branching, Fixing, and Upper Bounds Heuristics

Course: Recent Research Results

Simon Spoorendonk
DIKU, University of Copenhagen

November 28th, 2006

Overview

- Branching and Fixing
- Upper bound heuristics
- Test results

Branching and Fixing

- Tree traversal and Diving
- Strong Branching
- Reduced Cost Fixing

Tree Traversal and Diving

Branch node evaluation:

- Eager, solve branch nodes before search direction is chosen
- Lazy, chose search direction then solve branch node

Tree search strategies:

- Best bound search, eager branching
- Breath search, lazy branching
- Depth search

Diving:

- Combining depth first search with any other
- Explore a (part of) sub-tree by diving
- Used to find feasible solutions fast

Strong Branching

See Achterberg et al. [2]

Combines eager and lazy branch node evaluation:

- Choose a branch node candidate set
- Solve branch nodes given an LP iteration limit
- Choose search direction

Pros:

- Avoids getting stuck in hard solvable nodes (for the time being)
- Infeasible nodes are often detected
- More candidates can be processed than with eager branching

Cons:

- Stores many half solved LPs, or recalculates LP when processing branch node

Reduced Cost Fixing

See Wolsey [6]

Fix binary variables based on reduced cost

Consider MIP:

$$\min\{cx : Ax \leq b, x \in \mathrm{B}\}$$

and reduced cost $\bar{c}$

Fix non basic variables with $\bar{c} > 0$ to 0 iff:

$$LB + \bar{c}_j \geq UB, \quad \forall j \in B : x_j = 0$$

Note:

- Only valid for true LB, i.e., take care in Branch-and-Price algorithms
- Can be generalized to integer variables at their lower or upper bound

Upper bound heuristics

Why do we need good upper bounds?

- Prune nodes in Branch-And-Bound tree
- Reduced cost fixing of variables in LP

Approaches

- Rounding
- Feasibility Pump (FP)
- Objective Feasibility Pump (OFP)
- Local Branching (LB)
- Relaxation Induced Neighborhood Search (RINS)
- Guided Dives (GD)

Rounding

Simple approach

- Round fractional variables to nearest integer
- Check feasibility

Pros:

- Fast to compute
- Simple to implement

Cons:

- High probability for infeasibility

Feasibility Pump (FP) – 1

See Bertacco et al. [3]

Outline:

- Solve LP relaxation of MIP

$$\min\{cx : Ax \leq b, l \leq x \leq u\}$$

- Round solution to $\tilde{x}$, stop if feasible
- Change objective function of LP to minimize on variable distance from integrality.

$$\Delta(x, \tilde{x}) = \sum_{j \in I} |x_j - \tilde{x}_j|$$

- Minimize on integer variables by adding artificial variables d to measure distance from upper bound u and lower bound l
- Solve sub-MIP

$$\begin{aligned} \min\ & \Delta(x, \tilde{x}) = \sum_{j \in I : \tilde{x}_j = l_j} (x_j - l_j) + \sum_{j \in I : \tilde{x}_j = u_j} (u_j - x_j) + \sum_{j \in I : l_j < \tilde{x}_j < u_j} d_j \\ \text{s.t.}\ & Ax \leq b \\ & d \geq x - \tilde{x} \\ & d \geq \tilde{x} - x \\ & l \leq x \leq u \end{aligned}$$

Feasibility Pump (FP) – 2

Notes:

- Can cycle between solutions, handled by restarting with random perturbed solution
- A three step procedure:
  1. *Pumping rounds* on binary variables only
  2. Include integer variables
  3. Process MIP

$$\min\{\Delta(x, \tilde{x}) : Ax \leq b, l \leq x \leq u, x_j \in \mathbb{Z} \forall j \in I\}$$

Pros:

- Does not need an incumbent

Cons:

- Can be time consuming to solve sub-MIP
- Quality of integral solutions can be poor

Objective Feasibility Pump (OFP)

See Achterberg and Berthold [1]

Extends previous method:

- Gradually reduces influence of original objective function
- Replace $\Delta(x, \tilde{x})$ with

$$\Delta_\alpha(x, \tilde{x}) = (1-\alpha)\Delta(x, \tilde{x}) + \alpha \frac{||\Delta||}{||c||} cx, \quad 0 \leq \alpha \leq 1$$

- Depending on step: $||\Delta|| = \sqrt{|B|}$ or $||\Delta|| = \sqrt{|I|}$
- α decreased with a fixed factor in each iteration

Same notes pros/cons as before plus:

Pros:

- Generally finds higher quality solutions than normal Feasibility Pump

Local Branching (LB) – 1

See Fischetti and Lodi [5]

Solves a sub-MIP problem

- Improves on the current incumbent
- Considers Hamming distance from a subset of the binary variable in the incumbent

$$H(x, x^*) = \sum_{j \in B} |x_j - x_j^*|$$

- Add constraint

$$\sum_{j \in B \cap \{x_j^* = 1\}} (1 - x_j) + \sum_{j \in B \cap \{x_j^* = 0\}} x_j \leq r$$

  with radius r and optimize sub-MIP
- Can be extended to include integer variables
- Adding complementary constraint

$$H(x, x^*) \geq r + 1$$

  to LP avoids reexploitation of B

Local Branching (LB) – 2

Pros:

- Despite the name "Local Branching", cuts

$$H(x, x^*) \geq r + 1$$

  are globally valid

Cons:

- Adding complementary constraints makes LP dense
- Needs incumbent to improve, i.e., only usable when new IP solution is found in Branch-and-Bound tree
- Can be time consuming to solve sub-MIP

Relaxation Induced Neighborhood Search (RINS) – 1

See Danna et al. [4]

At a node of the Branch-and-Bound tree:

1. Fix variables that have same value as in incumbent
2. Set an objective cut-off based in incumbent value
3. Solve a sub-MIP

Notes:

- Sub-MIP can be large. Only explore part of tree
- RINS is run for each $f >> 1$ branch nodes
- Cutting planes from original LP can be exploited
- RINS can be regarded as a Large Neighborhood Search method

Relaxation Induced Neighborhood Search (RINS) – 2

Pros:

- Loose relaxations are guided by incumbent
- Poor quality incumbents are guided by LP relaxation

Cons:

- Can be time consuming to solve sub-MIPs
- Same RINS sub-MIPs can be constructed in different branch nodes
- Sub-MIP can be identical to part of branch tree
- An incumbent solution is needed

Guided Dives

See Danna et al. [4]

Does not solve sub-MIP problems.

- Modified tree traversal strategy
- Choose variable to branch on
- Choose which child node to explore - based on value of branching variable in incumbent

Pros:

- Easy to implement
- Surprisingly good results

Cons:

- Calculation of child node candidates before branching
- An incumbent solution is needed

Test Results – 1

FP and OFP – MIBLIB 3.0

	Original Feasibility Pump						Objective Feasibility Pump					
Name	Objective	Gap %	Time	Iter	Rst	St	Objective	Gap %	Time	Iter	Rst	St
10teams	958	3.7	1	6	0	1	952	3.0	5	79	16	1
a1c1s1	17762	53.2	1	8	0	1	16076.6	38.6	1	29	0	1
aflow30a	2549	120.1	0	7	1	1	4105	254.5	0	40	0	1
aflow40b	7682	557.7	0	22	3	1	2049	75.4	0	15	0	1
air04	58608	4.4	15	4	0	1	57298	2.1	164	51	9	1
air05	30883	17.1	3	3	0	1	26942	2.2	8	7	0	1
arki001	7.75064e+06	2.2	12	797	67	3	7.70474e+06	1.6	10	737	62	3
atlanta-ip	166.014	74.7	45	128	23	2	138.012	45.3	56	65	5	1
cap6000	-2.37503e+06	3.1	0	2	0	1	-2.42701e+06	1.0	0	30	0	1
dano3mip	1000	43.3	30	1	0	1	769.25	10.2	383	70	1	1
danoint	93	41.6	1	174	197	3	87	32.5	3	245	172	2
disctom	-5000	0.0	9	6	0	1	-5000	0.0	11	22	0	1
ds	–	–	>3600	265	4	2	–	–	>3600	254	6	2
fast0507	245	40.8	20	1	0	1	179	2.9	21	8	0	1
fiber	4.01694e+06	889.6	0	6	2	1	1.20751e+06	197.5	0	29	0	1
fixnet6	9283	133.1	0	3	1	1	4807	20.7	0	29	0	1
gesa2-o	4.91411e+07	90.6	0	15	3	2	2.6504e+07	2.8	0	29	0	2
gesa2	2.82478e+07	9.6	1	9	1	2	2.67652e+07	3.8	1	29	0	2
glass4	5.20005e+09	333.3	0	159	96	2	3.10003e+09	158.3	0	183	45	2
harp2	-6.06939e+07	17.9	0	43	13	1	-5.58762e+07	24.4	0	368	67	2
liu	6378	444.2	0	1	0	1	4100	249.8	1	168	17	1
manna81	-12891	2.1	0	6	0	2	-12894	2.1	0	8	0	2
markshare1	362	36100.0	0	3	0	1	194	19300.0	1	66	4	1
markshare2	1523	152200.0	0	2	0	1	365	36400.0	0	69	1	1
mas74	18692.3	58.4	0	3	0	2	19033.1	61.3	0	106	12	1
mas76	72860.6	82.1	0	2	0	1	50124	25.3	0	107	15	1
misc07	4100	45.9	1	37	6	1	3425	21.9	0	129	24	1
mkc	-288.01	48.9	0	3	0	1	-289.95	48.6	0	13	0	1
mod011	-2.38751e+07	56.2	0	1	0	1	-4.56201e+07	16.4	1	12	0	1
modglob	3.08143e+07	48.6	0	1	0	1	2.10876e+07	1.7	0	63	24	1
momentum1	359238	3.7	818	416	186	3	346535	0.0	223	351	118	3
momentum2	–	–	>3600	504	139	3	–	–	>3600	522	158	3
momentum3	509585	37.7	272	148	26	1	420724	13.7	599	173	13	1
msc98-ip	3.02737e+07	30.1	34	19	0	1	3.02655e+07	30.1	38	81	10	1
mzzv11	-11286	48.0	118	688	114	3	-17688	18.6	112	699	75	3
mzzv42z	-12472	39.3	22	21	6	1	-15470	24.7	78	696	117	3
net12	337	57.5	8	103	52	1	337	57.5	14	257	66	2
noswot	-26	36.6	0	2	0	1	-40	2.4	0	30	0	2
nsrand-ipx	78240	52.8	0	2	0	1	89120	74.1	1	10	0	1
nw04	19124	13.4	3	2	0	1	17856	5.9	9	33	0	1
opt1217	-16	0.0	0	3	0	1	-16	0.0	0	43	1	1
p2756	91972	2844.0	2	799	32	3	89266	2757.4	3	772	41	3
pk1	78	609.1	0	1	0	1	83	654.5	0	57	3	1
pp08a	12180	65.7	1	4	0	1	10940	48.8	0	11	0	1
pp08aCUTS	10750	46.3	0	2	0	1	8530	16.1	0	9	0	1
protfold	-10	66.7	683	620	128	3	-12	60.0	268	713	106	3
qiu	1945.5	1564.2	0	1	0	1	625.709	570.9	0	9	0	1
rd-rplusc-21	173065	1.1	375	637	120	3	171182	0.0	790	525	109	3
roll3000	18812	45.5	0	68	17	1	24417.6	88.9	6	536	118	3
rout	1720.82	59.7	0	96	35	2	1773.95	64.6	0	86	6	2
set1ch	72987.8	33.8	1	4	0	1	84167.5	54.3	0	29	0	1
seymour	527	24.6	1	1	0	1	445	5.2	3	8	0	1
sp97ar	9.57074e+08	44.0	3	1	0	1	9.40566e+08	41.5	3	9	0	1
stp3d	–	–	>3600	26	2	1	–	–	>3600	17	0	1
swath	1630.8	192.4	3	107	13	1	1280.95	129.6	13	238	39	2
t1717	237564	22.9	556	56	10	1	195779	1.3	171	30	0	1
timtab1	1.51227e+06	97.7	1	174	23	2	1.33858e+06	75.0	1	334	30	2
timtab2	1.91798e+06	57.7	1	477	38	2	1.73262e+06	42.5	4	1557	116	3
tr12-30	269910	106.7	0	9	0	1	163794	25.4	0	24	0	1
vpm2	19.5	41.8	0	2	0	1	15.25	10.9	0	10	0	1

Table 1. Comparison of original feasibility pump and objective sensitive version

Test Results – 2

RINS, LB, GD – includes job-shop scheduling, network flows

Best solution ratio:

Table 2. Ratio solution reached/best known solution

Instance	Default CPLEX	RINS	Local branching	Guided dives	LB+RINS 1	LB+RINS 2
'Small spread' problems - one hour						
A1C1S1	1.039	1.006	1.011	1.016	**1.000**	1.002
A2C1S1	1.038	1.007	**1.000**	1.035	1.006	**1.000**
arki001	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
B2C1S1	1.070	1.041	1.079	1.047	**1.010**	1.024
biella1	1.004	**1.000**	1.001	**1.000**	1.001	**1.000**
nsrand_ipx	1.006	**1.000**	1.003	**1.000**	**1.000**	**1.000**
rail2586c	1.015	1.016	1.014	1.018	**1.007**	1.019
rail4284c	1.007	1.009	1.007	1.005	**1.002**	1.009
rail4872c	1.010	1.008	1.008	1.014	**1.000**	1.006
rococoB10-011000	1.041	1.022	1.034	1.022	**1.008**	1.044
rococoB10-011001	1.041	1.033	**1.025**	1.083	1.101	1.079
rococoB11-010000	**1.042**	1.061	1.118	**1.042**	1.119	1.093
rococoC10-001000	1.001	1.001	1.001	1.001	**1.000**	**1.000**
roll3000	1.014	1.005	1.003	1.016	**1.000**	1.001
seymour	1.012	1.002	1.009	1.005	1.002	1.002
sp97ar	1.012	1.001	1.012	1.006	1.005	**1.000**
sp97ic	1.026	1.002	1.012	1.013	**1.000**	1.006
sp98ar	1.007	**1.002**	**1.002**	1.003	**1.002**	1.003
sp98ic	1.015	1.003	1.004	1.004	**1.002**	1.005
tr12-30	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
UMTS	1.001	**1.000**	1.001	1.001	**1.000**	1.001
'Medium spread' problems - one hour						
B1C1S1	1.118	**1.011**	1.020	1.096	1.018	1.013
glass4	1.123	1.096	1.130	1.113	**1.000**	1.119
ljb2	1.225	**1.000**	1.116	1.038	**1.000**	**1.000**
net12	1.192	**1.000**	1.192	**1.000**	**1.000**	**1.000**
rococoB11-110001	1.205	**1.121**	1.254	1.224	1.216	1.148
rococoB12-111111	1.141	**1.029**	1.057	1.099	1.084	1.062
rococoC10-100001	1.180	**1.095**	1.214	1.132	1.137	1.294
rococoC11-010100	1.173	1.081	1.081	1.123	**1.053**	1.060
rococoC11-011100	1.312	**1.055**	1.330	1.118	**1.055**	1.452
rococoC12-100000	1.270	**1.096**	1.110	1.268	1.117	1.278
rococoC12-111100	1.148	**1.025**	1.125	1.112	1.070	1.081
swath	1.222	1.048	1.154	1.093	**1.031**	1.048
'Large spread' problems - two hours						
ljb7	2.375	1.061	1.580	1.582	**1.028**	1.255
ljb9	1.858	**1.581**	1.995	1.718	1.646	1.809
ljb10	1.601	**1.212**	1.693	1.295	1.226	1.500
ljb12	2.568	**1.512**	3.083	2.012	2.097	2.418

- LB+RINS1 is LB used to improve RINS solution
- LB+RINS2 is LB used in RINS neighborhood

Test Results – 3

Time to improved solution:

Table 5. Diversification efficiency

	Default CPLEX	RINS	Guided dives
Starting from a poor solution			
Number of fails (out of 37): no improved solution found in half an hour	1	0	1
Median time (in seconds) to first improved solution ... in the 36 cases where all algorithms succeed	36.48	24.98	30.59
Starting from a good solution			
Number of fails (out of 37): no improved solution found in half an hour	32	4	14
Median time (in seconds) to first improved solution ... in the 5 cases where all algorithms succeed	284.44	9.09	53.19
... in the 23 cases where only RINS and guided dives succeed	-	99.56	119.74

Remarks

- RINS implemented in CPLEX
- Appears to be the best working approach overall
- Hybrid method between LB and RINS implies
- MIBLIB instance *swath* has value

  471.03 – best known UB
  1630.20 – by FP
  1280.95 – by OFP

  in neither case impressive

References

[1] T. Achterberg and T. Berthold. Improving the feasibility pump. ZIB-report 05-42, Konrad-Zuse-Zentrum fur Infortionstechnik Berlin, 2005.

[2] T. Achterberg, T. Koch, and A. Martin. Branching rules revisited. *Operations Research Letters*, 33:42–54, 2004.

[3] L. Bertacco, M. Fischetti, and A. Lodi. A feasibility pump heuristic for general mixed-integer problems. Technical Report OR-05-5, DEIS - Università di Bologna., 2005.

[4] E. Danna, E. Rothberg, and C. Le Pape. Exploring relaxation induced neighborhoods to improve mip solutions. *Mathematical Programming*, A(102):71–90, 2005.

[5] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, B(98):23–47, 2003.

[6] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, Inc., New York, USA, 1998.