

TOBIAS ACHTERBERG, TIMO BERTHOLD

Improving the Feasibility Pump

Improving the Feasibility Pump

Tobias Achterberg*, Timo Berthold†

September 16, 2005

Abstract

The Feasibility Pump of Fischetti, Glover, Lodi, and Bertacco [8, 7] has proved to be a very successful heuristic for finding feasible solutions of mixed integer programs. The quality of the solutions in terms of the objective value, however, tends to be poor. This paper proposes a slight modification of the algorithm in order to find better solutions. Extensive computational results show the success of this variant: in 89 out of 121 MIP instances the modified version produces improved solutions in comparison to the original Feasibility Pump.

Keywords: mixed integer programming, primal heuristics, feasibility pump

1 Introduction

A mixed integer program can be stated as

$$(\text{MIP}) \quad \min \{c^T x \mid Ax \leq b, l \leq x \leq u, x_j \in \mathbb{Z} \text{ for all } j \in I\}$$

with $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c, l, u \in \mathbb{R}^n$, and $I \subseteq N = \{1, \dots, n\}$. For solving such problems it is important to quickly find feasible solutions of high quality: A good primal bound helps to cut off suboptimal branches in the search tree of a branch-and-bound based algorithm, and improvement heuristics like Local Branching [9], guided dives, and RINS [6] can only be applied after a feasible solution has been found.

Several heuristic methods to produce feasible solutions for (MIP) have been proposed in the literature, including Hillier [14], Balas and Martin [4], Saltzman and Hillier [18], Glover and Laguna [10, 11, 12], Løkketangen and Glover [15], Glover et al. [13], Nediak and Eckstein [17], and Balas et al. [3, 5].

The so-called Feasibility Pump was proposed by Fischetti, Glover, and Lodi [8] and improved by Fischetti, Bertacco, and Lodi [7]. This heuristic turned out to be very successful in finding feasible solutions even for very hard MIP instances. However, the quality of the solutions in terms of the objective value is sometimes poor.

This paper suggests a slight modification of the Feasibility Pump. In contrast to the original version, the modified *Objective Feasibility Pump* takes the objective function c of the MIP into account during the course of the algorithm. Computational results show that the solution quality can indeed be improved by our approach without losing the ability to find feasible solutions in a reasonable amount of time.

*Konrad-Zuse-Zentrum für Informationstechnik Berlin, achterberg@zib.de

†Konrad-Zuse-Zentrum für Informationstechnik Berlin, berthold@zib.de

The rest of the paper is organized as follows. The remainder of Section 1 reviews the original version of the Feasibility Pump as described by Fischetti, Bertacco, and Lodi [7]. Section 2 introduces the modifications included in the Objective Feasibility Pump. Finally, Section 3 gives computational results on a large test set of 121 MIP instances from the MIPLIB 2003 [2], the Mittelmann test set [16], and instances from Danna et al. [6].

1.1 The Feasibility Pump

The Feasibility Pump heuristic proceeds as follows: First the LP relaxation

$$(LP) \quad \min \{c^T x \mid Ax \leq b, l \leq x \leq u\}$$

of (MIP) is solved and its solution x^* is rounded to an integer vector $\tilde{x} = [x]^I$. We define $[x]^S$ with $S \subseteq N$ as

$$[x]_j^S := \begin{cases} \lfloor x_j + 0.5 \rfloor & \text{if } j \in S \\ x_j & \text{if } j \notin S. \end{cases} \quad (1)$$

If \tilde{x} is not feasible, an additional LP is solved in order to find a new point in the LP polyhedron

$$P := \{x \in \mathbb{R}^n \mid Ax \leq b, l \leq x \leq u\}$$

that is, w. r. t. the integer variables, closest to \tilde{x} , i.e., that minimizes

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

The procedure is iterated using this point as new solution $x^* \in P$. Thereby, the algorithm creates two sequences of points: one with points x^* that fulfill the inequalities, and one with points \tilde{x} that fulfill the integrality requirements. The algorithm terminates if the two sequences converge or if a predefined iteration limit is reached.

In order to determine a nearest point $x^* := \operatorname{argmin}\{\Delta(x, \tilde{x}) \mid x \in P\}$ in P , the following LP is solved:

$$\begin{aligned} \min \quad & \Delta(x, \tilde{x}) = \sum_{j \in I: \tilde{x}_j = l_j} (x_j - l_j) + \sum_{j \in I: \tilde{x}_j = u_j} (u_j - x_j) + \sum_{j \in I: l_j < \tilde{x}_j < u_j} d_j \\ \text{s.t.} \quad & Ax \leq b \\ & d \geq x - \tilde{x} \\ & d \geq \tilde{x} - x \\ & l \leq x \leq u. \end{aligned} \quad (2)$$

The variables d_j are introduced to model the nonlinear function $d_j = |x_j - \tilde{x}_j|$ for integer variables x_j that are not equal to one of their bounds in the rounded solution \tilde{x} .

1.2 Implementational Issues

In the course of the algorithm, two main problems arise: First, the procedure can be caught in a cycle. That means, the same sequence of integer and LP-feasible points is visited over and over again. Second, the progress in driving the integer points towards feasibility might be very slow. The first problem is handled by performing a

so-called *restart* each time an integer point \tilde{x} is generated that was already visited in a prior iteration. In a restart a random perturbation step is executed, which shifts some of the variables randomly up or down and installs this perturbed vector as new integer point \tilde{x} to continue the search. The second issue is also handled by restarts if there was no large enough improvement in the “fractionality” measure

$$f(x^*) := \sum_{j \in I} f(x_j^*) \quad \text{with} \quad f(x_j^*) := |x_j^* - \lfloor x_j^* + 0.5 \rfloor|$$

in a certain number of iterations.

The Feasibility Pump as described by Fischetti, Bertacco, and Lodi [7] consists of three stages. In the first stage, a couple of iterations (so-called *pumping rounds*) are performed just on the binary variables $B \subseteq I$ by relaxing the integrality conditions on the general integer variables. If this does not yield a feasible solution, the second stage invokes pumping rounds taking all integer variables into account. As initial integer point \tilde{x} one chooses a point visited in Stage 1 which was closest to the LP polyhedron. If still no solution is found, a third stage is executed. Using a point \tilde{x} from Stage 2 closest to P , the MIP

$$\min \{ \Delta(x, \tilde{x}) \mid Ax \leq b, l \leq x \leq u, x_j \in \mathbb{Z} \text{ for all } j \in I \} \quad (3)$$

is processed by a MIP solver which stops after the first feasible solution is found. One expects that the nearly feasible point \tilde{x} has integer feasible points in its vicinity. It is therefore likely that the MIP solver finds a feasible solution early in the search process if the objective function of (3) is used.

In the implementation used by Fischetti, Bertacco, and Lodi [7], Stage 1 is stopped

- after an LP solution x^* was found with all binary variables being integral,
- the fractionality measure f could not be decreased by at least 10% in a certain number of pumping rounds, or
- a pumping round limit is reached.

The second stage is aborted for analogous reasons (using different parameter settings), or if 100 restarts have been performed in Stage 2.

2 The Objective Feasibility Pump

Finding a high-quality solution of a MIP means to find a point $x \in \mathbb{R}^n$ satisfying three conditions: $x \in P$, $x_j \in \mathbb{Z}$ for all $j \in I$, and $c^T x$ is as small as possible. The Feasibility Pump generates sequences of points fulfilling the first and the second criteria, respectively, hopefully resulting in a point which satisfies both. However, despite the computation of the starting point, which is chosen to be the optimum of the LP relaxation, the third condition is disregarded. Therefore, the solution quality is usually rather poor (see [7]).

Fischetti, Bertacco, and Lodi address this issue by either updating an objective limit each time a new solution has been found and calling the Feasibility Pump again on this restricted MIP, or by applying local search strategies like Local Branching [9] or RINS [6] to improve the solution.

We take a different approach. Instead of instantly discarding the original objective function of the MIP, we gradually reduce its influence and increase the weight

of the artificial objective function Δ of the Feasibility Pump. The hope is that we still converge to a feasible solution but concentrate the search on the region of high-quality points.

In the remainder of the paper we assume $c \neq 0$. Our modification of the Feasibility Pump replaces the distance function $\Delta(\cdot, \tilde{x})$ by a convex combination of Δ and c :

$$\Delta_\alpha(x, \tilde{x}) := (1 - \alpha)\Delta(x, \tilde{x}) + \alpha \frac{\|\Delta\|}{\|c\|} c^T x \quad \text{with } \alpha \in [0, 1]. \quad (4)$$

Here, $\|\cdot\|$ is the Euclidean norm of a vector, and Δ is the objective function vector of (2): depending on the stage either $\|\Delta\| = \sqrt{|B|}$ or $\|\Delta\| = \sqrt{|I|}$. We compute x^* using Δ_α instead of Δ in (2). In each pumping round α is geometrically decreased with a fixed factor $\varphi < 1$, i.e., $\alpha_{t+1} = \varphi \alpha_t$ and $\alpha_0 \in [0, 1]$. With increasing iteration index t , this puts the emphasis more and more towards feasibility and decreases the influence of the original objective function. Note that you can obtain the original Feasibility Pump by choosing $\alpha_0 = 0$.

The introduction of Δ_α requires a modification of the cycle detection step in the Feasibility Pump algorithm. Especially during the first pumping rounds it might happen frequently that integer points are revisited, because of the higher resemblance of subsequent functions $\Delta_\alpha(\cdot, \tilde{x})$. Reaching the same point once again, however, does not automatically implicate that the process runs into an infinite cycle like in the original heuristic. This is due to the fact that we are now using different directions Δ_{α_t} in different iterations t in contrast to the old Δ which only depends on the current integer point \tilde{x} . After α was decreased sufficiently, it is likely that the algorithm leaves the cycle. We therefore remember the visited points as pairs (\tilde{x}, α_t) and conduct a restart at iteration t only if the point \tilde{x} was already visited at iteration $t' < t$ with $\alpha_{t'} - \alpha_t \leq \delta_\alpha$ and $\delta_\alpha \in [0, 1]$ being a fixed parameter value.

The pseudocode of the modified Feasibility Pump reads as follows.

Algorithm 2.1 (Objective Feasibility Pump)

Stage 1:

1. Initialize $x^* := \operatorname{argmin}\{c^T x \mid x \in P\}$, $\tilde{x} := [x^*]^B$, $S := B$, $t := 0$, $\mathit{maxIter} := \mathit{maxIter}_{ST1}$, $\mathit{maxStalls} := \mathit{maxStalls}_{ST1}$, $\mathit{restarts} := 0$, $L := \emptyset$.
2. If \tilde{x} did not change since the last iteration, round the T most fractional variables x_j^* to the other side compared to \tilde{x}_j (with T being a parameter setting).
3. While there exists $(\tilde{x}', \alpha_{t'}) \in L$ with $\tilde{x}' = \tilde{x}$ and $\alpha_{t'} - \alpha_t \leq \delta_\alpha$, perform a random perturbation on \tilde{x} (see [8, 7]) and set $\mathit{restarts} := \mathit{restarts} + 1$.
4. If \tilde{x} is feasible for (MIP) \rightarrow stop.
5. Set $L := L \cup \{(\tilde{x}, \alpha_t)\}$. Set $t := t + 1$, $\alpha_t := \varphi \alpha_{t-1}$.
6. If $t > \mathit{maxIter}$, goto next stage.
7. Solve $x^* := \operatorname{argmin}\{\Delta_{\alpha_t}(x, \tilde{x}) \mid x \in P\}$.
8. If $x^* = \tilde{x}$, goto next stage.
9. If $f(x^*)$ did not decrease by at least 10% in the last $\mathit{maxStalls}$ pumping rounds, goto next stage.
10. Set $\tilde{x} := [x^*]^S$. Goto Step 2.

Stage 2:

1. Initialize \tilde{x} to be an integer point of Stage 1 with minimal $\Delta(x, \tilde{x})$, $x \in P$. Set $t := 0$, $S := I$, $maxIter := maxIter_{ST2}$, $maxStalls := maxStalls_{ST2}$, $restarts := 0$, $L := \emptyset$.
2. Perform Steps 2 to 10 of Stage 1, but if $restarts > maxRestarts$ in Step 3, goto Stage 3.

Stage 3:

1. Solve MIP (3) with \tilde{x} being an integer point of Stage 2 with minimal $\Delta(x, \tilde{x})$, $x \in P$. Stop after the first feasible solution has been found.

3 Computational Results

This section compares the performances of the Feasibility Pump described in [7] and the Objective Feasibility Pump described in this paper. All computations were performed on a 3.4 GHz Pentium-4 with 512 KB cache and 3 GB RAM. CPLEX 9.03 was used as underlying LP solver. We used a time limit of one hour in all runs.

3.1 Test Set and Settings

The computations were performed on a wide test set consisting of 121 instances taken from

- MIPLIB 2003 [2],
- the MIP collection of Mittelmann [16], and
- the instances used in [7], which are described in [6].

In all runs we used the parameter settings for the Feasibility Pump as suggested in [7], as follows. The maximum number of total iterations for Stages 1 and 2 were set to $maxIter_{ST1} = 10000$ and $maxIter_{ST2} = 2000$. The maximum number of iterations without a pumping cycle of at least 10% improvement were set to $maxStalls_{ST1} = 70$ and $maxStalls_{ST2} = 600$. For $j \in S$, the rounding (1) is modified to $[x]_j^S := [x_j + \rho(z)]$ with $z \in [0, 1]$ chosen uniformly at random and

$$\rho(z) = \begin{cases} 2z(1-z) & \text{if } z \leq \frac{1}{2} \\ 1-2z(1-z) & \text{if } z > \frac{1}{2}. \end{cases}$$

The shifting in Step 2 is applied on a random number of $T \in [10, 30]$ variables, but only variables x_j with current fractionality $f(x_j^*) > 0.02$ are regarded as shifting candidates. For the unmodified Feasibility Pump we set $\alpha_0 = 0$, for the Objective Feasibility Pump we set $\alpha_0 = 1$, $\varphi = 0.9$, and $\delta_\alpha = 0.005$.

Because we wanted to investigate the performance of the Feasibility Pump used as heuristic inside a MIP solver, we applied the MIP preprocessing of CPLEX prior to running the Feasibility Pump algorithm itself. This usually avoids difficulties with the scaling of degenerated objective functions in the modified distance function Δ_α , see (4). For example, some instances in our test set have objective functions consisting of only a single artificial variable which is defined as a linear combination of several other variables by an equality constraint. Such equations lead to unbalanced situations in the calculation of Δ_α , since in this case the norm $\|\Delta\|$ is misleading. We observed that MIP preprocessing usually resolves this issue.

3.2 Results

Table 1 and Table 2 compare the performance of the two Feasibility Pump versions. The left hand sides show the results of the original version proposed by Fischetti, Bertacco, and Lodi [7]. The right hand sides show the results of the Objective Feasibility Pump described in this paper. The column ‘Objective’ contains the objective values of the feasible solutions that were found with either algorithm. A bar ‘–’ means that no solution was found in the time limit of one hour. ‘Gap’ denotes the percentage gap γ to the optimal or best known solution of the corresponding instance. It is printed in bold face if the corresponding algorithm produced a solution with a better or equal value than the other version of the Feasibility Pump. The gap is calculated as

$$\gamma := 100 \cdot \frac{(\tilde{c} - c^*)}{|c^*|}$$

with \tilde{c} being the value of the heuristic solution and c^* being the optimal or best known solution value of the instance. If $\tilde{c} = c^*$ we define $\gamma := 0$. If $\tilde{c} > c^*$ and $c^* = 0$ we define $\gamma := \infty$. The instances displayed in italics in the tables are those for which we do not know the optimal solution. In this case we compare to the best solution we know, which was either generated by CPLEX 9.1 running for an hour with default settings, retrieved from the MIPLIB 2003 web site [1], or produced by one of the Feasibility Pump versions.

Column ‘Time’ shows the time in seconds to find a solution. ‘Iter’ and ‘Rst’ represent the total number of pumping rounds and restarts, respectively, applied in Stage 1 and Stage 2. The stage at which the solution was found is shown in Column ‘St’. The geometric means at the bottom of Table 2 are calculated over all instances for which a solution was found by both versions of the Feasibility Pump. In the calculations of the geometric means individual values smaller than 1 are replaced by 1.

As one can see in the tables the Objective Feasibility Pump produced a strictly better solution in 89 out of 121 cases, whereas the unmodified Feasibility Pump ranked first in 17 cases. On 11 instances both versions computed the same objective value, and for 4 instances both versions were not able to find any feasible solution within one hour. Only on one instance, namely *acc-6*, the original version could find a solution while the Objective Feasibility Pump did not succeed.

The running times usually differ only slightly in terms of absolute numbers. Only on some instances the Objective Feasibility Pump was significantly slower, namely on *air04*, *dano3mip*, *momentum3*, *mzzv42z*, *rd-rplusc-21*, the three *dano* instances, on *qap10*, *acc-6*, and *neos16*. However, for all those instances except *acc-6* a better solution was found. The Objective Feasibility Pump was substantially faster on *momentum1*, *protfold*, *t1717*, *icir97_potential*, and *neos10*. Nevertheless, the solutions on these instances are at least as good as the ones of the unmodified Feasibility Pump. The quality improvement can also be seen in the geometric means: The mean gap was reduced from 55.0% to 29.5%, while the running time increased only slightly.

The different behavior of the two Feasibility Pump versions is displayed in Figure 1 for selected problem instances. The figures show the evolution of the objective values of the LP solutions x^* and their fractionalities $f(x^*)$ during the course of the algorithm. The graphs on the left hand side arise from the unmodified version, while the ones on the right hand side arise from the Objective Feasibility Pump.

In the upper plots (*aflow40b*) one can see that the original version of the Feasibility Pump rapidly left the region of small objective values while the fractionality

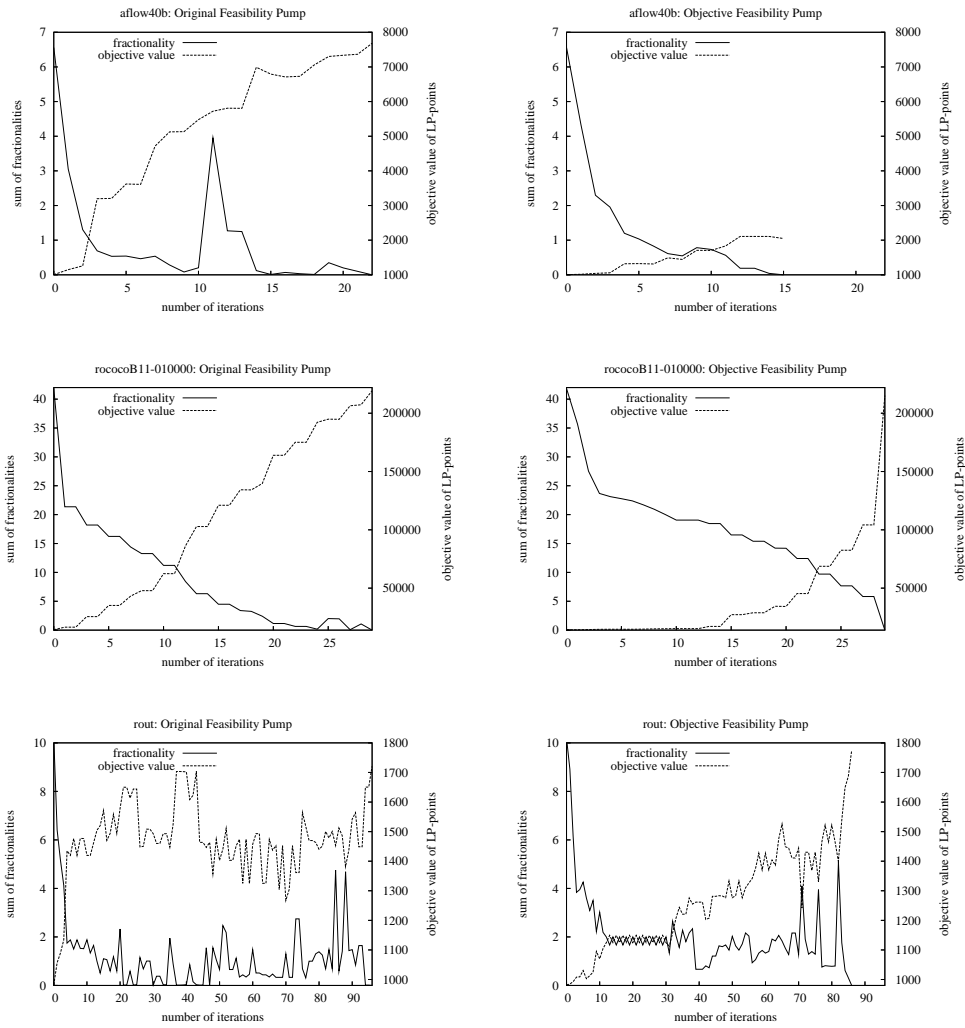


Figure 1. Evolution of the objective value and sum of fractionalities for aflow40b, rococoB11-010000, and rout.

measure decreased quite fast. However, the algorithm was not able to drive the solution to integrality before restarting at iteration 10. At this point, the objective value was already far away from the optimal solution. In contrast, the Objective Feasibility Pump stayed much closer to the optimal solution value of 1168 but did not decrease the fractionality measure as fast. Nevertheless, after 15 iterations a feasible solution was found with an objective value that was already exceeded after four iterations of the unmodified Feasibility Pump.

The two plots of `rococoB11-010000` show an example where both versions produced a similar solution in the same number of iterations, although the two algorithms behaved differently. Again, the unmodified Feasibility Pump increased the objective value and decreased the fractionality faster than the Objective Feasibility Pump.

The bottom plots (`rcout`) show a situation where the Objective Feasibility Pump is inferior. The original version performed 35 restarts, most of which can be seen as spikes in the fractionality graph. The last random perturbation at iteration 96 “coincidentally” produced a feasible solution. The Objective Feasibility Pump did not succeed to drive the fractionality to a value less than 0.5 until the last two iterations. Only six restarts were performed. Interestingly, the cycle between two points from iteration 12 to 30 was left without a restart just by decreasing α .

As already shown by Fischetti, Glover, and Lodi [8] and Fischetti, Bertacco, and Lodi [7] the Feasibility Pump is a useful heuristic for mixed integer programming, because it usually finds feasible solutions in a reasonable amount of time. Our results show that the modification presented in this paper further improves the Feasibility Pump: The quality of the resulting solutions is substantially enhanced with only a slight increase in the running time.

References

- [1] T. Achterberg, T. Koch, and A. Martin. The mixed integer programming library: MIPLIB 2003. <http://miplib.zib.de>.
- [2] T. Achterberg, T. Koch, and A. Martin. MIPLIB 2003. *Operations Research Letters*, 2005. to appear.
- [3] E. Balas, S. Ceria, M. Dawande, F. Margot, and G. Pataki. OCTANE: A new heuristic for pure 0-1 programs. *Operations Research*, 49(2):207–225, 2001.
- [4] E. Balas and C. H. Martin. Pivot-and-complement: A heuristic for 0-1 programming. *Management Science*, 26(1):86–96, 1980.
- [5] E. Balas, S. Schmieta, and C. Wallace. Pivot and shift - a mixed integer programming heuristic. Technical report, Carnegie Mellon University, 2004.
- [6] E. Danna, E. Rothberg, and C. Le Pape. Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming*, 102(1), 2004.
- [7] M. Fischetti, L. Bertacco, and A. Lodi. A feasibility pump heuristic for general mixed-integer problems. Technical report, Università di Bologna – D.E.I.S. – Operations Research, 2005.
- [8] M. Fischetti, F. Glover, and A. Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2005.
- [9] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1–3):23–47, 2003.
- [10] F. Glover and M. Laguna. General purpose heuristics for integer programming - part I. *Journal of Heuristics*, 3, 1997.

- [11] F. Glover and M. Laguna. General purpose heuristics for integer programming - part II. *Journal of Heuristics*, 3, 1997.
- [12] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publisher, Boston, Dordrecht, London, 1997.
- [13] F. Glover, A. Løkketangen, and D. L. Woodruff. Scatter search to generate diverse MIP solutions. In M. Laguna and J. L. González-Velarde, editors, *OR Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, pages 299–317. Kluwer Academic Publishers, 2000.
- [14] F. S. Hillier. Efficient heuristic procedures for integer linear programming with an interior. *Operations Research*, 17:600–637, 1969.
- [15] A. Løkketangen and F. Glover. Solving zero/one mixed integer programming problems using tabu search. *European Journal of Operations Research*, 106:624–658, 1998.
- [16] H. Mittelmann. Decision tree for optimization software: Benchmarks for optimization software, 2003. <http://plato.asu.edu/bench.html>.
- [17] M. Nediak and J. Eckstein. Pivot, cut, and dive: A heuristic for 0-1 mixed integer programming. Technical Report RRR 53-2001, Rutgers University, 2001.
- [18] R. M. Saltzman and F. S. Hillier. A heuristic ceiling point algorithm for general integer linear programming. *Management Science*, 38(2):263–283, February 1992.

Name	Original Feasibility Pump						Objective Feasibility Pump					
	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
<i>a1c1s1</i>	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
<i>arki001</i>	7.75064e+06	2.2	12	797	67	3	7.70474e+06	1.6	10	737	62	3
<i>atlanta-ip</i>	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
<i>dano3mip</i>	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
<i>ds</i>	-	-	>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
<i>glass4</i>	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
<i>liu</i>	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
<i>momentum1</i>	359238	3.7	818	416	186	3	346535	0.0	223	351	118	3
<i>momentum2</i>	-	-	>3600	504	139	3	-	-	>3600	522	158	3
<i>momentum3</i>	509585	37.7	272	148	26	1	420724	13.7	599	173	13	1
<i>msc98-ip</i>	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
<i>protfold</i>	-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
<i>rd-rplusc-21</i>	173065	1.1	375	637	120	3	171182	0.0	790	525	109	3
<i>roll3000</i>	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
<i>sp97ar</i>	9.57074e+08	44.0	3	1	0	1	9.40566e+08	41.5	3	9	0	1
<i>stp3d</i>	-	-	>3600	26	2	1	-	-	>3600	17	0	1
<i>swath</i>	1630.8	192.4	3	107	13	1	1280.95	129.6	13	238	39	2
<i>t1717</i>	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
<i>timtab2</i>	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

Name	Original Feasibility Pump						Objective Feasibility Pump					
	Objective	Gap %	Time	Iter	Rst	St	Objective	Gap %	Time	Iter	Rst	St
bell3a	9.85707e+07	11121.2	0	78	35	2	7.21256e+07	8110.7	0	63	6	2
bell5	4.81498e+07	437.0	0	29	10	2	4.08948e+07	356.1	0	148	37	2
gesa3	3.5368e+07	26.4	0	21	5	2	2.89813e+07	3.5	1	29	0	2
gesa3_o	6.76543e+07	141.7	1	29	8	2	2.87697e+07	2.8	0	29	0	2
l152lav	4781	1.2	0	11	1	1	4757	0.7	0	22	0	1
stein45	45	50.0	0	1	0	1	35	16.7	0	9	0	1
ran8x32	6033	15.0	0	2	0	1	5817	10.9	0	9	0	1
ran10x26	5050	18.3	0	2	0	1	4833	13.2	0	9	0	1
ran12x21	4330	18.2	0	2	0	1	4231	15.5	0	8	0	1
ran13x13	3705	13.9	0	2	0	1	3820	17.5	0	9	0	1
binkar10_1	7170.23	6.3	0	3	0	1	7156.21	6.1	1	3	0	1
eilD76	1616.97	82.6	10	530	113	3	2300.06	159.8	10	575	106	3
irp	12715.3	4.6	2	9	1	1	12162.4	0.0	1	5	0	1
mas284	105336	15.2	0	3	0	1	99522.7	8.9	0	108	8	2
prod1	-42	25.0	0	29	9	1	-53	5.4	0	29	0	1
bc1	3.52343	5.5	2	1	0	1	5.4391	62.9	2	7	0	1
bienst1	72.9757	56.1	0	7	3	1	55.5	18.7	0	57	8	1
bienst2	88.2326	61.6	0	7	4	1	73.6667	34.9	1	60	7	1
dano3_3	629.604	9.2	25	1	0	1	576.345	0.0	47	12	0	1
dano3_4	646.702	12.2	26	1	0	1	576.435	0.0	76	33	0	1
dano3_5	667.574	15.7	26	1	0	1	576.994	0.0	106	21	0	1
mkc1	-522.815	13.9	0	3	0	1	-563.1	7.3	0	7	0	1
neos1	85	347.4	1	189	36	2	68	257.9	1	82	8	1
neos2	898.216	97.5	8	829	117	3	958.977	110.8	7	846	102	3
neos3	1278.4	2241.4	12	809	110	3	1630.21	2885.7	8	762	102	3
neos4	-4.81256e+10	1.0	3	264	152	3	-4.8132e+10	1.0	3	226	119	3
neos5	-4.81256e+10	1.0	4	264	152	3	-4.8132e+10	1.0	3	226	119	3
neos6	137	65.1	4	26	3	1	93	12.0	12	37	0	1
neos7	5.2039e+06	620.8	0	52	23	2	1.10593e+06	53.2	1	91	17	2
nug08	214	0.0	2	2	0	1	214	0.0	3	2	0	1
qap10	442	30.0	27	1	0	1	386	13.5	62	5	0	1
seymour1	435.9	6.1	2	1	0	1	427.063	4.0	3	6	0	1
swath1	499.711	31.8	1	4	0	1	439.106	15.8	2	44	3	1
swath2	1337.12	247.1	0	10	2	1	641.544	66.5	2	40	1	1
acc-0	0	0.0	0	2	0	1	0	0.0	0	2	0	1
acc-1	0	0.0	2	4	0	1	0	0.0	1	3	0	1
acc-2	0	0.0	2	7	0	1	0	0.0	3	7	0	1
acc-3	0	0.0	10	6	0	1	0	0.0	12	7	0	1
acc-4	-	>3600	323	141	3		-	>3600	247	124	3	
acc-5	0	0.0	1847	316	116	3	0	0.0	2054	300	110	3
acc-6	0	0.0	1017	305	116	3	-	>3600	481	109	3	
ic97_potential	4568	14.7	2	752	24	3	4433	11.3	2	797	20	3
ic97_tension	4487	13.8	1	707	74	3	4539	15.1	1	958	105	3
icir97_tension	7309	13.8	8	718	117	3	7288	13.4	9	783	113	3
icir97_potential	7724	16.6	62	771	17	3	7526	13.7	24	807	16	3
nh97_potential	1598	12.7	10	760	32	3	1554	9.6	17	902	41	3
nh97_tension	1575	4.2	10	1076	59	3	1511	0.0	11	1634	110	3
B10-011000	117462	499.1	1	21	2	1	108472	453.3	1	29	0	1
B10-011001	117109	444.5	0	20	2	1	108472	404.3	1	29	0	1
B11-010000	219275	547.4	2	29	2	1	215163	535.2	1	29	0	1
B11-110001	206342	342.0	3	29	1	1	208823	347.3	4	29	0	1
B12-111111	80931	89.0	46	1368	104	3	83096	94.1	35	855	66	3
C10-001000	255030	2124.4	0	29	4	1	159137	1288.0	0	29	0	1
C10-100001	239789	1135.3	1	29	4	1	134440	592.6	1	29	0	1
C11-010100	146524	450.4	1	33	5	1	136976	414.6	2	29	0	1
C11-011100	130241	488.1	1	9	1	1	128149	478.7	1	29	0	1
C12-100000	534256	1282.7	4	29	0	1	456140	1080.5	5	29	0	1
C12-111100	115593	209.1	2	7	0	1	113210	202.8	2	26	0	1
neos10	2	100.2	41	58	10	1	2	100.2	5	21	0	1
neos16	458	2.0	88	696	67	3	451	0.4	383	857	104	3
neos20	-199	54.1	3	711	48	3	-199	54.1	7	748	70	3
Geom. Mean		55.0	3.2	17.1	4.3			29.5	3.8	53.4	4.6	

Table 2. Comparison of original feasibility pump and objective sensitive version (cont'd)