Using the analytic center in the feasibility pump

Daniel Baena Jordi Castro
Dept. of Stat. and Operations Research
Universitat Politècnica de Catalunya
daniel.baena@upc.edu jordi.castro@upc.edu
Research Report UPC-DEIO DR 2010-03
August 2010

Report available from http://www-eio.upc.es/~jcastro

Using the analytic center in the feasibility pump

Daniel Baena[a], Jordi Castro[*,a]

[a]*Dept. of Statistics and Operations Research, Universitat Politècnica de Catalunya, Barcelona, Catalonia, Spain*

Abstract

The feasibility pump (FP) [5, 7] has proved to be a successful heuristic for finding feasible solutions of mixed integer linear problems (MILPs). FP was improved in [1] for finding better quality solutions. Briefly, FP alternates between two sequences of points: one of feasible solutions for the relaxed problem (but not integer), and another of integer points (but not feasible for the relaxed problem). Hopefully, the procedure may eventually converge to a feasible and integer solution. Integer points are obtained from the feasible ones by some rounding procedure. This short paper extends FP, such that the integer point is obtained by rounding a point on the (feasible) segment between the computed feasible point and the analytic center for the relaxed linear problem. Since points in the segment are closer (may be even interior) to the convex hull of integer solutions, it may be expected that the rounded point has more chances to become feasible, thus reducing the number of FP iterations. When the selected point to be rounded is the feasible solution of the relaxation (i.e., one of the two end points of the segment), this analytic center FP variant behaves as the standard FP. Computational results show that this variant may be efficient in some MILP instances.

Key words: Analytic Center, Interior-point Methods, Mixed-integer Linear Programming, Feasibility Problem, Primal Heuristics

---

1. Introduction

The problem of finding a feasible solution of a generic mixed integer linear problem (MILP) of the form

$$\begin{array}{ll} \min\limits_{x} & c^T x \\ \text{s. to} & Ax = b \\ & x \geq 0 \\ & x_j \quad \textit{integer} \quad \forall j \in \mathcal{I}, \end{array} \tag{1}$$

where $A \in \mathbb{R}^{mxn}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$ and $\mathcal{I} \subseteq \mathcal{N} = \{1, \ldots, n\}$, is a NP-hard problem. In [5, 7] the authors proposed a new heuristic approach to compute MILP solutions, named the *feasibility pump* (FP). This heuristic turned out to be successful in finding feasible solutions even for some hard MILP instances. A slight modification of FP was suggested in [1], named the *objective feasibility*

---

[*]Corresponding address: Dept. of Statistics and Operations Research, Universitat Politècnica de Catalunya, Campus Nord, Office C5218, Jordi Girona 1–3, 08034 Barcelona, Catalonia, Spain.

Email addresses: `daniel.baena@upc.edu` (Daniel Baena), `jordi.castro@upc.edu` (Jordi Castro)

Preprint submitted to *October 4, 2010*

initialize $t := 0$ and $x^* := \arg\min\{c^T x : Ax = b, x \geq 0\}$
if $x^*_{\mathcal{I}}$ is integer **then** return(x^*) **end if**
$\tilde{x} := [x^*]$ (rounding of x^*)
while time < TimeLimit **do**
    $x^* := \arg\min\{\triangle(x, \tilde{x}) : Ax = b, x \geq 0\}$
    if $x^*_{\mathcal{I}}$ is integer **then** return(x^*) **end if**
    if $\exists j \in \mathcal{I} : [x^*_j] \neq \tilde{x}_j$ **then**
        $\tilde{x} := [x^*]$
    else
        restart
    end if
    $t := t + 1$
end while
return(FP failed)

Figure 1: The feasibility pump heuristic (original version)

pump, in order to improve the quality of the solutions in terms of the objective value. The main difference between both versions is that the objective FP, in contrast to the original version, takes the objective function of the MILP into account during the course of the algorithm. FP alternates between feasible (for the linear relaxation of MILP) and integer points, hopefully converging to a feasible integer solution. The integer point is obtained by applying some rounding procedure to the feasible solution. This paper suggests an extension of FP where all the points in a feasible segment are candidates to be rounded. The end points of this segment are the feasible point of the standard FP and some interior point of the polytope of the relaxed problem, the analytic center being the best candidate. When the end point of the segment in the boundary of the polytope is considered for rounding, we obtain the standard FP algorithm. The motivation of this approach is that rounding a point of the segment closer to the analytic center may increase the chances of obtaining an integer point, in some instances, thus reducing the number of FP iterations. Although interior-point methods have been applied in the past in branch-and-bound frameworks for MILP and mixed integer nonlinear problems (MINLP) [3, 4, 11, 12], as far as we know this is the first attempt to apply them to a primal heuristic. The computational results show that, for some instances, taking a point in the interior of the feasible segment may be more effective than the standard end point of the objective FP. A recent version of FP [8] introduced a new improved rounding scheme based on constraint propagation. Although in this work we considered as base code a freely available implementation of the objective FP, the analytic center FP approach could also be used with the above new rounding scheme.

The paper is organized as follows. The remainder of Section 1 reviews the original FP version of [5, 7] and the modified objective FP of [1]. Section 2 introduces the analytic center FP variant. Finally, Section 3 reports computational results on a subset of MILP instances from MIPLIB 2003 [2].

1.1. The original feasibility pump

The FP heuristic starts by solving the linear programming (LP) relaxation of (1)

$$\min_x \{c^T x \ : \ Ax = b, x \geq 0\}, \tag{2}$$

and its solution x^* is rounded to an integer point $\tilde{x}$, which may be infeasible for (2). The rounding $\tilde{x}$ of a given x^*, denoted as $\tilde{x} = [x^*]$, is obtained by setting $\tilde{x}_j = [x^*_j]$ if $j \in \mathcal{I}$ and $\tilde{x}_j = x^*_j$ otherwise, where $[\cdot]$ represents scalar rounding to the nearest integer. If $\tilde{x}$ is infeasible, FP finds the closest $x^* \in P$, where

$$P = \{x \in \mathbb{R}^n \; : \; Ax = b, x \geq 0\}, \tag{3}$$

by solving the following LP

$$x^* = \arg\min\{\triangle\,(x, \tilde{x}) : Ax = b, x \geq 0\}, \tag{4}$$

$\triangle\,(x, \tilde{x})$ being defined (using the L_1 norm) as

$$\triangle\,(x, \tilde{x}) = \sum_{j \in \mathcal{I}} |x_j - \tilde{x}_j|. \tag{5}$$

Notice that continuous variables $\tilde{x}_j$, $j \notin \mathcal{I}$, don't play any role. If $\triangle\,(x^*, \tilde{x}) = 0$ then $x^*_j (= \tilde{x}_j)$ is integer for all $j \in \mathcal{I}$, so x^* is a feasible solution for (1). If not, FP finds a new integer point $\tilde{x}$ from x^* by rounding. The pair of points $(\tilde{x}, x^*)$ with $\tilde{x}$ integer and $x^* \in P$ are iteratively updated at each FP iteration with the aim of reducing as much as possible the distance $\triangle\,(x^*, \tilde{x})$. An outline of the FP algorithm is showed in Figure 1. To avoid that the procedure gets stuck at the same sequence of integer and feasible, there is a restart procedure when the previous integer point $\tilde{x}$ is revisited (lines 7–11 of algorithm of Figure 1). In a restart, a random perturbation step is performed.

The FP implementation has three stages. *Stage 1* is performed just on the binary variables by relaxing the integrality conditions on the general integer variables. In *stage 2* FP takes all integer variables into account. The FP algorithm exits stage 1 and goes to stage 2 when either (a) a feasible point with respect to only the binary variables has been found; (b) the minimum $\triangle\,(x^*, \tilde{x})$ was not updated during a certain number of iterations; or (c) the maximum number of iterations was reached. The point $\tilde{x}$ that produced the smallest $\triangle\,(x^*, \tilde{x})$ is stored and passed to stage 2 as the initial $\tilde{x}$ point. When FP turns out to be unable to find a feasible solution within the provided time limit, the default procedure of the underlying MILP solver (CPLEX 12 [10] in this work) is started; this is named *stage 3*.

1.2. The modified objective feasibility pump

According to [1], although the original FP heuristic of [5, 7] has proved to be a very successful heuristic for finding feasible solutions of mixed integer programs, the quality of their solutions in terms of objective value tends to be poor. In the original FP algorithm of [5, 7] the objective function of (1) is only used at the beginning of the procedure. The purpose of the objective FP [1] is, instead of instantly discarding the objective function of (1), to consider a convex combination of it and $\triangle\,(x, \tilde{x})$, reducing gradually the influence of the objective term. The hope is that FP still converges to a feasible solution but it concentrates the search on the region of high-quality points. The modified objective function $\triangle_\alpha\,(x, \tilde{x})$ is defined as

$$\triangle_\alpha\,(x, \tilde{x}) := (1 - \alpha)\,\triangle\,(x, \tilde{x}) + \alpha \frac{\|\triangle\|}{\|c\|} c^T x, \quad \alpha \in [0, 1], \tag{6}$$

where $\|\cdot\|$ is the Euclidean norm of a vector, and $\triangle$ is the objective function vector of $\triangle\,(x, \tilde{x})$ (i.e., at stage 1 is the number of binary variables, and at stage 2 is the number of integer (both general integer and binary) variables). At each FP iteration α is geometrically decreased with a fixed factor $\varphi < 1$, i.e., $\alpha_{t+1} = \varphi \alpha_t$ and $\alpha_0 \in [0, 1]$. Notice that the original FP algorithm is obtained using $\alpha_0 = 0$. The objective FP algorithm is basically the same as the original FP algorithm of Figure 1, replacing $\triangle(x, \tilde{x})$ by $\triangle_{\alpha_t}(x^*, \tilde{x})$ at line 5, performing at the beginning the initialization of α_0, and adding at the end of the loop $\alpha_{t+1} = \varphi\alpha_t$.

2. The analytic center feasibility pump

2.1. The analytic center

Given the LP relaxation (2), its analytic center is defined as the point $\bar{x} \in P$ that minimizes the *primal potential function* $-\sum_{i=1}^n \ln x_i$, i.e.,

$$\begin{array}{rl} \bar{x} = \arg\min\limits_x & -\sum_{i=1}^n \ln x_i \\ \text{s. to} & Ax = b \\ & x > 0. \end{array} \tag{7}$$

Note that constraints $x > 0$ could be avoided, since the domain of ln are the positive numbers. Problem (7) is a linearly constrained strictly convex optimization problem. It is easily seen that the objective function $\min -\sum_{i=1}^n \ln x_i$ is equivalent to $\max \prod_{i=1}^n x_i$. Therefore, the analytic center provides the point that maximizes the distance to the hyperplanes $x_i = 0, i = 1, \dots, n$, and it is thus expected to be well centered in the interior of the polytope P. We note that the analytic center is not a topological property of a polytope, and it depends on how P is defined. In this sense, redundant inequalities may change the location of the analytical center. Additional details can be found in [13].

The analytic may be computed by solving the KKT conditions of (7)

$$\begin{array}{rcll} Ax & = & b & \\ A^T y + s & = & 0 & \\ x_i s_i & = & 1 & i = 1, \dots, n \\ (x, s) & > & 0, & \end{array} \tag{8}$$

$y \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ being the Lagrange multipliers of $Ax = b$ and $x > 0$ respectively. Alternatively, and in order to use an available highly efficient implementation, the analytic center was computed in this work by applying a primal-dual path-following interior-point algorithm to the barrier problem of (2), after removing the objective function term (i.e., setting $c = 0$):

$$\begin{array}{rl} \min\limits_x & -\mu \sum_{i=1}^n \ln x_i \\ \text{s. to} & Ax = b \\ & x > 0, \end{array} \tag{9}$$

where μ is a positive parameter (the parameter of the barrier) that tends to zero. The arc of solutions of (9) $x^*(\mu)$ is named the central path. The central path converges to the analytic center of the optimal set. When $c = 0$ (as in (9)) the central path converges to the analytic center of the feasible set P [13].

2.2. Using the analytic center in the feasibility pump heuristic

Once the analytic center has been computed, it can be used to (in theory infinitely) increase the number of feasible points candidates to be rounded. Instead of rounding, at each FP iteration, the feasible point $x^* \in P$, points on the segment

$$x(\gamma) = \gamma\bar{x} + (1-\gamma)x^* \quad \gamma \in [0, 1] \tag{10}$$

initialize $t := 0$, $\alpha_0 \in [0, 1]$, $\varphi \in [0, 1]$, and $x^* := \arg\min\{c^T x : Ax = b, x \geq 0\}$
{ *Beginning of stage 0*}
compute analytic center $\overline{x} := \arg\min\left\{-\sum_{i=1}^{n} \ln x_i : Ax = b, x > 0\right\}$
for $\gamma \in [0, 1]$ **do**
    $x(\gamma) := \gamma\overline{x} + (1 - \gamma)x^*$
    $\tilde{x}(\gamma) := [x(\gamma)]$ (rounding of $x(\gamma)$)
    if $\tilde{x}(\gamma)$ is feasible **then** return($\tilde{x}(\gamma)$) **end if**
end for
{ *End of stage 0*}
select $\tilde{x}$ from the set $\{\tilde{x}(\gamma)\}$
while time < TimeLimit **do**
    $x^* := \arg\min\{\triangle_{\alpha_t}(x, \tilde{x}) : Ax = b, x \geq 0\}$
    for $\gamma \in [0, 1]$ **do**
        $x(\gamma) := \gamma\overline{x} + (1 - \gamma)x^*$
        $\tilde{x}(\gamma) := [x(\gamma)]$ (rounding of $x(\gamma)$)
        if $\tilde{x}(\gamma)$ is feasible **then** return($\tilde{x}(\gamma)$) **end if**
    end for
    select $\hat{x}$ from the set $\{\tilde{x}(\gamma)\}$
    if $\hat{x}_{\mathcal{I}} \neq \tilde{x}_{\mathcal{I}}$ **then**
        $\tilde{x} := \hat{x}$
    else
        restart
    end if
    $\alpha_{t+1} := \varphi\alpha_t$
    $t := t + 1$
end while
return(FP failed)

Figure 2: The analytic center feasibility pump heuristic

will be considered. Note that the segment is feasible, since it is a convex combination of two feasible points.

The analytic center FP first considers a *stage 0* (which is later applied at each FP iteration) where several $x(\gamma)$ points are tested, from $\gamma = 0$ to $\gamma = 1$ (i.e, from x^* to $\overline{x}$). Each $x(\gamma)$ is rounded to $\tilde{x}(\gamma)$. If $\tilde{x}(\gamma)$ is feasible, then a feasible integer solution was found and the procedure is stopped at the stage 0. Otherwise the algorithm proceeds with the next stage of FP, considering two different options:

a) using the point $\tilde{x}(0) = [x^*]$ (option $\gamma = 0$);
b) using the point $\tilde{x}(\gamma)$ that minimizes $||\tilde{x}(\gamma) - x(\gamma)||_\infty$ (option L_∞).

If the first option is applied at each FP iteration, and no feasible $\tilde{x}(\gamma)$ for $\gamma > 0$ is found, the analytic center FP behaves as the standard FP algorithm. In the second option, if no feasible $\tilde{x}(\gamma)$ is found, the procedure selects the $x(\gamma)$ which is closer to $[x(\gamma)]$ according to the L_∞ norm. The aim is to select the point with more chances to become both integer and feasible, in an attempt to reduce the number of FP iterations. This second option provided better results in general and it was used in the computational results of Section 3. It is worth to note that if the rounding of several $x(\gamma)$ points is feasible, the procedure selects the one with a lower γ, i.e., the one closer to x^* (instead of the one closer to the analytic center $\overline{x}$), since this point was computed considering the objective function (for $\alpha > 0$). An outline of the algorithm is shown in Figure 2.

3. Computational results

The analytic center FP was implemented using the base code of the objective FP, freely available from `http://www.or.deis.unibo.it/research_pages/ORcodes/FP-gen.html`. The base FP implementation was extended for computing the analytic center using three different interior-point solvers, CPLEX [10], GLPK [9] and PCx [6]. The new code can be obtained from the authors on request. CPLEX integrates better with the rest of the FP code, which also relies on CPLEX, and it also turned out to be significantly more efficient than GLPK and PCx. On the other hand, even deactivating all the preprocessing options and removing the crossover postprocess, CPLEX was not always able to provide the analytic center of P because of its aggressive reduced preprocessing (which can not be deactivated as we were told by CPLEX developers). For instance, for $P = \{x : \sum_{i=1}^{n} = n, x \geq 0\}$, the barrier option of CPLEX did not apply the interior-point algorithm, not providing an interior solution (i.e., it provided $x_i = n$, $x_j = 0$, $j \neq i$), whereas both GLPK and PCx reported the right analytic center $x_i = 1, i = 1, \ldots, n$. Of the other two solvers, PCx turned out to be much more efficient than GLPK. Indeed, PCx may handle upper bounds implicitly (i.e., $0 \leq x \leq 1$ from linear relaxations of $x \in \{0, 1\}$) in its interior-point implementation, whereas GLPK transforms the problem to the standard form (replacing $x \leq 1$ by $x + s = 1, s \geq 0$), significantly increasing the size of the Newton's system to be solved at each interior-point iteration.

The analytic center FP implementation was applied to a subset of MIPLIB2003 instances, whose dimensions are shown in Table 1. Columns "rows", "cols", "nnz", "int", "bin" and "con" provide respectively the number of constraints, variables, nonzeros, general integer variables, binary variables, and continuous variables of the instances. Column "objective" shows the optimal objective function. Unknown optimal objectives are marked with a "?".

Table 2 shows the results obtained for the objective FP, the analytic center FP using PCx, and the analytic center FP using CPLEX-12.1. For each variant, Table 2 reports the number of FP iterations (columns "niter"), objective value of feasible point found ("fobj"), gap between

Instance	rows	cols	nnz	int	bin	con	objective
10teams	230	2025	12150	0	1800	225	924
a1c1s1	3312	3648	10178	0	192	3456	11503.40
aflow30a	479	842	2091	0	421	421	1158
aflow40b	1442	2728	6783	0	1364	1364	1168
air04	823	8904	72965	0	8904	0	56137
air05	426	7195	52121	0	7195	0	26374
arki001	1048	1388	20439	96	415	877	7580810
atlanta-ip	21732	48738	257532	106	46667	1965	90.00
cap6000	2176	6000	48243	0	6000	0	-2451380
dano3mip	3202	13873	79655	0	552	13321	?
danoint	664	521	3232	0	56	465	65.66
disctom	399	10000	30000	0	10000	0	-5000
ds	656	67732	1024059	0	67732	0	93.52
fast0507	507	63009	409349	0	63009	0	174
fiber	363	1298	2944	0	1254	44	405935
fixnet6	478	878	1756	0	378	500	3983
gesa2-o	1248	1224	3672	336	384	504	25779900
gesa2	1392	1224	5064	168	240	816	25779900
glass4	396	322	1815	0	302	20	1200010000
harp2	112	2993	5840	0	2993	0	-73899800
liu	2178	1156	10626	0	1089	67	?
manna81	6480	3321	12960	3303	18	0	-13164
markshare1	6	62	312	0	50	12	1
markshare2	7	74	434	0	60	14	1
mas74	13	151	1706	0	150	1	11801.20
mas76	12	151	1640	0	150	1	40005.10
misc07	212	260	8619	0	259	1	2810
mkc	3411	5325	17038	0	5323	2	-563.84
mod011	4480	10958	22254	0	96	10862	-54558500
modglob	291	422	968	0	98	324	20740500
msc98-ip	15850	21143	92918	53	20237	853	19839500
mzzv11	9499	10240	134603	251	9989	0	-21718
mzzv42z	10460	11717	151261	235	11482	0	-20540
net12	14021	14115	80384	0	1603	12512	214
noswot	182	128	735	25	75	28	-41
nsrand-ipx	735	6621	223261	0	6620	1	51200
nw04	36	87482	636666	0	87482	0	16862
opt1217	64	769	1542	0	768	1	-16
p2756	755	2756	8937	0	2756	0	3124
pk1	45	86	915	0	55	31	11
pp08aCUTS	246	240	839	0	64	176	7350
pp08a	136	240	480	0	64	176	7350
protfold	2112	1835	23491	0	1835	0	-31
qiu	1192	840	3432	0	48	792	-132.87
roll3000	2295	1166	29386	492	246	428	12890
rout	291	556	2431	15	300	241	1077.56
set1ch	492	712	1412	0	240	472	54537.80
seymour	4944	1372	33549	0	1372	0	423
sp97ar	1761	14101	290968	0	14101	0	660706000
swath	884	6805	34965	0	6724	81	467.40
timtab1	171	397	829	94	64	239	764772
timtab2	294	675	1482	164	113	398	1096560
tr12-30	750	1080	2508	0	360	720	130596
vpm2	234	378	917	0	168	210	13.75

?: Unknown value

Table 1: Characteristics of the subset of MILP instances from MIPLIB 2003

Instance	objective FP niter	fobj	tt	stage	gap%	analytic center FP with PCx niter	fobj	tFP(tAC)	stage	gap%	AC value	analytic center FP with CPLEX niter	fobj	tFP(tAC)	stage	gap%	AC value
10teams	278	1014	19	3	9.73	179	1022	26(0)	3	10.59	1020	177	1056	25(0)	3	14.27	1020
a1c1s1	351	22714.68	8	2	97.45	0	46756.40	0(0)	0	306.43	50396.80	0	38193.60	0(0)	0	232	40504.70
aflow30a	41	2355	0	1	103.28	171	3802	1(0)	2	228.13	5377.34	298	5578	2(0)	2	381.36	4714.70
aflow40b	21	2329	1	1	99.32	394	8300	12(0)	3	610.09	7234.03	54	7051	2(0)	1	503.25	6635.63
air04	45	58229	181	1	3.73	186	72098.00	1220(2)	3	28.43	79260.30	186	71223.99	1147(0)	3	26.87	79989.5
air05	3	26930	2	1	2.11	186	37907	162(1)	3	43.73	45309	190	35798	148(0)	3	35.73	45732.10
arki001	803	7719381.38	15	3	1.83	871	7729296.21	43(0)	3	1.96	7807100	1573	7763720.15	79(0)	3	2.41	7822170
atlanta-ip	454	156.01	227	3	75.52	42	198.02	68(9398)	1	118.68	171.31	397	154.01	934(11)	3	70.32	159.76
cap6000	31	-2442163	0	1	0.38	0	-2442800	1(0)	0	0.35	-596562	0	-2442800	1(0)	0	0.35	-109362
dano3mip	70	763.97	361	1	?	205	1000	1892(17)	3	?	12849.20	252	1000	1947(4)	3	?	995.15
danoint	96	74	3	1	12.50	99	76	4(0)	1	15.50	434.456	230	85.50	9(0)	3	29.75	66.77
disctom	3	-5000	3	1	0	4	-5000	3(1)	1	0	-5000	4	-5000	4(0)	1	0	-5000
ds	446	5418.56	9495	3	5633.77	198	5418.56	1945(10)	3	5633.77	1053.93	0	5418.56	1(2)	0	5633.77	5418.56
fast0507	8	184	51	1	5.71	39	11884	131(4)	1	6691.43	8254.52	0	275	2(1)	0	57.71	122425
fiber	41	6481506.12	0	1	1496.68	41	6481510	0(0)	1	1496.68	19694200	15	3147830	0(0)	1	675.45	45602200
fixnet6	67	41304	0	1	936.77	18	38401	0(0)	1	863.91	60883.20	0	97271.70	0(0)	0	2341.58	101827
gesa2-o	33	36205441.29	1	2	40.44	20	71213100	0(0)	2	176.23	116914000	35	32635500	1(0)	2	26.59	166784000
gesa2	33	28181419.78	0	2	9.32	3	38472300	1(0)	2	49.23	124095000	47	40307000	1(0)	2	56.35	188208000
glass4	374	12700154400	1	3	958.34	254	10500117800	2(0)	3	775	142889000000	224	5000046800	1(0)	3	316.67	8862840000
harp2	138	-60669440	3	1	17.90	178	-40631391	3(0)	3	45.02	-50758200	59	-49759800	1(0)	1	32.67	-46262500
liu	119	3286	1	1	?	119	3036	4(5)	1	?	9218.57	121	5876	5(0)	1	?	959.02
manna81	52	-12940	2	2	1.70	0	-12948	0(6)	0	1.64	-7307.16	0	-12878	0(0)	0	2.17	0
markshare1	65	725	0	1	36200	65	603	0(0)	1	30100	30.48	0	7286	0(0)	0	364250	7286
markshare2	65	963	0	1	48100	66	925	0(0)	1	46200	36.10	0	10512	0(0)	0	525550	10512
mas74	109	16534.04	0	1	40.10	0	57195600000	0(0)	0	484618022.50	571956000000	0	50000000000	0(0)	0	423649728.01	1000000000000
mas76	106	46242.57	1	1	15.59	0	26804400000	0(0)	0	67000682.38	536000000000	0	50000000000	0(0)	0	124980840.41	1000000000000
misc07	188	3690	1	1	31.31	217	3935	3(0)	2	40.02	3601.66	219	3410	2(0)	2	21.34	4894.40
mkc	13	-288.96	0	1	48.67	13	-276.96	1(0)	1	50.79	-253.58	12	38.81	1(1)	1	106.69	-95.53
mod011	12	-45633967.33	1	1	16.36	23	-37482400	3(1)	1	31.30	-31430100	23	-35547800	3(0)	1	34.84	-36600000
modglob	60	22995521.33	0	1	10.87	60	21809700	1(0)	1	5.16	272286000	0	82243300	0(0)	0	296.53	142949000
msc98-ip	61	30502274.00	26	1	53.75	33	30196300	16(949)	1	52.20	29571000	29	30928000	19(22)	1	55.89	29545100
mzzv11	540	-17898	127	3	17.59	567	-16262	435(116)	3	25.12	-4264.40	561	-13744	484(7)	3	36.71	-4794.93
mzzv42z	25	-14502	49	1	29.39	23	-12736	13(147)	1	37.99	-3210.77	27	-14192	12(15)	1	30.90	-3825.70
net12	216	337	12	2	57.21	25	337	10(86)	1	57.21	325.12	25	337	8(27)	1	57.21	337
noswot	13	-41	1	2	0	34	-15	0(0)	2	61.90	-21.82	33	-31	0(0)	2	23.81	-15.67
nsrand-ipx	132	211040	5	2	312.38	883	258080	367(2)	3	404.05	761986	694	203040	265(0)	3	296.56	802647
nw04	10	17858	10	1	5.91	2	18380	9(8)	1	9	50318.90	42	61640	120(2)	1	265.54	52460.90
opt1217	40	-16	0	1	0	124	-12.11	0(0)	1	22.80	-8.23	0	0	0(0)	0	94.12	0
p2756	377	51338	2	3	1542.85	244	51338	7(0)	3	1542.85	139225	279	51338	7(0)	3	1542.85	164724
pk1	56	36	0	1	208.33	57	86	0(0)	1	625	34.13	0	731	0(0)	0	6000	731
pp08aCUTS	10	8360	0	1	13.74	11	16390	0(0)	1	122.98	18715	0	21671.40	0(0)	0	194.82	23012.40
pp08a	11	12010	0	1	63.39	15	15850	0(0)	1	115.63	21666.70	0	18439.30	0(0)	0	150.85	18778.70
protfold	286	-16	90	2	46.88							307	-18.90	365(2)	3	37.81	-18.42
qiu	9	160.76	0	1	219.34	41	868.57	1(0)	1	748.05	722.04	0	3693.35	0(0)	0	2858.10	4188.61
roll3000	793	36109.80	17	3	180.12	818	40048.40	65(1)	3	210.68	44336.80	175	18507	11(1)	2	43.57	38004.10
rout	117	1652.55	0	1	53.31	79	1644.41	1(0)	1	52.56	1455.68	74	1337.27	1(0)	1	24.08	1474.90
set1ch	46	95845.5	0	1	75.74	0	268719	0(0)	0	392.71	224714	0	216475	0(0)	0	296.92	262834
seymour	7	471	3	1	11.32	39	754	5(35)	1	78.07	728.54	0	588	0(0)	0	38.92	1345
sp97ar	9	919778417.68	4	1	39.21	63	1161990000	57(4)	1	75.87	8272000000	97	11702100000	88(1)	1	1671.15	18441700000
swath	395	35951.85	14	2	7575.56	795	34774.58	96(1)	3	7324.22	1470.14	795	34774.58	100(0)	3	7324.22	1470.14
timtab1	216	1400493.99	1	2	83.13	169	1081000	1(0)	2	41.35	1475570	819	1401240.99	3(0)	3	83.22	419539
timtab2	1222	1982037.99	2	2	80.75	972	2105005.99	6(0)	3	91.96	2052380	1072	1772242.99	7(0)	3	61.62	671850
tr12-30	25	164128	1	1	25.68	214	289227.99	7(0)	3	121.47	135860	221	285716	6(0)	3	118.78	75936.50
vpm2	12	18.25	0	1	30.51	11	29.50	0(0)	1	106.78	48.48	27	23.75	0(0)	1	67.8	14.08

?: Unknown value

Table 2: Computational results using the objective and the analytic center FP

the feasible and the optimal solution ("gap%"), and FP stage where the feasible point was found ("stage"). For the objective FP, the column "tt" shows the total CPU time. For the two analytic center FP variants, columns "tFP(tAC))" report separately the CPU time spent in stages 1 to 3 ("tFP") and the time for computing the analytic center at stage 0 (in brackets, "tAC"); the total time is the sum of the two values. For the two analytic center FP variants columns "AC value" show the value of the original objective function evaluated at the analytic center. Differences are due to different computed analytic centers because both solvers apply very distinct preprocessing strategies. The default FP settings were used as suggested in [1]. All runs were carried on a Dell PowerEdge 6950 server with four dual core AMD Opteron 8222 3.0 GHZ processors (without exploitation of parallelism capabilities) and 64 GB of RAM.

Although from Table 2, in general it can be concluded that the analytic center FP is inferior to the objective FP, there are some notable exceptions. For instance, for the 13 instances with both binary and general integer variables, the analytic center FP (either with PCx or CPLEX) obtained a solution with a lower gap than the objective FP in eight of the 13 instances; in some cases more efficiently and even being able to find a solution when the objective FP failed (i.e., it required stage 3), as for instances "roll3000" and "atlanta-ip" (in this latter case, however, at the expense of a very large CPU time). On the other hand, for problems with only binary variables the analytic center FP obtained solutions with a lower gap in very few instances. A possible explanation of this different behaviour in problems with and without general integer variables is that, for a binary problem, the only feasible integer points "close" to the segment $x(\gamma)$ are $\{0, 1\}^n$, which in addition may be far from the center. For problems with general integer variables, the number of feasible integer solutions close to the analytic center will be, in general, much larger. For some problems with only integer binary variables, the analytic center FP behaved very poorly, as for "mas74" and "mas76" (it stopped at stage 0 in those cases). However, in other instances it was much more efficient obtaining the same gap that the objective FP, as for "ds". Note that for "ds" the analytic center FP with CPLEX obtained the feasible solution in one second at stage 0 (the other two variants failed, requiring stage 3). However, in that case CPLEX did not really compute the analytic center: it solved $\min_x\{0 : x \in P\}$ heuristically, instead of applying the barrier algorithm, as required. It thus considered a segment between two feasible solutions, none of them being the analytic center of P. Therefore, the idea of using a segment of feasible points is not restricted to the case where one of the endpoints is the analytic center, and it can be extended to more general situations.

4. Conclusions

The analytic center FP is an extension of the original FP where candidate points to be rounded are found in a segment of feasible points, one of the extremes being the analytic center. The objective FP is a particular case where the endpoint associated to the solution of the relaxed problem is selected as the point to be rounded. The analytic center FP has not been shown to outperform the objective FP, in general. However for problems with both general integer and binary variables, and for some particular binary problems, it may result in more efficient and lower gap solutions. The analytic center FP could also be used with the recent rounding scheme based on constraint propagation suggested in [8].

Acknowledgments

This work has been supported by grants MTM2009-08747 of the Spanish Ministry of Science and Innovation, and SGR-2009-1122 of the Government of Catalonia.

References

[1] T. Achterberg, T. Berthold, Improving the feasibility pump, Discrete Optimization 4 (2007), 77–86.
[2] T. Achterberg, G. Gamrath, T. Koch, A. Martin, The mixed integer programming library: MIPLIB 2003. http://miplib.zib.de .
[3] H.Y. Benson, Mixed integer nonlinear programming using interior-point methods, Optimization Methods and Software (2010), doi: 10.1080/10556781003799303.
[4] P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuejols, I.E. Grossman, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, A.Wachter, An algorithmic framework for convex mixed integer nonlinear programs, Discrete Optimization 5 (2008), 186–204.
[5] L. Bertacco, M. Fischetti, A. Lodi, A feasibility pump heuristic for general mixed-integer problems, Discrete Optimization 4 (2007), 63–76.
[6] J.Czyzyk, S. Mehrotra, M. Wagner, S.J. Wright, PCx: an interior-point code for linear programming, Optimization Methods and Software 11 (1999) 397–430.
[7] M. Fischetti, F. Glover, A. Lodi, The Feasibility Pump, Mathematical Programming 104 (2005), 91–104.
[8] M.Fischetti, D. Salvagnin, Feasibility pump 2.0, Mathematical Programming Computation 1 (2009), 201–222.
[9] GNU, GNU Linear Programming Kit v. 4.43, 2010.
[10] IBM ILOG CPLEX 12.1, User's Manual, 2010.
[11] J.E. Mitchell, Fixing variables and generating classical cutting planes when using an interior point branch and cut method to solve integer programming problems, European Journal of Operational Research 97 (1997), 139–148.
[12] J.E. Mitchell, M.J. Todd, Solving combinatorial optimization problems using Karmarkar's algorithm, Mathematical Programming 56 (1992), 245–284.
[13] Y. Ye, Interior Point Algorithms. Theory and Analysis, Wiley, 1997.