

A New Approach to the Feasibility Pump in Mixed Integer Programming*

N. L. Boland ^{‡¶} A. C. Eberhard ^{†§} F. Engineer ^{‡**} A. Tsoukalas, ^{†||}

26th December, 2011

Abstract

The *feasibility pump* is a recent, highly successful heuristic for general mixed integer linear programming problems. We show that the feasibility pump heuristic can be interpreted as a discrete version of the proximal point algorithm. In doing so, we extend and generalize some of the fundamental results in this area to provide new supporting theory. We show that feasibility pump algorithms implicitly minimize a weighted combination of the objective and a term which penalizes lack of integrality. This function has many local minima, some of which correspond to feasible integral solutions; the feasibility pump's use of random restarts can be viewed as seeking to escape these local minima when they are not feasible integral solutions. This interpretation suggests alternative ways of incorporating restarts, one of which is the application of cutting planes. Numerical experiments with cutting planes show encouraging results on standard test libraries.

1 Introduction

In spite of the continuous improvement of both commercial and open source solvers, numerous *Mixed Integer Programming* (MIP) problems of practical importance remain intractable. In practice, where rigorous algorithms fail, heuristics often succeed to provide feasible solutions of good quality. Apart from their self evident value, good feasible solutions are also useful in speeding up the search of branch and cut algorithms. General purpose heuristics include [2, 3, 4, 13, 14, 15, 16, 19, 20, 21, 22, 23]. We direct the reader to [6] for a recent survey.

A heuristic that has attracted significant interest in recent years is the *Feasibility Pump*(FP) [17]. The FP starts from the *Linear Program* (LP) optimum and computes two trajectories of points, one integer and the other LP feasible, by iteratively applying rounding and projection operations. Every LP-feasible point is rounded to an integer point and every integer point is projected back onto the LP feasible region using the l_1 norm. As both the rounding and the projection are minimal distance operations (with respect to the l_1 norm), the distance between pairs of points on the two trajectories decreases monotonically until the method cycles, unable to further decrease the distance. If the distance is reduced to zero before cycling then a feasible point has been obtained. Otherwise a random move is used to restart the method from a new point. In the original FP [17], the only bias of the method towards points of good objective value is the starting point, so the random restarts threaten to escape the good quality area. Thus, the feasibility pump uses a sophisticated restart scheme, attempting to displace the current point in an economic way by “minor perturbations”, and performs a “major restart” only in the presence of further failures.

The FP method demonstrates encouraging performance on a large class of test problems, but behaves better for binary problems than for general integer variables. Thus, [5] first deal with the binary variables in isolation, treating general integers as continuous variables. General integer variables are

*This research was supported by the ARC Discovery grant no. DP0987445

[‡]Faculty of Science and Information Technology, School of Mathematical and Physical Sciences, University of Newcastle, University Drive, Callaghan NSW 2308 Australia

[†]School of Mathematical and Geospatial Sciences, RMIT, GPO Box 2476V, Melbourne, Victoria, Australia 3001.

[¶]Natashia.Boland@newcastle.edu.au

[§]andy.eb@rmit.edu.au

^{**}Faramroze.Engineer@newcastle.edu.au

^{||}angelos.tsoukalas@rmit.edu.au

treated as integer in a subsequent second round only after a point satisfying integrality for all binary variables is obtained. In [1] the *Objective Feasibility Pump* (OFP) is proposed, where the projection problem is modified to include an objective function term, while in [18] simple rounding is substituted by a sophisticated constraint propagation approach. The latter makes use of information about the LP feasible region, with rounding done one variable at a time, while exploiting constraint information to disqualify unacceptable values. A modified penalty function is used in [27] that implicitly promotes a bias towards fixing more variables at integral values during the projection step.

The numerical results presented in [1] indicate that adding an objective term to the projection problem is beneficial to the quality of the solution, while not compromising computation time too much. Constraint propagation [18] is shown to further improve solution quality while also decreasing the failure rate of the method. The modified penalty function of [27] yields an improvement in efficiency and number of iterations relative to the original FP [17]. Recently the FP has been successfully extended to nonlinear problems, see for example [8, 10, 11], which focus on the challenge introduced by nonlinear (and nonconvex) constraints.

In this paper, we explore the relationship between generalized forms of the FP and proximal point algorithms, in the context of linearly constraint (possibly nonlinear) integer programs (see, for example, [9, 26]). In doing so, we provide insights about the properties of the points generated by the FP method, and of its mathematical behaviour. Aside from the OFP, methods that improve the original FP can be characterized as either putting some of the LP into the rounding step (e.g. constraint propagation in [18]), or putting some integrality into the projection step ([27]). Here we also propose to explore the latter, but in a different way: by incorporating integer cutting planes into the LP projection step.

We consider the general nonlinear MIP

$$\begin{aligned} \min \quad & g(x) \\ \text{s.t.} \quad & Ax \geq b, \quad x_{\mathcal{I}} \in \mathbf{I}, \end{aligned} \tag{1}$$

where $g : \mathbf{R}^n \rightarrow \mathbf{R}$ is a possibly nonlinear continuous function, \mathcal{I} is the index set of integer variables, and $\mathbf{I} = \mathbf{Z}^m$, where $m = |\mathcal{I}|$. The remaining variables, denoted by $x_{\mathcal{R}}$, are allowed to vary over \mathbf{R}^{n-m} ; the nonlinear MIP feasible set is $F := \{x \in \mathbf{R}^n \mid Ax \geq b, x_{\mathcal{I}} \in \mathbf{I}\}$. We will refer to *integral points* as being those points $x \in \mathbf{R}^n$ for which $x_{\mathcal{I}} \in \mathbf{I}$. Although the objective may be nonlinear, throughout we will refer to the feasible set with integrality relaxed as the *LP feasible set*, defined as $F_{LP} := \{x \in \mathbf{R}^n \mid Ax \geq b\}$.

In this paper we will show that the FP heuristic can be interpreted as a kind of proximal point method, by comparing the FP with the primal-proximal heuristic introduced in [12]. Our analysis focusses on the “inner loop” of the FP method, during which the two trajectories of rounded and projected points are calculated, before cycling necessitates a perturbation or restart. We call this the *pumping process*, and the points generated by rounding during this process the *discrete proximal point iterates*. Each of these points is associated with the LP feasible point it was rounded from: we call these the *LP-feasible iterates*. The key departure of the FP method from standard proximal point methods is the rounding step, which in effect is a projection onto the integer points, giving a point minimizing $I(x) := \min \{\|x_{\mathcal{I}} - y_{\mathcal{I}}\|_1 \mid y_{\mathcal{I}} \in \mathbf{I}\}$. Thus the rounding step implicitly calculates the measure of lack of integrality given by $I(x)$, where x is an LP-feasible iterate. One of the important results of our analysis in this paper is that LP-feasible iterates generated during the pumping process converge to a local minimum of the function $g + rI$ on F_{LP} , where $r > 0$ is a parameter that weights the importance of the two conflicting objectives. Thus local minimization of the objective $g + rI$ over F_{LP} is associated with cycling events in the FP method.

In [24] and [27] pure *Integer Programs* (IP) were studied and it was noted that the *global* minimizers of various penalized continuous problems over F_{LP} coincide with the global optimal solutions of the original IP. In this paper we will show that a similar interpretation of the original FP [17] and the Objective Feasibility Pump (OFP) [1] can be made for a *Mixed IP* (MIP). Thus if the pumping process of an FP method can be started in the right region, (and with the right value for the parameter r), it will converge to an optimal solution of the MIP. Interestingly, FP methods do contain a mechanism for re-starting the pumping process.

When the FP method detects cycling it performs either a “major” or “minor” restart, both using some form of randomization. These restarts form the *outer loop* of iterates of the FP. We provide numerical results which show that the repeated use of major restarts in particular correlates well with cases in which poorer solution quality is obtained by the FP method. Our analysis explains that

cycling occurs because local optimality of the function $g + rI$ over F_{LP} has been obtained, and hence suggests some alternative ways to overcome this problem when this local minimum is not integral. Two approaches to escaping these undesirable local optimum present themselves: effect a change on either the objective function or the feasible region F_{LP} . The former can be achieved by varying the weighting parameter $r > 0$; we plan to consider this approach in future work. In this paper, we focus on the latter, and propose to alter the feasible region via the introduction of cuts to remove the local, non-integral minimum at which the cycling has occurred. We give encouraging numerical experiments which demonstrate the potential for this cutting strategy. We show the strategy tends to delay the use of major restarts, and leads to improvements in the quality of the feasible integral solutions found.

The paper is structured as follows. In Section 2 we reinterpret the inner iterates of the FP as a kind of a proximal point iteration and discuss the connection this interpretation has to the primal-proximal heuristic introduced in [12]. As we require more general versions of some results of [12, sections 1.3 and 2] we provide extensions of these results for our nonlinear MIP in Section 3. In particular, we show that quite general coercive penalties can be used in place of the l_2 norm which was used in [12] in order to exploit Lagrangian relaxation techniques. As the FP exploits LP relaxations our analysis departs from that of [12] significantly at this point and in Section 4 we complete the development begun in Section 2 by showing how the minimization of $g + rI$ over F_{LP} results from such proximal point iterates (the ‘‘pumping process’’). In Section 5 cycling is shown to be associated with the failure to locate an integral LP feasible point at a local minimum of $g + rI$. A modification of the OFP algorithm of [18] is proposed in Section 6. Here extensive numerical testing is undertaken in which we selectively introduce cuts to prevent cycling instead of the sole use of minor perturbations.

2 FP, Proximal Points and Concave Penalties

In this section we draw out some comparisons and contrasts with recent approaches that convert integer programs to nonlinear continuous optimization problems. Consider the general nonlinear MIP (1). Proximal point approaches [12] consider regularising this problem using a penalty $\rho(x)$ to obtain a continuous optimization problem

$$\min_y \varphi_r(y) \quad \text{where} \quad (2)$$

$$\varphi_r(y) := \min \{g(x) + r\rho(x - y) \mid x \in F\}. \quad (3)$$

Regularization is a common tool used in nonlinear and nonsmooth optimization. In a sense, MIPs are highly nonsmooth, as their domains are disconnected. Regularization in the form of (3) can be seen as a mathematical tool to construct, in a systematic way, an equivalent continuous problems, with smoother functions. The results can have attractive properties, as we will see in Section 3 and Section 4.

The analysis of the problem (2) when ρ is the Euclidean l_2 norm was carried out in some detail in [12]. The main property that arises is that integrality of local minima of (2) is assured and do often correspond to the feasible solutions of (1). Indeed y_0 (where $(y_0)_{\mathcal{I}} \in \mathbf{I}$) is a local minimum of φ_r exactly when $x = y_0$ is the solution to the minimization problem $\varphi_r(y_0)$.

Since the calculation of the objective φ_r is as difficult as the original MIP, in [12] Lagrangian relaxation is used to obtain a solvable relaxed problem. This gives an underestimate $\tilde{\varphi}_r$ of φ_r . Unfortunately the local minima of the relaxation $\tilde{\varphi}_r$ are not assured to be integral. In Section 3 of [12], approaches to varying the parameter $r > 0$ to obtain feasible integral points are considered, but the results are inconclusive and the tone of [12] is quite negative in this respect. In the final section of [12] binary problems are addressed and a penalty used to enforce integrality, but no role for rounding is considered.

The OFP, by contrast, drops the integrality constraint in (3) and takes ρ to be an l_1 norm, so that a relaxation of φ_r when $g(x) = c^T x$ can be calculated as a linear program. Thus we arrive at the problem

$$\min_y \varphi_r^{LP}(y) \quad \text{where} \quad (4)$$

$$\varphi_r^{LP}(y) := \min \{g(x) + r \|x - y\|_1 \mid x \in F_{LP}\}. \quad (5)$$

Proximal point algorithms seek local optima by a kind of fixed point iteration, where the *center* y is replaced at each step by the optimal solution of the regularization problem (e.g. of (5)). FP methods also take this approach, but to try and encourage integrality of a solution, a new center at y_0 with

$(y_0)_I \in \mathbf{I}$ is sought at each iterate. If $x = y_0$ is the solution to the minimization problem, the FP method is clearly finished, having obtained a feasible integral point. Otherwise, the non-integral solution of (5) is rounded to the nearest integral point in order to provide the new center for the next iteration.

Since FP methods use the l_1 norm, the analysis of [12] does not directly apply. Thus in order to establish similar properties, we in Section 3 extend these results, and in doing so, prove they hold for general class of coercive functions ρ , including the l_1 norm as a special case. This allows us to show that regularisation preserves the global minimizers (which are the optimal solutions of the MIP) and retains the same local minimizers (which correspond to the feasible integrals solutions of the MIP). These correspondences are established in Section 3 for our general nonlinear MIP.

In Section 5 we show that the pumping process applied in an OFP setting finds local optima of $g+rI$ over F_{LP} and so the function I acts as a penalty for the deviation from integrality. The original FP is essentially the limiting version of the scheme for $r \rightarrow \infty$. Thus the pumping phase of the FP implicitly performs a minimization of a concave nonlinear integrality penalty via a combination of projective LPs and roundings. Our results provide some additional insights. For example, if the OFP finishes early without major restarts, it appears likely that the resulting solution will be both an optimal for the original MIP and a global minimum of $g + rI$.

This approach is distinct from that proposed in [24] where piecewise *concave* penalties are used to obtain an exact penalization for IPs and is motivated by the long observed connection that binary programs have to concave minimization. This is exploited in [27] to provide an interpretation of the FP for the case of binary problems. Motivated by the observation of [14] they note that the original FP [17] for binary problems can be interpreted as a Frank-Wolfe method [7, Section 2.2.2] as applied to a non-differentiable concave function over the feasible set F_{LP} . They develop theory that relates the *global* minima of these continuous penalized problems to the solution of an IP. We are able to provide similar results in Section 5 thus showing a connection of our approach to that used in [24].

3 General Properties of Regularizations

In this section we generalise key results of [12]. The generalisation takes two forms: we use a general class of coercive penalty ρ instead of $\rho = \|\cdot\|_2$, and give results for *Mixed* rather than pure IPs. This yields results for $\rho = \|\cdot\|_1$ that apply to the FP heuristic for MIPs. In particular, we show that the regularization (2) has global and local minimizers closely related to the optimal and feasible points of the original MIP. We posit the following assumptions.

Assume: A *continuous* function $\rho : \mathbf{R}^n \rightarrow \mathbf{R}$ that satisfies the following conditions is called an *integer compatible regularization function* (ICRF) iff:

Condition 1 $\rho(x) \geq 0$ for all x and $x = 0$ iff $\rho(x) = 0$.

Condition 2 If $\gamma \in (0, 1)$ then $\rho(\gamma x) < \rho(x)$ for all $x \neq 0$.

Condition 3 There exists a continuous, strictly increasing, function $s(\cdot) : \mathbf{R}_+ \rightarrow \mathbf{R}_+$ and a \bar{K} such that for all $K < \bar{K}$ we have $\rho(x) \leq K$ implies $\|x\|_1 \leq s^{-1}(K)$ for the l_1 norm $\|\cdot\|_1$ on \mathbf{R}^n .

Remark 1 *These properties are deliberately chosen to be general enough to allow all of the theory to cover the case of any norm on \mathbf{R}^n and for most of the theory to encompass the cases of some concave penalties already introduced in the literature for binary problems. Recall that in finite dimensions all norms are equivalent to the l_1 norm i.e. for all norms $\|\cdot\|$ there exists constants $A, B > 0$ such that $A\|\cdot\|_1 \geq \|\cdot\| \geq B\|\cdot\|_1$. With this in mind it is clear how these assumptions relate to equivalent norms.*

In [27] and [24] penalty functions were generated as follows:

$$\rho(x) := \sum_i p(|x_i|) \tag{6}$$

where $p : \mathbf{R}_+ \rightarrow \mathbf{R}_+$ is a concave, strictly increasing function. Some examples taken from [27] that were

used explicitly for binary problems are:

$$p(t) = \log(t + \alpha) - \log \alpha, \quad (7)$$

$$p(t) = -[t + \alpha]^{-q} + \alpha^{-q}, \quad (8)$$

$$p(t) = 1 - \exp(-\alpha t), \quad (9)$$

$$p(t) = [1 + \exp(-\alpha t)]^{-1} - \frac{1}{2}, \quad (10)$$

for some $\alpha, q > 0$. We wish to include these functions in our analysis in order to make some comparisons with the approaches in [27] and [24]. It is interesting to note that the case when $p(x) = x$ corresponds to p being both simultaneously concave and convex and to $\rho(\cdot) = \|\cdot\|_1$.

The following justifies the use of the conditions 1 to 3 above for penalties of the form (6).

Proposition 2 *Suppose ρ is defined via (6) with $p(0) = 0$, $p(t) > 0$ for $t \neq 0$ and p concave, lower semi-continuous and strictly increasing function. Then ρ is an integer compatible regularization function. In particular when the $\text{Range } p = [0, +\infty)$ we may take $\bar{K} = +\infty$.*

Proof. Suppose

$$\rho(x) = \sum_i p(|x_i|) \leq K$$

then for each i we have $|x_i| \leq p^{-1}(K)$ when $K \in \text{Range } p$. As $\text{Range } p$ contains an interval of the type $[0, b)$ we may take $0 < K < \bar{K} := b$. In particular when $b = +\infty$ we can take $\bar{K} = +\infty$. Consequently

$$\|x\|_1 \leq \sqrt{n} \|x\|_\infty \leq \sqrt{n} p^{-1}(K)$$

and hence ρ satisfies Condition 3 with $s(\cdot) := \frac{1}{\sqrt{n}} p(\cdot)$. Clearly $\rho \geq 0$ and $\rho(x) = 0$ iff $\|x\|_1 = 0$. As s is strictly increasing $\rho(\gamma x) = \sum_i p(\gamma |x_i|) < \sum_i p(|x_i|) = \rho(x)$. ■

Let

$$f(x) := \begin{cases} g(x) & \text{if } Ax \geq b, x_{\mathcal{I}} \in \mathbf{I} \\ +\infty & \text{otherwise} \end{cases},$$

where $g : \mathbf{R}^m \rightarrow \mathbf{R}$ is a continuous function. Then

$$\begin{aligned} \varphi_r(y) &= \inf \{g(x) + r\rho(x - y) \mid Ax \geq b, x_{\mathcal{I}} \in \mathbf{I}\} \\ &= \inf \{f(x) + r\rho(x - y) \mid x \in \mathbf{R}^n\}. \end{aligned}$$

Define

$$f_r(x, y_0) := f(x) + r\rho(x - y_0) \quad \text{for all } x \in \mathbf{R}^n$$

and so $\varphi_r(y_0) = \inf \{f_r(x, y_0) \mid x \in \mathbf{R}^n\}$. Note that as g is continuous and the feasible region $F := \{x \mid Ax \geq b, x_{\mathcal{I}} \in \mathbf{I}\}$ is a closed set it follows that $f : \mathbf{R}^m \rightarrow \bar{\mathbf{R}}$ is a lower semi-continuous function. We also observe that as φ_r is an infimum of a family of continuous functions $y \mapsto f(x) + r\rho(x - y)$ it is, at worst, an upper semi-continuous function of the variable y . We can claim more under mild conditions. Denote the epigraph $\text{epi } f := \{(x, \alpha) \mid f(x) \leq \alpha\}$ and note that $\text{epi } h \supseteq \text{epi } f + \text{epi } r\rho$ if and only if $h(y) \leq f(x) + r\rho(x - y)$ for all $x, y \in \mathbf{R}^n$. Thus geometrically $\text{epi } \varphi_r$ is the largest extended real valued function whose epigraph contains the sum $\text{epi } f + \text{epi } r\rho$.

Theorem 3 *Suppose $g : \mathbf{R}^n \rightarrow \mathbf{R}$ is continuous and the feasible region F is closed and nonempty. Suppose in addition that the regularized function $\varphi_{\bar{r}}(y_0) > -\infty$ for some y_0 and that ρ is a ICRF that is also Lipschitz continuous (with a global Lipschitz constant). Then φ_r is a finite valued for $r \geq \bar{r}$ and globally Lipschitz continuous.*

Proof. Clearly $\varphi_r(y) \geq \varphi_{\bar{r}}(y)$ for all y and $r \geq \bar{r}$. Thus when $\varphi_{\bar{r}}(y) > -\infty$ for all y then $\varphi_r(y) > -\infty$ for all $r \geq \bar{r}$. Moreover as $\varphi_r(y) \leq g(x) + r\rho(x - y) < +\infty$ for any $x \in F$ finiteness of φ_r follows. Thus if we can show $\varphi_r(y)$ is bounded away from negative infinity for all y then we would have shown it to be finite valued.

Let C_μ denote the cone $\text{epi } \mu\|\cdot\|$. Note next that the Lipschitz continuity property of ρ corresponds to the existence of a Lipschitz constant $\mu > 0$ such that $\text{epi } \rho + C_\mu \subseteq \text{epi } \rho$. Now $\text{epi } \rho + C_\mu \subseteq \text{epi } \rho$ in turn implies $\text{epi } r\rho + C_{r\mu} \subseteq \text{epi } r\rho$ with $\text{int } C_{r\mu} \neq \emptyset$ for all r .

Suppose $\varphi_r(y) = -\infty$ for some y then

$$(y, -n) \in \text{epi } f + \text{epi } r\rho$$

for all $n \in \mathbf{Z}^+$ and so (using $\text{int } C_{r\mu} \neq \emptyset$)

$$\begin{aligned} \mathbf{R}^{n+1} &\subseteq \bigcup_{n=1}^{\infty} ((y, -n) + C_{r\mu}) \subseteq (\text{epi } f + \text{epi } r\rho) + C_{r\mu} \\ &\subseteq \text{epi } f + (\text{epi } r\rho + C_{r\mu}) \subseteq \text{epi } f + \text{epi } r\rho \subseteq \text{epi } \varphi_r \end{aligned}$$

implying $\varphi_r \equiv -\infty$, contradicting $\varphi_r(y_0)$ finite. Let $\varphi_r(y) < \alpha$ and take x such that $f(x) + r\rho(x-y) < \alpha$. Let $z \in \mathbf{R}^n$ then

$$\begin{aligned} \varphi_r(z) &\leq f(x) + r\rho(x-z) \leq (f(x) + r\rho(x-y)) + r(\rho(x-z) - \rho(x-y)) \\ &< \alpha + r(\rho(x-z) - \rho(x-y)) \leq \alpha + r\mu\|z-y\|, \end{aligned}$$

where we have used the Lipschitz continuity of ρ again. As this holds for all $\alpha > \varphi_r(y)$ we have $\varphi_r(z) \leq \varphi_r(y) + r\mu\|z-y\|$. As $y, z \in \mathbf{R}^n$ are arbitrary we are finished.

■

Remark 4 *The conditions of Theorem 3 are satisfied for any norm and all the concave penalties formed via (6) using the various choices for the function p .*

The following results are a generalization of the results of [12] Section 1 and these form a basic tool in the proof of subsequent properties. The most important property is Property 3 which roughly says that local minima of φ_r are not only integral but correspond to “fixed points” of the mapping $y_0 \mapsto \arg \min \{f_r(x, y_0) \mid x \in \mathbf{R}^n\}$.

Lemma 5 *Posit the assumption that ρ is an ICRF.*

1. For all $y \in \mathbf{R}^n$ the function $r \mapsto \varphi_r(y)$ is nondecreasing.
2. $\varphi_r(y) \leq f(y)$ for all $y \in \mathbf{R}^n$ for all $r \geq 0$.
3. Assume f is bounded from below. Then there exists $\bar{r} > 0$ such that if y_0 is a local minimum of φ_r for some $\bar{r} < r$ then $x = y_0$ is the unique minimum point of $f_r(\cdot, y_0)$ in

$$\varphi_r(y_0) = \inf \{f_r(x, y_0) \mid x \in \mathbf{R}^n\}. \quad (11)$$

In particular $\varphi_r(y_0) = f(y_0)$ and y_0 is integral. Moreover if $\bar{K} = +\infty$ in the definition of ICRF then $\bar{r} = 0$.

4. Local minima of φ_r are also local minima of $\varphi_{r'}$ for $r' \geq r$.

Proof. We show part 3 only, as it is the critical observation; the other observations follow in a similar way to the results of [12] Section 1. As f is lower semi-continuous and bounded from below and Condition 3 of the definition of a integer compatible regularization function holds, implying the existence of function s , we have

$$\begin{aligned} &\{x \mid g(x) + r\rho(x-y_0) \leq K, Ax \geq a, x_{\mathcal{I}} \in \mathbf{I}\} \\ &\subseteq \left\{x \mid \min_x f(x) + r\rho(x-y_0) \leq K\right\} \subseteq \left\{x \mid \|x-y_0\|_1 \leq s^{-1} \left(\frac{K - \min_x f(x)}{r}\right)\right\} \end{aligned} \quad (12)$$

which is bounded and nonempty for any $K > \min_x f(x)$ and $r > 0$ sufficiently large so that

$$\left(\frac{K - \min_x f(x)}{r}\right) < \bar{K}.$$

Let y_0 be a local minimum of φ_r , so there exists an integral point $x_0 \in F$ and a neighbourhood of y_0 denoted by $B_\varepsilon(y_0)$ (for some $\varepsilon > 0$) such that

$$f(x_0) + r\rho(x_0 - y_0) = \varphi_r(y_0) \leq \varphi_r(y) \leq f(x_0) + r\rho(x_0 - y)$$

for any $y \in B_\varepsilon(y_0)$. Hence $\rho(x_0 - y_0) \leq \rho(x_0 - y)$ for any $y \in B_\varepsilon(y_0)$. Assume that $\rho(x_0 - y_0) > 0$. Take $y = y_0 + t(x_0 - y_0)$ and so if $x_0 - y_0 \neq 0$ we have for t sufficiently small

$$\rho(x_0 - y_0) \leq \rho(x_0 - y) = \rho((1-t)(x_0 - y_0)) < \rho(x_0 - y_0)$$

which is a contradiction when $0 < t < 1$ under Condition 3 if $\rho(x_0 - y_0) > 0$. Thus $\rho(x_0 - y_0) = 0$ and so $x_0 = y_0$ and $(y_0)_\mathcal{I} \in \mathbf{I}$ with $\varphi_r(y_0) = f_r(y_0, y_0) = f(y_0)$. ■

The relationship between $\inf_{y \in \mathbf{R}^n} \varphi_r(y)$ and $\inf \{g(x) \mid Ax \geq b, x_\mathcal{I} \in \mathbf{I}\}$ is developed in the next result which is essentially Theorem 1.5 of [12] and provided for completeness. This result demonstrates that there is a direct correspondence between the local and global minimizers of the nonlinear MIP and the unconstrained continuous regularized problem φ_r .

Theorem 6 *Posit the assumption that ρ is an ICRF. Then the minimization of f and φ_r are related as follows:*

1. $\inf \{g(x) \mid Ax \geq b, x_\mathcal{I} \in \mathbf{I}\} = \inf \{\varphi_r(y) \mid y \in \mathbf{R}^n\}$,
2. if $x^* \in \arg \min \{f(y) \mid y \in \mathbf{R}^n\} = \arg \min \{g(x) \mid Ax \geq b, x_\mathcal{I} \in \mathbf{I}\}$, then $x^* \in \arg \min \{\varphi_r(y) \mid y \in \mathbf{R}^n\}$ and
3. Suppose \bar{r} is as specified in Lemma 5 part 3 and f is bounded below. Then for $r > \bar{r}$ we have y^* minimizes (resp. locally minimizes) φ_r then y^* minimizes (resp. locally minimizes) f .

Proof. Consider the first part. For any x and y in \mathbf{R}^n we have $f_r(x, y) \geq f(x)$ and hence $\varphi_r(y) \geq \inf f$ and so $\inf \varphi_r \geq \inf f$. On the other hand Lemma 5 part 2 gives $\inf \varphi_r \leq \inf f$ and equality. Suppose $x^* \in \arg \min \{f(x) \mid x \in \mathbf{R}^n\}$. Then using Lemma 5 part 2 again we have $\inf \varphi_r = \inf f = f(x^*) \geq \varphi_r(x^*)$ and so $x^* \in \arg \min \{\varphi_r(y) \mid y \in \mathbf{R}^n\}$. Finally, posit the assumption of Lemma 5 part 3 and suppose y^* minimizes (resp. locally minimizes) φ_r then $f(y^*) = \varphi_r(y^*) \leq \varphi_r(y) \leq f(y)$ for all y (y close to y_0). ■

The converse of Lemma 5 part 3 may be developed and this result establishes when the local minimizers of the regularized problem are actually strict local minimizers. As the results of this paper are in the main based only on the implication given in Lemma 5 part 3 we relegate the proof of the next result to an appendix.

Proposition 7 *Suppose g is bounded below on the feasible region F , ρ is an ICRF and $r > \bar{r}$ (as specified in Lemma 5 part 3). A point y_0 is a local minimum of φ_r if and only if $x = y_0$ with $x_\mathcal{I} \in \mathbf{I}$ is a unique optimal solution of*

$$\varphi_r(y_0) = \min \{g(x) + r\rho(x - y_0) \mid Ax \geq b, x_\mathcal{I} \in \mathbf{I}\}. \quad (13)$$

Indeed if y_0 is a local minima of φ_r then it is a strict local minimum of φ_r in the following cases.

1. We have an IP (i.e. $m = n$).
2. $(y_0)_\mathcal{R}$ is a strict local minimum of the continuous partial regularization

$$G_r(y_\mathcal{R}) := \min_{z_\mathcal{R} \in \mathbf{R}^{n-m}} \{G_{y_0}(z_\mathcal{R}) + r\rho(z_\mathcal{R} - y_\mathcal{R})\}$$

of the function $G_{y_0}(z_\mathcal{R}) := f((y_0)_\mathcal{I}, z_\mathcal{R}) : \mathbf{R}^{n-m} \rightarrow \mathbf{R}$.

In general we can only assert that when y_0 is a local minimum of φ_r then $(y_0)_\mathcal{R}$ must be a local minimum of G_r while $(y_0)_\mathcal{I}$ is always a strict local minimum of $x_\mathcal{I} \mapsto \varphi_r(x_\mathcal{I}, (y_0)_\mathcal{R})$.

The next result completes our goal of relating the local minimizers of φ_r to the feasible solution of the MIP defined in (1) and the global minimizers of the regularized problem φ_r to the global minimizers of the original non-linear MIP defined in (1). This is a generalization of Theorem 2.2 of [12].

Theorem 8 *Suppose ρ is an ICRF. Then there exists an $\bar{r} > 0$ such that for $r > \bar{r}$ any local minimum of φ_r lies in the feasible region F . Moreover*

1. For $r > \bar{r}$ large enough, the local minima of φ_r are exactly the points of F .
2. Suppose that $\text{Range } \rho = [0, +\infty)$ (and so we may take $\bar{r} = 0$). Then for $r > 0$ small enough, the local minima of φ_r are exactly the optimal solutions of

$$\min \{g(x) \mid Ax \geq b, x_{\mathcal{I}} \in \mathbf{I}\} \quad (14)$$

in the following cases:

- (a) we have $m = n$ i.e. a pure IP, or
- (b) both g and ρ are convex functions.

Proof. Consider the first part. In view Lemma 5 part 3 we only need show that if $y_0 \in F$ then y_0 is a local minimum of φ_r for r sufficiently large. Consider $K := \varphi_r(y_0) + 2$, $r > \bar{r}$ sufficiently large so that via (12) we have the set

$$F_K(y_0) := \{x \in \mathbf{R}^m \mid \{f(x) + r\rho(x - y_0)\} \leq K\} \quad (15)$$

bounded. As f is lower semi-continuous and $r\rho(\cdot - y_0)$ is continuous their sum is also lower semi-continuous implying $F_K(y_0)$ is a closed set. Next note that as $\varphi_r(y)$ is defined as an infimum of continuous functions $y \mapsto f(x) + r\rho(x - y)$ it is upper semi-continuous. Thus we may take y sufficiently close to y_0 so that $\varphi_r(y) + 1 < \varphi_r(y_0) + 2 := K$ and for all $x \in F_K(y_0)$ with $x_{\mathcal{I}} \neq (y_0)_{\mathcal{I}}$ then

$$\begin{aligned} \rho(x - y) &= \rho(x - y_0) + [\rho(x - y) - \rho(x - y_0)] \\ &\geq \inf \{\rho(x - y_0) \mid x \in F_K(y_0) \text{ with } x_{\mathcal{I}} \neq (y_0)_{\mathcal{I}}\} \\ &\quad - [\rho(x - y_0) - \rho(x - y)] \\ &= \epsilon - [\rho(x - y_0) - \rho(x - y)] > 0 \end{aligned}$$

where $\inf \{\rho(x - y_0) \mid x \in F_K(y_0) \text{ with } x_{\mathcal{I}} \neq (y_0)_{\mathcal{I}}\} := \epsilon > 0$. Take $r > 0$ sufficiently large so that $\frac{\epsilon}{2} > \frac{\Gamma}{r} > 0$ where

$$\Gamma := \max \{f(x) - f(x') \mid x, x' \in F_K(y_0)\} = \max \{g(x) - g(x') \mid x, x' \in F \cap F_K(y_0)\},$$

which is finite as g is continuous, $F_K(y_0)$ is compact and F is closed. We will assume we take y sufficiently close to y_0 so that at least $[\rho(x - y_0) - \rho(x - y)] < \frac{\epsilon}{2}$ and hence $\rho(x - y) \geq \frac{\epsilon}{2} > \frac{\Gamma}{r}$.

Then for $x \in F_K(y_0)$ with $x_{\mathcal{I}} \neq (y_0)_{\mathcal{I}}$ we have (as $y_0 \in F_K(y_0)$)

$$\begin{aligned} f(x) + r\rho(y - x) &= \{f(x) - f(y_0) + r\rho(x - y)\} + f(y_0) \\ &\geq -\Gamma + r\frac{\epsilon}{2} + f(y_0) \geq -\Gamma + r\frac{\Gamma}{r} + f(y_0) = f(y_0) \geq \varphi_r(y_0). \end{aligned}$$

Next note that if $x \in F_{K-1}(y)$ and we have y is sufficiently close to y_0 so that $r|\rho(x - y) - \rho(x - y_0)| \leq 1$ then

$$\begin{aligned} K - 1 &\geq f(x) + r\rho(x - y) \\ &= f(x) + r\rho(x - y_0) + r(\rho(x - y) - \rho(x - y_0)) \\ &\geq f(x) + r\rho(x - y_0) - 1 \end{aligned}$$

and so $F_{K-1}(y) \subseteq F_K(y_0)$. Consequently

$$\begin{aligned} \varphi_r(y) &= \inf \{f(x) + r\rho(y - x) \mid x \in F_{K-1}(y)\} \\ &\geq \varphi_r(y_0). \end{aligned}$$

For the second part 2 we argue again along the lines of Theorem 2.2 of [12]. Let y_0 be a local minimizer of φ_r and we already know that $y_0 \in F$ and that $\varphi_r(y_0) = f(y_0)$ and need to show that y_0 is optimal for the MIP for r small. Denote V^* the optimal solutions of the MIP. Place $V_{\mathcal{I}}^* := \{x_{\mathcal{I}} \mid x \in V^*\}$,

$$\begin{aligned} v^+ &= \min \{f(x) \mid x \in F \cap \{x \mid x_{\mathcal{I}} \notin V_{\mathcal{I}}^*\}\} \quad \text{and} \\ v^* &= \min \{f(x) \mid x \in F\}. \end{aligned}$$

Now as the variables are partly integral valued we have $v^+ > v^*$ as $x_{\mathcal{I}} \neq y_{\mathcal{I}}$ if $y \in V^*$ and $x_{\mathcal{I}} \in V_{\mathcal{I}}^*$. Let $K = v^* + 1$ and suppose contrary to assertion that $y_0 \notin V^*$ for any $r > 0$. Denote

$$D := \max \{ \rho(x - y_0) \mid x_{\mathcal{I}} \neq (y_0)_{\mathcal{I}}, x \in V^* \}.$$

When $D = 0$ we immediately have $(y_0)_{\mathcal{I}} \in V_{\mathcal{I}}^*$. Otherwise $D > 0$, then we choose $0 < r < (v^+ - v^*)/D$ and we have for any $x^* \in V^*$

$$\begin{aligned} f(y_0) &= \varphi_r(y_0) \leq f(x^*) + r\rho(x^* - y_0) \\ &\leq f(x^*) + rD < f(x^*) + v^+ - v^* = v^+. \end{aligned}$$

As $f(y_0) < v^+$ we must have $(y_0)_{\mathcal{I}} \in V_{\mathcal{I}}^*$ once again. If $n = m$ (i.e. we have a IP) we are finished as then $y_0 \in V^*$.

In the final case we assume contrary to theorem that $(y_0)_{\mathcal{R}} \notin V_{\mathcal{R}}^*$ and $v^* < f((y_0)_{\mathcal{I}}, (y_0)_{\mathcal{R}}) = f(y_0)$. As $(y_0)_{\mathcal{I}} \in V_{\mathcal{I}}^*$, $y_0 \in F$ and $y_0 \notin V^*$ we must have the existence $\bar{y}_{\mathcal{R}}$ such that $((y_0)_{\mathcal{I}}, \bar{y}_{\mathcal{R}}) \in F$ and $v^* = f((y_0)_{\mathcal{I}}, \bar{y}_{\mathcal{R}}) < f((y_0)_{\mathcal{I}}, (y_0)_{\mathcal{R}})$. Thus by Lemma 5 part 3 we have

$$\begin{aligned} \varphi_r((y_0)_{\mathcal{I}}, \bar{y}_{\mathcal{R}}) &\leq f((y_0)_{\mathcal{I}}, \bar{y}_{\mathcal{R}}) \\ &< f((y_0)_{\mathcal{I}}, (y_0)_{\mathcal{R}}) = \min_y \{ f(y) + r(y - y_0) \} = \varphi_r(y_0). \end{aligned}$$

For t sufficiently small we have $(1-t)y_0 + t((y_0)_{\mathcal{I}}, \bar{y}_{\mathcal{R}})$ in the neighborhood in which y_0 is the local minimum of φ_r . When ρ and g are convex we have φ_r convex and so

$$\begin{aligned} \varphi_r((1-t)y_0 + t((y_0)_{\mathcal{I}}, \bar{y}_{\mathcal{R}})) &\leq (1-t)\varphi_r(y_0) + t\varphi_r((y_0)_{\mathcal{I}}, \bar{y}_{\mathcal{R}}) \\ &< (1-t)\varphi_r(y_0) + t\varphi_r(y_0) = \varphi_r(y_0) \end{aligned}$$

implying y_0 is not a local minimum of φ_r , a contradiction. ■

4 The Proximal Point Heuristic and Relaxation of Integrality

The relationship of the analysis above to FP methods lies in the observation that an OFP iteration solves (5), i.e. solves the LP relaxation of problem (11) with $g(x) = c^T x$, and the integer compatible regularization function given by $\rho(x) = \|x\|_1$. At this juncture it is important to note that $\text{Range } \rho = [0, +\infty)$ so we may take $\bar{r} = 0$, which will be the case for the remainder of the paper unless otherwise indicated. Recall the definition of φ_r^{LP} from (5) and observe that

$$\varphi_r^{LP}(y) \leq \varphi_r(y) \tag{16}$$

for all y . When $g(x) = c^T x$ the problem of calculating $\varphi_r^{LP}(y)$ may be reformulated as an LP. Indeed this is the basis of the FP strategy. Calculating $\varphi_r^{LP}(y)$ is not so easy when g is nonlinear, but we carry out the following analysis without any linearity assumptions on g , and this allows for the possibility of developing some other novel numerical method in the future.

Remark 9 *The analysis of this section (and the next) can be carried out using a much more general form of an integer compatible regularization function ρ than the l^1 norm. In particular, a general norm, or any other convex, integer compatible regularization function, would suffice. For example, if g were quadratic, one could use $\rho(x) = \|x\|^2$ and then $\varphi_r^{LP}(y)$ could be solved as a quadratic program. As we wish to emphasis our comparison with the feasibility pump, these generalisations are not pursued here.*

Placing

$$h(x) := \begin{cases} g(x) & \text{if } Ax \geq b, x \in \mathbf{R}^n \\ +\infty & \text{otherwise} \end{cases}$$

we may write φ_r^{LP} as a proximal point problem:

$$\varphi_r^{LP}(y_0) := \inf_x \{ h(x) + r\|x - y_0\|_1 \} \tag{17}$$

which is well known to define a convex, Lipschitz function of y_0 when g is convex (see [12, Theorem 1.6] and references). The same proof technique as used in Lemma 5 shows that if y_0 is a local minimum of

φ_r^{LP} then $x = y_0$ is the unique minimum point of $h(\cdot) + r\rho\|\cdot - y_0\|_1$. Now in the event that $(y_0)_{\mathcal{I}} \in \mathbf{I}$ then clearly $\varphi_r^{LP}(y_0) = \varphi_r(y_0)$ and $x = y_0$ is the unique minimum point of $h(\cdot) + r\rho\|\cdot - y_0\|_1$ over x constrained to $x_{\mathcal{I}} \in \mathbf{I}$. Furthermore using (16) and the fact that y_0 is a local minimum of φ_r^{LP} we have

$$\varphi_r(y_0) = \varphi_r^{LP}(y_0) \leq \varphi_r^{LP}(y) \leq \varphi_r(y) \quad \text{for all } y \text{ close to } y_0. \quad (18)$$

Thus y_0 must also be a local minimum of φ_r . By Theorem 8 we have $y_0 \in \{Ax \geq b, x_{\mathcal{I}} \in \mathbf{I}\}$ and moreover if g is convex and r is sufficiently small y_0 is actually optimal for (14). The OFP [1] increases r at each step of the pumping process and so between major restarts the magnitude of r is directly related to the iteration number in this pumping process. When the OFP terminates early, without any restarts, r will be smaller than it would be if multiple restarts occurred (long durations of the pumping process are not observed in numerical tests). As smaller values of r are needed for the FP to generate optimal solutions the likelihood that the FP generates an optimal point will be higher when we have fewer restarts.

Proposition 7 indicates that we can detect in $(y_0)_{\mathcal{I}} \in \mathbf{I}$ a local minimizer of φ_r by checking if $x_{\mathcal{I}} = (y_0)_{\mathcal{I}} (\in \mathbf{I})$ is a unique optimal solution of (13) or in our case it suffices to check if $x_{\mathcal{I}} = (y_0)_{\mathcal{I}} (\in \mathbf{I})$ is a unique optimal solution of (17) (for any fixed $r > 0 = \bar{r}$). Now when $y_0 \in F := \{Ax \geq b, x_{\mathcal{I}} \in \mathbf{I}\}$ we have from Theorem 8 that y_0 is a local minimum of φ_r for r sufficiently large and hence $x = y_0$ is the unique optimal solution of (13). Is the same true for our relaxation (17)? The following partly answers this question.

Theorem 10 *Assume g is continuous and bounded below on $\{Ax \geq b, x \in \mathbf{I}\}$.*

1. *Suppose r is sufficiently small and y_0 with $(y_0)_{\mathcal{I}} \in \mathbf{I}$ is a local minima of φ_r^{LP} . Then y_0 is an optimal solution of (14). In particular $x = y_0$ with $(y_0)_{\mathcal{I}} \in \mathbf{I}$ is the minimum of the problem*

$$\inf \{g(x) + r\|x - y_0\|_1 \mid Ax \geq b, x \in \mathbf{R}^n\}. \quad (19)$$

2. *Suppose r is sufficiently large and y_0 with $(y_0)_{\mathcal{I}} \in \mathbf{I}$ is a local minima of φ_r^{LP} . Then $y_0 \in F := \{Ax \geq b, x_{\mathcal{I}} \in \mathbf{I}\}$ and $x = y_0$ solves the problem (19). Conversely, when g is convex and $y_0 \in F$ then $x = y_0$ with $x_{\mathcal{I}} \in \mathbf{I}$ solves the problem (19) for r sufficiently large.*

Proof. Suppose y_0 with $(y_0)_{\mathcal{I}} \in \mathbf{I}$ is a local minima of φ_r^{LP} then by (18) we have y_0 a local minimum of φ_r and we may now apply Theorem 8 to obtain all but the last assertion.

Let $y_0 \in F$ and as h is convex, lower-semi-continuous and bounded below with $\text{dom } h = \{Ax \geq b, x \in \mathbf{R}^n\}$ we may take $z_0 \in \partial h(y_0) \neq \emptyset$. Then for all $x \in \text{dom } h$ we have

$$\begin{aligned} h(x) - h(y_0) &\geq \langle z_0, x - y_0 \rangle \\ \text{implying } g(x) + r\|x - y_0\|_1 &\geq g(y_0) + r\|x - y_0\|_1 - \|z_0\|_{\infty} \|x - y_0\|_1 \\ \text{or } g(x) + r\|x - y_0\|_1 &\geq g(y_0) + (r - \|z_0\|_{\infty}) \|x - y_0\|_1 > g(y_0) \end{aligned}$$

for all $x \in \{Ax \geq b, x \in \mathbf{R}^n\}$ and all $x \neq y_0$ when $r > \|z_0\|_{\infty}$. Thus $x = y_0$ with $(y_0)_{\mathcal{I}} \in \mathbf{I}$ is the local minimum in (19). ■

One can of course directly test if a point y_0 is feasible but if this test fails how can we improve our next test point? This is where the above comes into play. Suppose that we start with an initial feasible point $x_0 \in F_{LP}$. Denote by $y_0 \in \arg \min \{\|x_0 - y\|_1 \mid y_{\mathcal{I}} \in \mathbf{I}\}$ the closest integral point. We solve (19) to obtain $x_1 \in \arg \min \{g(x) + r\|x - y_0\|_1 \mid Ax \geq b, x \in \mathbf{R}^n\}$ and find that if $x_1 \neq y_0$ then y_0 fails to be a local minimum of φ_r^{LP} . Then when $x_0 \neq x_1$ we have

$$g(x_1) + r\|x_1 - y_0\|_1 \leq g(x_0) + r\|x_0 - y_0\|_1 \quad (20)$$

$$\text{or } g(x_1) \leq g(x_0) - r[\|x_1 - y_0\|_1 - \|x_0 - y_0\|_1] < g(x_0) \quad (21)$$

a decrease in objective value as long as

$$\begin{aligned} \|x_1 - y_0\|_1 &> \|x_0 - y_0\|_1 = I(x_0) \\ &:= \min \{\|x_0 - y\|_1 \mid y_{\mathcal{I}} \in \mathbf{I}\} = \|(x_0)_{\mathcal{I}} - (y_0)_{\mathcal{I}}\|_1. \end{aligned} \quad (22)$$

Inequality (21) indicates that this process does not cycle when (22) holds. Using (20), the fact that $I(x_1) \leq \|x_1 - y_0\|_1$ and $\|x_0 - y_0\|_1 = I(x_0)$ we have for any two successive iterates

$$g(x_1) + rI(x_1) \leq g(x_0) + rI(x_0). \quad (23)$$

Thus the proximal point iteration combined with rounding always attempts to decrease the weighted sum $g + rI$. It progressively decreases the weighted sum of objective and integrality gap measure I at the feasible (possibly non-integral points). As the point x_1 may not be integral and our test requires the new center at an integral point, we must round x_1 to get $(y_1)_{\mathcal{I}} \in \mathbf{I}$.

This observation somewhat justifies the strategy used in [1] where successive guesses of a suitable $(y_0)_{\mathcal{I}} \in \mathbf{I}$ are taken by solving the problem (19) and rounding the answer x_1 to a new estimate of y_1 . This provides a new optimization problem $\varphi_r^{LP}(y_1)$ to solve for another $x_2 \in \{Ax \geq b, x \in \mathbf{R}^n\} = F_{LP}$ etc. The parameter r starts off small and is increased as the algorithm proceeds, the object being to generate a “good” feasible solution to (14). Clearly one immediately terminates after a feasible $y_k \in \{Ax \geq b, x_{\mathcal{I}} \in \mathbf{I}\} = F$ is obtained.

Our model algorithm for a nonlinear MIP FP method is thus

$$\begin{aligned} x^{k+1} &\in \arg \min \{g(x) + r^{k+1} \|x - y^k\|_1 \mid Ax \geq b, x \in \mathbf{R}^n\}, \\ y^{k+1} &\in \mathcal{R}(x^{k+1}) := \arg \min_{z: z_{\mathcal{I}} \in \mathbf{I}} \|x^{k+1} - z\|_1 \text{ (a rounding)}, \end{aligned} \quad (\text{FP})$$

which we call the *discrete proximal point algorithm*. This gives a sequence of iterations that must have

$$g(x^{k+1}) + r^{k+1}I(x^{k+1}) \leq g(x^k) + r^{k+1}I(x^k).$$

Forming a telescoping series for $k = 0, \dots, N$ we find

$$g(x^N) - g(x^0) \leq - \sum_{k=0}^N r^{k+1} [I(x^{k+1}) - I(x^k)]$$

and as g is bounded below on F_{LP} and $\{r^k\}$ is bounded away from zero we see that an infinitely long sequence of iterates would lead to $I(x^{k+1}) - I(x^k) \rightarrow 0$. This indicates cycling is inevitable: the decrease in I will stall either at a zero value (a feasible integral solution) or one with a positive gap from integrality around which the algorithm will cycle.

5 An Interpretation of Cycling in the FP

So far we have shown that the discrete proximal point algorithm is a descent algorithm for the function $g + rI$. In this section we want to explain why we can associate the non-integral local minimizers of this function with the points around which the discrete proximal point algorithm will cycle. From this we may deduce that a local minimum over F_{LP} is obtained from this iterative process.

For the purposes of this section we say that the discrete proximal algorithm cycles if

$$x^k \in \arg \min \{g(x) + r \|x - y^k\|_1 \mid x \in F_{LP}\},$$

where $y^k \in \mathcal{R}(x^k)$. This still leaves the choice of rounded point in $\mathcal{R}(x^k)$ ambiguous when the set contains more than one point, a case we will discuss further later. However this definition does remove ambiguity about the possibility that $\min_{F_{LP}} \{g(\cdot) + r \|\cdot - y^k\|_1\}$ has multiple optimal solutions. In this case the cycling condition is easy to check by simply substituting in x^k . Alternative ways to describe cycling would be that $y^{k+1} = y^k$ or that the process fails to further reduce $g(\cdot) + r \|\cdot - y^k\|_1$ for an infinite number of iterations. The former is the cycling definition used in all implementations of the feasibility pump, while the latter would be the more accurate theoretical description of when you’ve lost all hope to make further progress without restarting. Our definition and that used in the feasibility pump would be equivalent if the mapping operations from x^k to y^k and vice versa, were uniquely defined. Certainly if $y^{k+1} = y^k$ then $x^{k+1} \in \min_{F_{LP}} \{g(\cdot) + r \|\cdot - y^{k+1}\|_1\}$, so our definition of cycling applies. The converse is not necessarily true if x^k has a non-unique rounding, in which case it depends purely on machine accuracy whether a component is rounded up or down; FP algorithms

don't specify which way values extremely close to 0.5 should be rounded. Thus current FP algorithms may cycle through alternative roundings of the same LP feasible point, or alternative LP optima for the projection problem. In what follows, we discuss these issues and how they relate to local minima of $g + rI$.

Remark 11 *The rounding $\mathcal{R}(x)$ is unique if and only if no components of x have fractional parts 0.5. Also $|\mathcal{R}(x)| = 1$, i.e. $y \in \mathcal{R}(x)$ is unique, if and only if $y \in \mathcal{R}(x')$ for all x' sufficiently close to x .*

We claim that the local minimizers of $g + rI$ over F_{LP} correspond to points at which the discrete proximal point method cycles. First, we see that if a point at which the method cycles has a unique rounding, then it must be a local minimizer of $g + rI$. It is helpful for the sequel to note that $y \in \mathcal{R}(x)$ if and only if $I(x) = \|x - y\|_1$.

Proposition 12 *If the discrete proximal point algorithm cycles at x^k , and the rounding $y^k \in \mathcal{R}(x^k)$ is unique then x^k is a local minimizer of $g + rI$ over F_{LP} .*

Proof. From Remark 11 and since the rounding of x^k is unique, there is a neighbourhood $N(x^k)$ such that for all points $x \in N(x^k)$, $y^k \in \mathcal{R}(x)$. Suppose x^k is not a local minimum of $g + rI$. Then there exists a point x' close to x^k , and certainly within $N(x^k)$, so that $g(x') + rI(x') < g(x^k) + rI(x^k)$. Now since $y^k \in \mathcal{R}(x^k)$,

$$g(x^k) + rI(x^k) = g(x^k) + r\|x^k - y^k\|_1 \leq g(x') + r\|x' - y^k\|_1$$

since the algorithm cycles at x^k , so $x^k \in \arg \min_{F_{LP}} \{g(\cdot) + r\|\cdot - y^k\|_1\}$. But $g(x') + r\|x' - y^k\|_1 = g(x') + rI(x')$ since $y^k \in \mathcal{R}(x')$. Thus $g(x^k) + rI(x^k) \leq g(x') + rI(x')$ which is a contradiction. ■

If the rounding of x^k is not unique, then it is possible that

$$x^k \in \arg \min_{F_{LP}} \{g(\cdot) + r\|\cdot - y^k\|_1\}$$

for some $y^k \in \mathcal{R}(x^k)$, but x^k is not a local minimum of $g + rI$. In this case, arguments in Proposition 12 show that there must exist $\hat{y}^k \in \mathcal{R}(x^k)$ such that $x^k \notin \arg \min_{F_{LP}} \{g(\cdot) + r\|\cdot - \hat{y}^k\|_1\}$. We thus have the following result.

Proposition 13 *If for all $y \in \mathcal{R}(x^k)$, $x^k \in \arg \min_{F_{LP}} \{g(\cdot) + r\|\cdot - y\|_1\}$, then x^k is a local minimum of $g + rI$.*

Proof. There exists a neighbourhood of x^k , say $N(x^k)$ such that for $x' \in N(x^k)$, $\mathcal{R}(x') \subseteq \mathcal{R}(x^k)$. The result now follows from arguments in Proposition 12. ■

In general, it will not be easy to check the condition of 13, since the size of $\mathcal{R}(x^k)$ could be exponential in the number of variables. In practice, it is often the case that solutions to linear relaxations of binary integer programs have many variables at 0.5. Thus we don't expect algorithms to search all possible roundings. However, one could view the minor perturbation step of the FP method, as described, for example, in [5], as attempting a randomized search of alternative roundings. The perturbation step selects a random number of variables (denoted by TT) to "flip", i.e. change the rounding direction, and chooses the variables with highest fractional value. Thus variables either at 0.5 in the LP solution, or near it, are chosen first to flip.

Thus we can view FP methods as seeking local minima of $g + rI$, first by using the discrete proximal point algorithm to find $x^k \in \arg \min_{F_{LP}} \{g(\cdot) + r\|\cdot - y^k\|_1\}$, and second, if the rounding y^k is not unique, by randomly searching the set of all roundings. Of course, current FP methods do not explicitly choose the number of variables to flip in minor perturbation based on the number of variables at or near 0.5 in the current LP solution, so when the latter is small, the minor perturbation step tries to escape the nearby local minimum.

We are able to prove a converse form of Proposition 12 – that if x^k is a strict local minimum of $g + rI$ then the discrete proximal point algorithm cycles – when g is convex. This result does not depend on a unique rounding.

Remark 14 *The assumption that we have a strict local minimum of $g + rI$ at x^k is easily satisfied. For any convex function g take*

$$r > \max \{ \|z\| \mid z \in \partial g(x^k) \}.$$

Then if x^k is a local minimum of $g + rI$, it is a strict local minimum.

Proposition 15 *Suppose g is convex and x^k is a strict local minimum of $g + rI$ over F_{LP} . Then the discrete proximal point algorithm cycles at x^k .*

Proof. Arguing by contradiction, assume $x^k \notin \arg \min_{F_{LP}} \{g(\cdot) + r \|\cdot - y^k\|_1\}$. Then we have

$$g(x^{k+1}) + r \|x^{k+1} - y^k\|_1 \leq g(x^k) + r \|x^k - y^k\|_1.$$

Define

$$x(\lambda) := (1 - \lambda)x^k + \lambda x^{k+1}$$

then $x(\lambda) \in F_{LP}$ for all $\lambda \in [0, 1]$ and using the convexity of g we have

$$\begin{aligned} g(x(\lambda)) + rI(x(\lambda)) &\leq g(x(\lambda)) + r \|x(\lambda) - y^k\|_1 \\ &\leq (1 - \lambda) [g(x^k) + r \|x^k - y^k\|_1] \\ &\quad + \lambda [g(x^{k+1}) + r \|x^{k+1} - y^k\|_1] \\ &\leq (1 - \lambda) [g(x^k) + r \|x^k - y^k\|_1] \\ &\quad + \lambda [g(x^k) + r \|x^k - y^k\|_1] \\ &= g(x^k) + r \|x^k - y^k\|_1 = g(x^k) + rI(x^k) \end{aligned}$$

We can take λ arbitrarily small so that $x(\lambda)$ can be in any small neighborhood of x^k . Thus x^k cannot be a strict local minimum and so the result is proved by contradiction. ■

In summary we have shown that in the convex case, when r is large enough, the discrete proximal point algorithm terminates only when a local minimum of $g + rI$ over F_{LP} has been computed, and such local minima are exactly the points at which the algorithm cycles.

The use of penalties to produce a continuous version of an IP has been studied in [27] and [24]. In these papers, *global* minimizers of certain concave penalty functions are shown to correspond to solutions of an IP. We now compare our approach here with that of [27] and [24].

Proposition 16 *Suppose g is Lipschitz continuous and $n = m$ (we have a pure IP). Then for $r > 0$ sufficiently large, global minimizers of $g + rI$ over F_{LP} coincide with the optimal solutions of the IP.*

Proof. Let $\bar{x} \in \arg \min_{x \in F_{LP}} [g + rI]$. Then

$$g(\bar{x}) + rI(\bar{x}) \leq g(y) + rI(y) = g(y) \quad \text{for all } y \in F. \quad (24)$$

Thus when \bar{x} is integral we have $g(\bar{x}) + rI(\bar{x}) = g(\bar{x})$ and so \bar{x} must be an optimal solution of the IP, establishing that global minimizers are optimal IP solutions. Thus we only need to establish that \bar{x} must be integral when $r > 0$ is sufficiently large.

Counter to the assertion of the proposition we assume that \bar{x} is not integral for any sufficiently large $r > 0$. Let $\bar{y} \in \arg \min_{y \in F} \|\bar{x} - y\|_1$ and by assumption $\bar{y} \neq \bar{x}$ and $I(\bar{x}) > 0$. By definition $g(\bar{x}) + rI(\bar{x}) \leq g(y) + rI(y)$ for all $y \in F$. Denote

$$\begin{aligned} \text{lev}_{g+rI}(y) &\in \{x \mid g(x) + rI(x) \leq g(y) + rI(y)\} \\ &= \{x \mid g(x) - g(y) + r[I(x) - I(y)] \leq 0\} \subseteq \left\{x \mid \frac{I(x) - I(y)}{\|x - y\|_1} \leq \frac{1}{r}L\right\} \cup \{y\}, \end{aligned} \quad (25)$$

where $L > 0$ is the Lipschitz constant of g . By definition $\bar{x} \in \text{lev}_{g+rI}(y)$ and $y \in \text{lev}_{g+rI}(y)$. Now there exists a fixed neighborhood $B_\delta(y)$ of each y within which $I(x) = \|x - y\|_1$. Hence using (25) for $r > L$ we see that y is isolated point in the set $\text{lev}_{g+rI}(y)$ in that

$$\{x \mid g(x) + rI(x) \leq g(y) + rI(y)\} \cap [B_\delta(y) \setminus \{y\}] = \emptyset.$$

Thus for $r > L$ there must exist $\delta > 0$ (independent of r) such that $\|\bar{x} - y\|_1 \geq \delta > 0$ for all $y \in F$. Consequently

$$\bar{x} \in F_{LP} \cap \left\{ \bigcup_{y \in F} B_\delta(y) \right\}^c$$

with this closed set containing no integral points and so

$$I(\bar{x}) \geq \min \left\{ I(x) \mid x \in F_{LP} \cap \left\{ \bigcup_{y \in F} B_\delta(y) \right\}^c \right\} := \epsilon > 0$$

independent of $r > L$. By (24) and the definition of \bar{y} we have

$$L \frac{\|\bar{y} - \bar{x}\|_1}{\epsilon} \geq L \frac{\|\bar{y} - \bar{x}\|_1}{I(\bar{x})} \geq \frac{g(\bar{y}) - g(\bar{x})}{I(\bar{x})} \geq r$$

and so arrive at a contradiction by choosing $r > \max\{L, L \frac{\|\bar{y} - \bar{x}\|_1}{\epsilon}\} := \bar{r}$. Thus $\bar{x} \in F$ is the optimal solution of the IP for $r > \bar{r}$. ■

This last result indicates that (magor) random restarts in FP methods can be viewed as a way of attempting to find a global minimizer of $g + rI$. Similar techniques are used in global optimization.

Remark 17 In [27] and [24] various penalty functions for the integrality constraint are suggested. Suppose the function ρ of Section 3 is defined using the functional form

$$\rho(x) := \sum_i p(|x_i|),$$

where p is one of the functions (7) to (10), all of which are integer compatible regularization functions. Then the induced penalty on integrality that would follow from a similar analysis to that made in this paper would be

$$\rho(x) = \min_{y \in \mathbf{Z}^m} \sum_i p(|x_i - y_i|). \quad (26)$$

For binary problems we have for (9) and (10) that

$$\begin{aligned} \rho(x) &= \sum_i \min \{1 - \exp(-\alpha|x_i|), 1 - \exp(-\alpha(|1 - x_i|))\}, \\ \rho(x) + \frac{n}{2} &= \sum_i \min \left\{ [1 + \exp(-\alpha|x_i|)]^{-1}, [1 + \exp(-\alpha(|1 - x_i|))]^{-1} \right\} \end{aligned}$$

The only difference between these penalties and that given by (26) using (7) to (10) is that the parameter α directly penalizes the deviation of each $|x_i - y_i|$ from zero. In our approach we use an 'external' multiplier r to increase the penalty while [24] use an 'internal' multiplier α . In order to reproduce the approach of [24] we would be required a rewrite of the theory presented here to incorporate the use of this alternative penalty parameter α instead of the parameter r into our framework. In [24] the authors prove results similar to Proposition 16.

6 Incorporating Cutting Planes within FP

These results indicate that the points at which the discrete proximal point algorithm cycles are local minimizers of $g + rI$ over F_{LP} (at least when r is sufficiently large and g convex). If we want to escape cycling episodes we need to change the structure of optimization problem we are, in effect, solving. One way is to remove (or cut) that part of the region F_{LP} that is associated with a non-integral local minimum. This changes the feasible region over which we are minimizing. Cuts have a long history in integer programming and are used extensively in the branch and bound algorithm to help with fathoming of nodes. We have an entirely different motivation for suggesting them here. This approach differs from the use of cutting planes to form outer approximations of nonlinear feasible regions associated with nonlinear mixed integer programming problems (NLMIP) (see [8]). These outer approximates do not make cuts to the continuous relaxation of the original constraint set but instead allow the formation of a relaxation of the NLMIP to a linearly constrained integer programming problem that is solve using a MIP solver. This solution is in turn used to form additional cutting planes that refine the outer approximation (in [8] a nonlinear projection problem is solved at each stage). This process is specifically designed to handle nonlinearity of constraints. An alternative method to avoid

Algorithm 1: The use of a cutting plane procedure within FP to recover from cycling (the FPC algorithm)

Input : constraint set (A, b) and objective vector c ; parameters K, Q and $\beta \in [0, 1]$;
Initialize: solve $x^0 \in \arg \min_x \{c^T x : Ax \geq b\}$; $k \leftarrow 0$; $\alpha \leftarrow 1$; and $y^0 \leftarrow \lceil x^0 \rceil$; $(A^0, b^0) \leftarrow (A, b)$

```

/* outer loop */
while  $k < K$  do
  /* inner loop */
  while  $q < Q$  and  $k < K$  do
     $\alpha \leftarrow 0.9\alpha$ 
    solve  $x^{k+1} = \arg \min_x \{\Delta_\alpha(x, y^k) : A^k x \geq b^k\}$ 
     $k \leftarrow k + 1$ 
     $y^k \leftarrow \lceil x^k \rceil$ 
    if  $y^k \in F$  then go to line 23
    if cycle detected then
       $q \leftarrow q + 1$ 
      if  $\text{rand}(0, 1) < \beta$  then
        | perform minor perturbation on  $y^k$ 
      else
        | generate cuts to eliminate  $x^k$  and append cuts to constraint set  $(A^k, b^k)$ 
      end
    else
      |  $q \leftarrow 0$ 
    end
  end
  end
  perform major restart on  $y^k$ 
end

```

Output y^k
Output *no feasible solution found*

some cycling in NLMIPs is used in [10] where a tabu list is kept of recently found solutions. Here, rather than finding the FP iterate by solving,

$$x^{k+1} \in \arg \min_x \{c^T x + r \|y^k - x\|_1 : Ax \geq b, x \in \mathbf{R}^n\}, \quad (27)$$

we instead attempt to solve the corresponding integer program

$$x^{k+1} \in \arg \min_x \{c^T x + r \|y^k - x\|_1 : Ax \geq b, x_{\mathcal{I}} \in \mathbf{I}\} \quad (28)$$

using a cutting plane algorithm. We terminate prematurely (i.e., before arriving at an integer solution) when the increase of the objective value in subsequent iterations of the cutting plane algorithm becomes small, indicating that additional cuts are of less value. The cutting process removes feasible LP points that are not in the convex hull of the integral feasible solutions. As we wish to remove the LP feasible point that we are currently cycling around we must use the problem (28) in the cutting procedure.

Algorithm 1 outlines the basic FP framework given in [18] with the addition of the cutting phase given in step 1. Starting with x^0 corresponding to the optimal solution over F_{LP} and y^0 corresponding to its rounded value (denoted by $\lceil x^0 \rceil$), at each iteration the algorithm minimises the distance between y^k and x^{k+1} during the pumping process (i.e., step 1 within the algorithm). Here, we solve the following LP that is clearly equivalent to (27) but parameterized differently:

$$\begin{aligned} \Delta_\alpha(x, y^k) &:= \sqrt{S} \min_{x \in F_{LP}} \left\{ \alpha \frac{c^T x}{\|c\|_2} + (1 - \alpha) \frac{\|x - y^k\|_1}{\sqrt{S}} \right\} \\ &\equiv \min_{x \in F_{LP}} \left\{ \alpha \frac{\sqrt{S} c^T x}{\|c\|_2} + (1 - \alpha) \|x - y^k\|_1 \right\} \end{aligned} \quad (29)$$

for $\alpha \in [0, 1]$ where $\|\cdot\|_2$ is the Euclidean norm and S is the number of integer variables in the current problem (this will be l when the problem is a mixed binary problem). One now drives $\alpha^k \rightarrow 0$ instead of $r^k \rightarrow \infty$. In passing note that the case $\alpha = 0$ corresponds to the original FP of [17]. In practice, the OFP codes [1] use objective (29). This is also the objective used in the FP 2.0 of [18] and in all our numerical experiments.

The problem defined in (29) is formulated and solved as an LP [17]. For binary problems this reformulation does not require the introduction of any additional artificial variables within the LP. Thus, we can carry any cuts generated into future iterations. Unfortunately, for general integer programs, the addition of artificial variables changes the structure of the problem, making cuts in one iterate not directly transferrable to future iterates. Carrying artificial variable forward would result in a rapid increase in the problems size so here we have focussed on (mixed) binary problems.

As in standard implementations, Algorithm 1 performs a major restart when Q consecutive attempts to recover from cycling within the inner loop fails. However, unlike standard FP, the inner loop contains both a perturbation and cutting phase to recover from cycling. The cutting phase is selected with probability $1 - \beta$ where $\beta \in [0, 1]$ is a parameter of the algorithm. When $\beta = 1$, the algorithm uses only the minor perturbations as per usual FP methods.

7 Computational Results

The results reported in this section are based on 171 binary and mixed binary problems from MIPLIB2003¹ and COR@L² libraries for which the optimal solutions are known. The cutting-plane procedure outlined in Section 6 was implemented within the Feasibility Pump 2.0 ([18]) code that was kindly provided to us by Fischetti and Salvagnin. In all cases, where applicable, we have used the default settings in FP 2.0 including all the parameters, tolerances, and limits related to FP 2.0.

To evaluate the merits of the proposed cutting-plane procedure against FP 2.0, we have tested each of the 171 instances against both FP 2.0, and FP 2.0 augmented with the cutting-plane procedure outlined in Algorithm 1, henceforth denoted by FPC. We claim that any improvements in the solution quality from using cutting planes within the FP is due to the reduction in major restarts as a result of selectively tightening the LP feasible region enabling iterates to escape points where FP cycles, and not simply as a result of any tightening of the formulation around the LP optimal region. To cement this claim, we also test all instances against a scheme denoted by FPP in which FP 2.0 is run on the root node relaxation of F rather than on F_{LP} , i.e., FPP is the scheme in which FP 2.0 is run on the LP obtained by relaxing the integrality requirements but including any preprocessing and cuts that may be generated from the integrality requirements at the root node. Finally, we also test a scheme denoted by FPPC in which FPC is combined with FPP, i.e., FPC is run on the root node relaxation of F rather than on F_{LP} .

Note that in our computational experiments, we have used the available constraint propagation engine together with rounding rather than use rounding only [18]. The constraint propagation complements our cutting framework as it can take advantage of the cuts generated. As an side, we note that when constraint propagation was turned off, sufficient improvement using cutting was obtained to make FPC without propagation at least as good as FP 2.0 with propagation. Finally, we found that $\beta = 0.5$ gave an effective mechanism to recover from cycling, alternating randomly between minor perturbations and the cutting procedure. Experiments suggests that regular use of the cutting phase helps shape the feasible region more uniformly in a way that aides FPC to progress by avoiding restarts.

We use CPLEX 12.1³ to solve the LP (29), to solve the root node relaxation within FPP and FPPC, and to generate cuts within FPC. Since we cannot interject in the cut generation process of CPLEX, we have restricted our test-bed to binary and mixed binary problems where artificial variables are not needed to solve (29) and thus, the cuts generated can be passed to future iterates (see discussion in Section 6). In the case of FPC, the cut generation process is terminated when it fails to increase the objective value of (29) by at least 30% from the previous value. This occurs fairly rapidly and encourages progress without excessive time being consumed in the cutting phase.

Finally, due to the random effects introduced by perturbations and major restarts, each instance is tested on each of the four schemes on 10 independent runs. Furthermore, since the 171 instances

¹<http://miplib.zib.de/miplib2003.php>

²<http://coral.ie.lehigh.edu/data-sets/mixed-integer-instances/>

³<http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>

present problems that pose varied levels of difficulty for FP 2.0, we classify each of the 171 instances into one of three classes depending on the number of major restarts needed before an integer feasible solution is obtained using standard FP 2.0. Class A includes 110 problems that terminated with no major restarts for at least 7 out of the 10 runs, Class B 30 problems not in class A with less than 10 major restarts for at least 7 out of 10 runs, and Class C the remaining 31 problems.

	FP 2.0	FPP	FPC	FPPC
Class A	0.02	0.18	0.18	0.12
Class B	2.9	13.43	0.83	1.70
Class C	93.3	99.6	3.1	2.76

Table 1: Average number of major restarts for each scheme over all runs associated with instances belonging to each class

Table 1 reports the average number of major restarts for each scheme over all runs associated with instances belonging to a given class. Hence, if I_A , I_B , and I_C are the set of instances belonging to class A, B, and C respectively, $S = \{\text{FP 2.0, FPP, FPC, FPPC}\}$ is the set of schemes, and R_{ij}^s is the number of major restarts observed during the j^{th} (out of 10) run of instance i using some scheme $s \in S$, then for each scheme $s \in S$ and class $l \in \{A, B, C\}$, Table 1 reports

$$\bar{R}_l^s = \frac{1}{10|I_l|} \sum_{i \in I_l} \sum_{j=1}^{10} R_{ij}^s.$$

Additionally, the box plots given in Figure 1, gives the distribution of the average (over the 10 independent runs for each instance) number of major restarts by scheme and problem class. More precisely, for each class $l \in \{A, B, C\}$, and each scheme $s \in S$, Figure 1 plots the distribution of points

$$\left\{ \frac{1}{10} \sum_{j=1}^{10} R_{ij}^s : i \in I_l \right\}.$$

To present a clearer picture, Figure 1 includes box plots of these distributions with and without outliers. Points are considered as outliers if they are larger than $q_3 + 1.5(q_3 - q_1)$ or smaller than $q_1 - 1.5(q_3 - q_1)$, where q_1 and q_3 are the 25th and 75th percentiles, respectively. These outliers correspond to a small fraction of the overall data (approximately 0.7% of the data if the data is approximately normally distributed).

From Table 1 and Figure 1 it is clear that adding cuts strategically at points where FP cycles has a dramatic effect on reducing the number of major restarts for problems in classes B and C. FPP, FPC and FPPC all increase the average number of major restarts for problems in class A. However, as can be observed from Figure 1, this is mainly due to a handful of outliers. For problems in class B, FPC reduces the number of major restarts by more than a factor of three on average however, both FPP and FPPC have more restarts than FP 2.0 and FPC respectively. This is not completely surprising. Indeed, tightening the formulation through cuts at the root node is likely to make the relaxation more fractional. The binary knapsack problem is a classic example of this. Here, the LP relaxation has only one fractional variable whereas any strengthening of the problem with the addition of cuts is likely to introduce greater fractionality. Similarly, for problems in class C, FPC reduces the number of major restarts by more than a factor of thirty over FP 2.0 on average whereas FPPC and FPP are comparable to FPC and FP 2.0 respectively. Finally, it is worth noting that although the number of major restarts is reduced significantly, the overall number of FP iterations remains comparatively unchanged within the various schemes. We next present the main results demonstrating the impact of the reduction in major restarts on solution quality and thus, the efficacy of the proposed cutting scheme.

Figures 2 and 3 provide performance profiles for the average and best solutions obtained respectively by the various schemes and for the various class of problems. Each plot gives the percentage of instances where a solution was obtained by a given scheme within a certain percentage to the optimal solution. More precisely, if $J_i^s \subseteq \{1, \dots, 10\}$ is the subset of runs of scheme s for instance i that produced a solution, and Z_{ij}^s is the objective value of the best solution found for instance i during run $j \in J_i^s$, then we define

$$\bar{Z}_i^s = \frac{1}{|J_i^s|} \sum_{j \in J_i^s} Z_{ij}^s$$

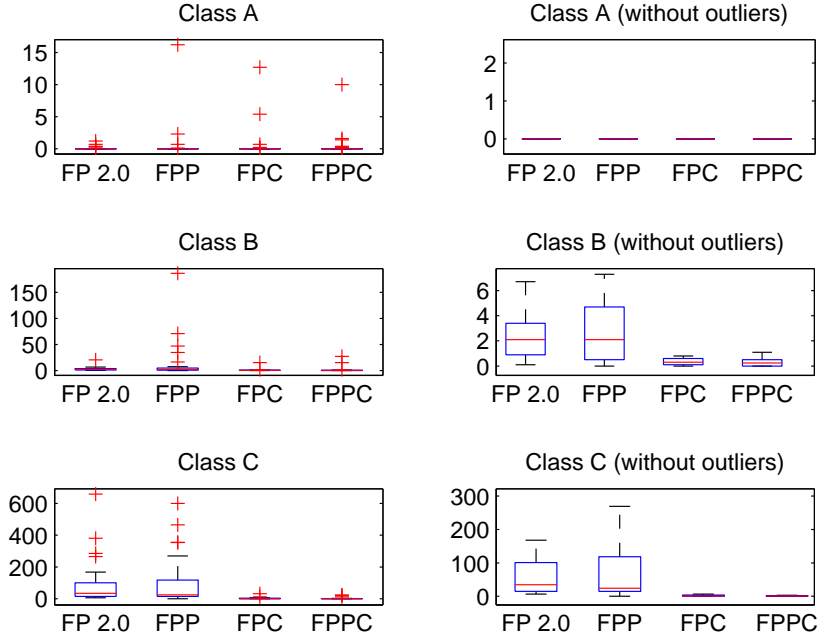


Figure 1: Distribution of the average number of major restarts by scheme and problem class

to be the average objective value of solutions found for instance i using scheme s , and

$$\hat{Z}_i^s = \max_{j \in J_i^s} Z_{ij}^s$$

to be the best objective value among solutions found for instance i using scheme s . We set \bar{Z}_i^s and \hat{Z}_i^s to ∞ if $J_i^s = \{\emptyset\}$. Moreover, given Z_i^* , the optimal solution for instance i , the percentage gap to the average objective value of solutions found for instance i using scheme s is given by

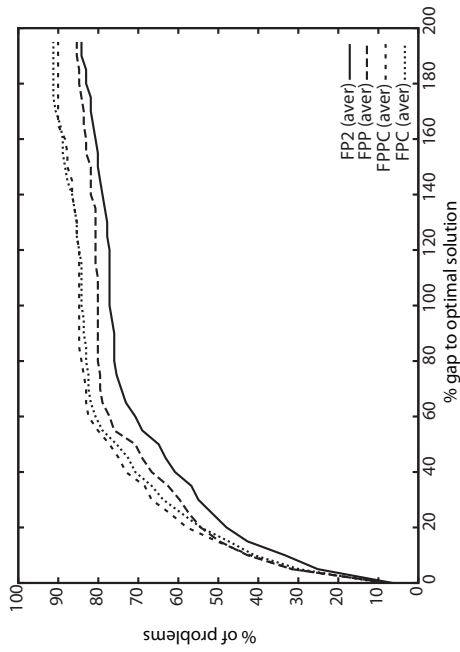
$$avgGap_i^s = \frac{(\bar{Z}_i^s - Z_i^*)}{Z_i^* + 1} \times 100,$$

and similarly, the percentage gap to the best solution found for instance i using scheme s is given by

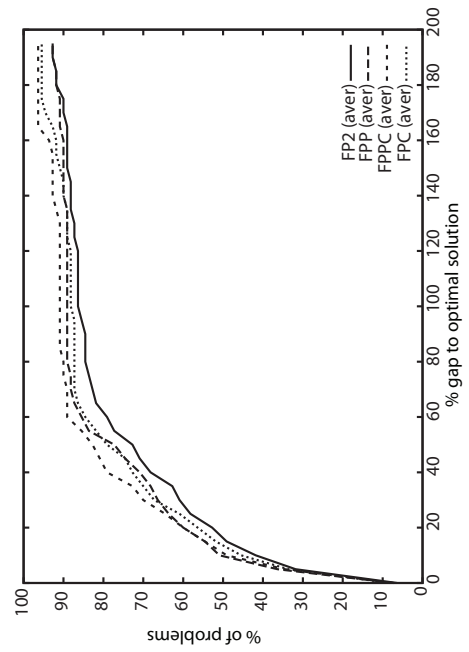
$$bestGap_i^s = \frac{(\hat{Z}_i^s - Z_i^*)}{Z_i^* + 1} \times 100.$$

Then, for each percentage $p \in [0, 100]$ and scheme s , Figure 2 plots the number of instances whose average objective is within p percent of the optimal solution, i.e. plots p against the number of instances i that have $avgGap_i^s \leq p$. Similarly, Figure 3 plots the number of instances i whose best objective is within p percent of the optimal solution, i.e. plots the number of instances i where $bestGap_i^s \leq p$. Figures 2(a) and 3(a) present this information over all instances for each of the four schemes while Figures 2(b)-2(d) and 3(b)-3(d) present this information for problems in each of the three classes A, B, and C respectively.

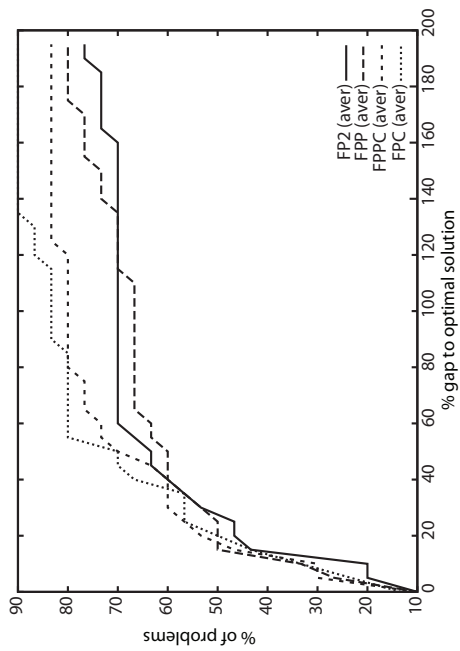
Although there is not much that separates FPC and FPPC, from Figures 2(a) and 3(a) it is clear that on the whole, the cutting plane approaches outperform FP 2.0 and FPP. When examining the three classes in isolation, we observe that for problems in class A, it is the tightening of the feasible region (FPP) that yields most of the improvement. This is to be expected as an attempt to reduce major restarts (FPC, FPPC) cannot be very useful for problems that do not require many major restarts to begin with. Here FPC does worse than FPP but better than FP 2.0 while FPPC does better than FPP. The true impact of the proposed cutting scheme as a mechanism for recovering from cycling is evident for the harder problems within classes B and C. Indeed, the percentage of problems that could



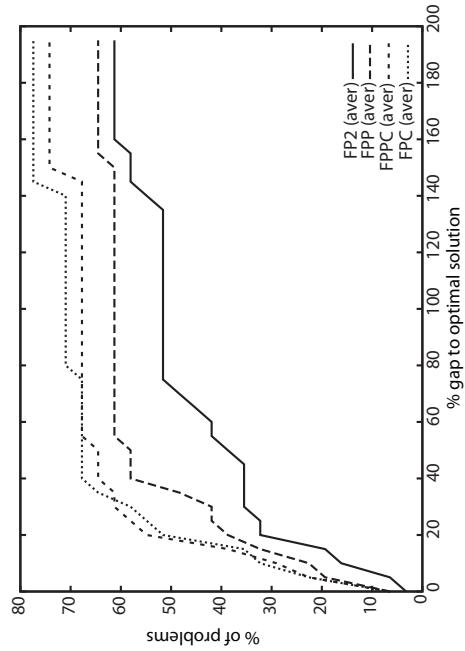
(a) All problems



(b) Class A

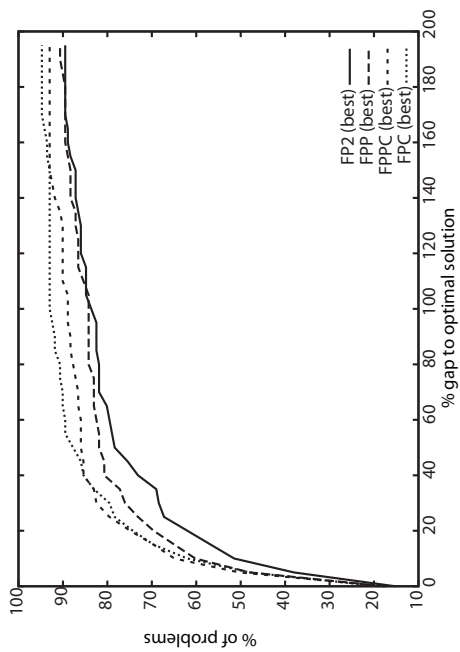


(c) Class B

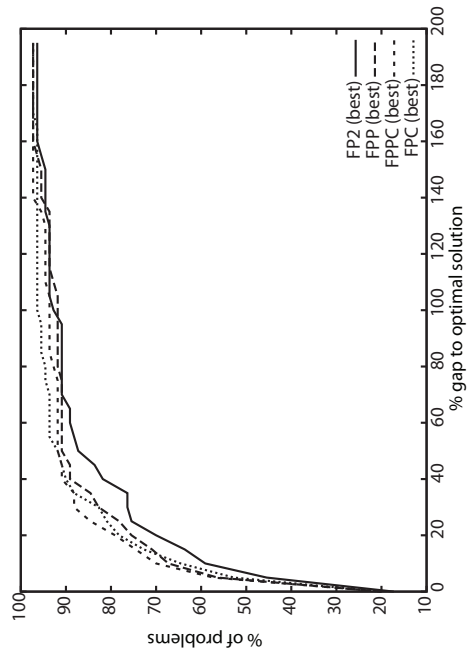


(d) Class C

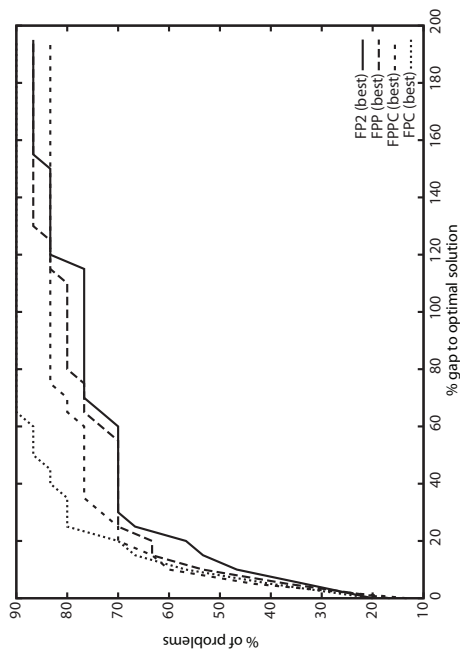
Figure 2: Percentage of instances where the average value of solutions obtained over the 10 independent runs by a given scheme is within a certain percentage to the optimal solution.



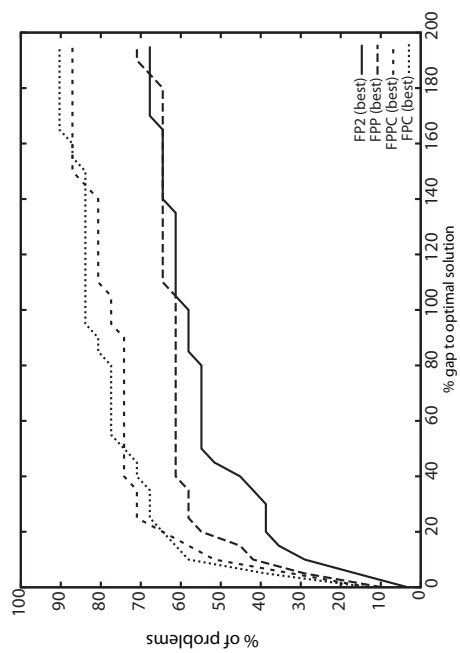
(a) All problems



(b) Class A



(c) Class B



(d) Class C

Figure 3: Percentage of instances where the value of the best solution obtained over the 10 independent runs by a given scheme is within a certain percentage to the optimal solution.

be solved to within 10% of optimality is almost twice that in FPC and FPPC as compared to FP 2.0 for class C, and almost 50% more for problems in class B.

Admittedly, by examining the relative performance of FPP versus FP 2.0, we can conclude that some of the aforementioned gains can be attributed to simply tightening the formulation at the root node. However, there are significant gains to be had over FPP by using cutting planes strategically to avoid cycling. When comparing the average objective values obtained over the 10 independent runs, FPC is better on 21 occasions, and worse on only 3 occasions for the 30 instances belonging to Class B., i.e., $|\{i \in I_B : \bar{Z}_i^{FPC} < \bar{Z}_i^{FP2.0}\}| = 21$ and $|\{i \in I_B : \bar{Z}_i^{FPC} > \bar{Z}_i^{FP2.0}\}| = 3$. For the same measure, FPC is better on 24 occasions and worse on only 6 occasions for the 31 instances in Class C. Moreover, when comparing the best objective value found over the 10 independent runs for instances in Class C, FPC was better than FP 2.0 on 26 occasions and worse on only 4 occasions, i.e., $|\{i \in I_C : \hat{Z}_i^{FPC} < \hat{Z}_i^{FP2.0}\}| = 26$ and $|\{i \in I_C : \hat{Z}_i^{FPC} > \hat{Z}_i^{FP2.0}\}| = 4$. Even if we compare the best solution found by FP 2.0 over the 10 independent runs with the average solution found by FPC, the comparison still looks favorable for FPC yielding 18 wins and 12 losses, i.e., $|\{i \in I_C : \bar{Z}_i^{FPC} < \hat{Z}_i^{FP2.0}\}| = 18$ and $|\{i \in I_C : \bar{Z}_i^{FPC} > \hat{Z}_i^{FP2.0}\}| = 12$. Finally, it is worth noting that FPC failed to find a feasible solution in all 10 runs for only one instance (for which FP 2.0 also failed), while FP 2.0 failed to find a feasible solution in all 10 runs for 5 problems. The reader is referred to Appendix B and the tables therein that provide detailed statistics comparing the results of FP 2.0 with FPC for individual instances.

	FP 2.0	FPP	FPC	FPPC
Class A	24.23	43.91	32.11	49.72
Class B	13.07	21.60	81.58	81.73
Class C	18.52	23.65	63.46	69.38

Table 2: Average time (in seconds) taken by each scheme over all runs associated with instances belonging to given class

Of course, the gains in terms of solution quality obtained from using cuts to tighten the formulation and/or recover from cycling comes at a cost in particular, the additional computational effort required in cut generation. Table 2 reports the average time (in seconds) for each scheme over all runs associated with instances belonging to a given class. Hence, if T_{ij}^s is the time taken for the j^{th} (out of 10) run of instance i using some scheme $s \in S$, then for each scheme $s \in S$ and class $l \in \{A, B, C\}$, Table 2 reports

$$\bar{T}_l^s = \frac{1}{10|I_l|} \sum_{i \in I_l} \sum_{j=1}^{10} T_{ij}^s.$$

The box plots given in Figure 4, gives the distribution of the average (over the 10 independent runs for each instance) time taken by each scheme for problems in each class. More precisely, for each class $l \in \{A, B, C\}$, and each scheme $s \in S$, Figure 1 plots the distribution of points

$$\left\{ \frac{1}{10} \sum_{j=1}^{10} T_{ij}^s : i \in I_l \right\}.$$

As before, 4 includes box plots with and without outliers.

From Table 2 and Figure 4 it is clear that incorporating cut generation within FP places a heavy burden on the computational effort required. However, almost all of the reported increase in time is due to the cut generation process itself. Hence, we are optimistic that through further tuning and a proper software solution/integration of the cutting procedure with FP (as we are using CPLEX as a “black box” only), any additional computational burden is likely to pay dividends in practical situations where time may also be of the essence.

8 Further Directions

The connection of FP methods with proximal point algorithms, and the theoretical results we have presented here, motivate several new directions of exploration for future research. One of our key insights is that the FP pumping cycle appears to be seeking local minima of a weighted combination of the original objective function and a measure of integrality. The method cycles at LP feasible points

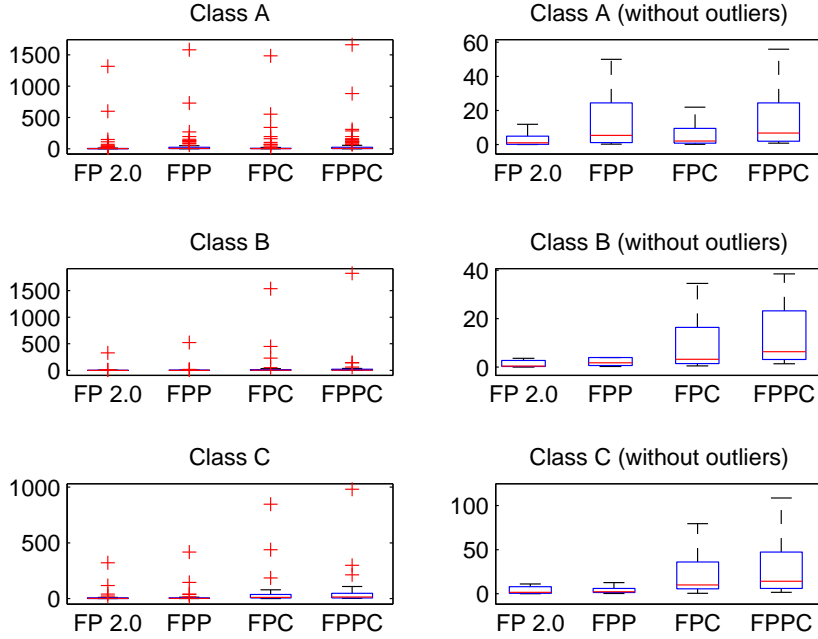


Figure 4: Distribution of the average time taken (in seconds) by each scheme for problems in each class

which are either local minima, or have multiple alternate roundings. Alternate roundings could break the cycle; if this is not possible then the point must be a local minimum. This insight motivates a different approach to the FP minor perturbation step, which chooses at random a number of variables to have their rounding changed, and changes those variables with fractional values closest to 0.5. Since variables at 0.5 are precisely those causing alternative roundings, it thus seems that minor perturbation has two functions: (a) to carry out a randomized search of the set of alternative roundings when these are not unique, to check whether or not the algorithm is likely to be at a local minimum, and (b) to escape the local minimum. If the number of variables to have their rounding changed exceeds the number at or near 0.5, then it is (b) that is occurring, otherwise it is (a). Since the minor perturbation step does not look at the number of variables at or near 0.5, there is an opportunity to re-design this step: do (a) until there is convincing evidence the current point is like to be a local minimum, and then (b) escape it. It could be that current minor perturbation is doing (a) quite well, and our cutting approach is doing (b) well, but in both standard FP methods and our cutting plane modification, steps of (a) and (b) occur in a mixed sequence.

Another direction of further research is to explore the role of the weighting parameter, that weights integrality versus the original objectives. Current FP methods simply increase this parameter to increasingly emphasize integrality. However the theoretical results show that optimal solutions require the parameter to be sufficiently small. It seems likely that further experimentation with how this parameter is adjusted could yield more effective algorithms, and in particular that phases in which the parameter is also decreased may be beneficial to producing higher quality solutions.

A Proof of Proposition 7

Proof. In view of Lemma 5 part 3 we need only show that if $x = y_0$ is the unique solution of (13) for $y = y_0$ then it is a strict local minimum of φ_r . Note that we already know that y_0 is integral. Consider $K := \varphi_r(y_0) + 2$ and denote

$$F_K^I(y_0) := \left\{ x_I \in \mathbf{I} \mid \inf_{x_{\mathcal{R}} \in \mathbf{R}^{n-m}} \{f(x_I, x_{\mathcal{R}}) + r\rho((x_I, x_{\mathcal{R}}) - y_0)\} \leq K \right\}$$

which is bounded due to the fact that $F_K(y_0)$ as defined in (15) is bounded. Now $y \mapsto \varphi_r(y)$ is upper semi-continuous when $y \mapsto \rho(x-y)$ is continuous. Thus y close to y_0 we have $\varphi_r(y) + 1 < \varphi_r(y_0) + 2 = K$. Consequently for y sufficiently close to y_0 we have $F_{K-1}^{\mathbf{I}}(y) \subseteq F_K^{\mathbf{I}}(y_0)$. Using the discreteness of the integral part of the solutions in $F_K^{\mathbf{I}}(y_0)$ and that y_0 is the unique minimum we have some $\varepsilon > 0$ such that

$$\begin{aligned} f(x) + r\rho(x - y_0) &\geq \inf_{x_{\mathcal{R}} \in \mathbf{R}^{n-m}} \{f(x_{\mathcal{I}}, x_{\mathcal{R}}) + r\rho((x_{\mathcal{I}}, x_{\mathcal{R}}) - y_0)\} \\ &\geq f(y_0) + 2\varepsilon = \varphi_r(y_0) + 2\varepsilon \quad \text{for all } x_{\mathcal{I}} \neq (y_0)_{\mathcal{I}} \text{ with } x_{\mathcal{I}} \in F_K^{\mathbf{I}}(y_0). \end{aligned}$$

By lower semi-continuity of ρ and compactness of F_K , for y sufficiently close to y_0 and all x with $x \in F_K(y_0)$ we have

$$r\rho(x - y) \geq r\rho(x - y_0) - \varepsilon. \quad (30)$$

Consequently for all y sufficiently close to y_0

$$f(x) + r\rho(x - y) \geq \varphi_r(y_0) + \varepsilon \quad \text{for all } x_{\mathcal{I}} \neq (y_0)_{\mathcal{I}} \text{ with } x_{\mathcal{I}} \in F_K^{\mathbf{I}}(y_0).$$

Thus $\min_x \{f(x) + r\rho(x - y) \mid (y_0)_{\mathcal{I}} \neq x_{\mathcal{I}} \text{ with } x_{\mathcal{I}} \in F_K^{\mathbf{I}}(y_0)\} \geq \varphi_r(y_0) + \varepsilon$ for all y close to y_0 . Using $F_{K-1}^{\mathbf{I}}(y) \subseteq F_K^{\mathbf{I}}(y_0)$ for y sufficiently close to y_0 we have

$$\begin{aligned} \varphi_r(y) &= \\ &\min \left\{ \min_{x: x_{\mathcal{I}} \neq (y_0)_{\mathcal{I}}} \{f(x) + r\rho(x - y) \mid x_{\mathcal{I}} \in F_{K-1}^{\mathbf{I}}(y)\} \right. \\ &\quad \left. , \min_{x: x_{\mathcal{I}} = (y_0)_{\mathcal{I}}} \{f(x) + r\rho(x - y) \mid x_{\mathcal{I}} \in F_{K-1}^{\mathbf{I}}(y)\} \right\} \\ &\geq \min \left\{ \min_{x: x_{\mathcal{I}} \neq (y_0)_{\mathcal{I}}} \{f(x) + r\rho(x - y) \mid x_{\mathcal{I}} \in F_{K-1}^{\mathbf{I}}(y)\} \right. \\ &\quad \left. , \min_{x_{\mathcal{R}}} \{f((y_0)_{\mathcal{I}}, x_{\mathcal{R}}) + r\rho(x_{\mathcal{R}} - y_{\mathcal{R}}) + r\rho((y_0)_{\mathcal{I}} - y_{\mathcal{I}})\} \right\} \\ &\geq \min \{\varphi_r(y_0) + \varepsilon, G_r(y_{\mathcal{R}}) + r\rho((y_0)_{\mathcal{I}} - y_{\mathcal{I}})\}. \end{aligned} \quad (31)$$

Now note that

$$\begin{aligned} G_r((y_0)_{\mathcal{R}}) &\geq \varphi_r(y_0) = f(y_0) \\ &\geq \min_{z_{\mathcal{R}} \in \mathbf{R}^{n-m}} \{f((y_0)_{\mathcal{I}}, z_{\mathcal{R}}) + r\rho(((y_0)_{\mathcal{I}}, z_{\mathcal{R}}) - y_0)\} = G_r((y_0)_{\mathcal{R}}) \end{aligned}$$

so $G_r((y_0)_{\mathcal{R}}) = \varphi_r(y_0)$. Also $(y_0)_{\mathcal{R}}$ is a local minimum of G_r because $G_r(y_{\mathcal{R}}) \geq \varphi_r((y_0)_{\mathcal{I}}, y_{\mathcal{R}})$ for all y and y_0 is a local minimum of φ_r i.e. for $y_{\mathcal{R}}$ close to $(y_0)_{\mathcal{R}}$

$$G_r(y_{\mathcal{R}}) \geq \varphi_r((y_0)_{\mathcal{I}}, y_{\mathcal{R}}) \geq \varphi_r(y_0) = G_r((y_0)_{\mathcal{R}}).$$

Thus for y sufficiently close to y_0 we have $\varphi_y(y) \geq \varphi_r(y_0)$.

When we have an IP then $G_r(y_{\mathcal{R}}) = \varphi_r(y_0)$ and so we have

$$\varphi_r(y) \geq \min \{\varphi_r(y_0) + \varepsilon, \varphi_r(y_0) + r\rho(y_0 - y)\}.$$

Thus we have a strict local minimum at y_0 . Further more in the more general case of a MIP when $(y_0)_{\mathcal{R}}$ is a strict local minimum of G_r then the inequality in (31) again implies y_0 must also be a strict local minimum of φ_r . ■

B Detail results for FP 2.0 and FPC

In Tables 3, 4, and 5 we report detailed statistics related to the runs associated with FP 2.0 and FPC for each of the 171 instances. The first column gives the problem's name, and for each of the two schemes FP 2.0 and FPC, the remaining columns give the average gap to the optimal solution over the 10 independent runs (i.e., $avgGap_i^{FP2.0}$ and $avgGap_i^{FPC}$), the gap to the optimal solution from the best solution obtained over the 10 independent runs (i.e., $bestGap_i^{FP2.0}$ and $bestGap_i^{FPC}$), the average number of FP iterations, the average number of major restarts (i.e., $\bar{R}_i^{FP2.0}$ and \bar{R}_i^{FPC}), the average

name	GAP				ITERATIONS		MR		CUTS	TIME		SUCCESS	
	AVERAGE		BEST		FP2.0	FPC	FP2.0	FPC	FPC	FP2.0	FPC	FP2.0	FPC
	FP2.0	FPC	FP2.0	FPC									
aflow30a	133.1	63.9	104.7	18.6	17.9	15.4	0.0	0.1	12.9	0.1	0.5	10	10
aflow40b	41.5	53.0	16.9	15.3	8.3	16.2	0.0	0.1	16.1	0.1	1.5	9	10
air04	1.4	2.3	0.1	0.6	8.2	8.6	0.0	0.0	7.1	11.9	13.5	10	10
air05	2.8	3.7	1.5	1.5	6.8	9.3	0.0	0.0	0.5	3.0	4.4	10	10
cap6000	2.3	0.4	1.3	0.4	29.6	16.3	0.0	0.0	1.5	0.8	3.0	9	10
disktom	0.0	0.0	0.0	0.0	1.9	1.9	0.0	0.0	0.0	1.8	1.8	9	9
ds	961.0	747.7	553.3	442.4	78.8	80.5	0.0	0.0	0.0	599.8	552.1	10	10
fast0507	5.0	5.4	3.4	1.7	3.9	3.6	0.0	0.0	0.0	8.4	8.8	10	10
fiber	244.3	45.9	133.1	3.8	17.3	7.3	0.0	0.0	19.4	0.1	0.5	10	10
fixnet6	12.2	15.2	11.3	7.3	15.0	14.7	0.0	0.0	18.8	0.0	0.8	10	10
harp2	9.3	11.0	7.1	10.7	11.9	9.4	0.0	0.0	6.0	0.1	0.5	10	10
misc07	36.6	37.3	6.6	11.2	21.9	14.6	0.1	0.0	11.6	0.1	0.6	10	10
mkc	49.7	23.7	49.7	13.1	11.2	10.0	0.0	0.0	75.9	0.2	1.6	1	10
mod011	16.5	11.3	11.8	2.0	11.7	12.5	0.0	0.0	419.1	0.3	3.2	10	10
net12	41.0	28.6	19.1	0.0	31.9	27.0	0.4	0.0	59.9	7.5	8.3	10	10
nsrand-ixp	60.9	34.3	47.2	24.4	9.4	7.6	0.0	0.0	36.9	0.5	2.1	10	10
nw04	31.2	21.4	5.9	5.9	13.4	9.3	0.0	0.0	2.0	3.5	7.0	10	10
p2756	123.3	30.6	59.6	20.8	15.0	5.1	0.0	0.0	18.1	0.1	0.6	10	10
pp08a	63.9	27.7	41.4	10.7	10.8	11.3	0.0	0.0	114.9	0.0	0.7	10	10
pp08aCUTS	38.6	21.4	22.7	4.4	9.9	10.8	0.0	0.0	61.4	0.0	0.6	10	10
protfold	52.8	57.2	43.8	31.3	18.9	34.9	0.0	0.1	3.0	10.8	23.5	10	10
qiu	496.7	478.0	68.7	167.1	8.0	8.1	0.0	0.0	1.6	0.2	0.8	10	10
seymour	3.6	3.4	2.8	2.6	4.6	4.8	0.0	0.0	0.8	2.4	2.6	10	10
sp97ar	21.3	17.8	17.4	15.4	9.3	67.0	0.0	12.7	4.1	3.1	25.1	10	9
t1717	149.3	151.9	99.3	34.6	36.5	38.8	0.0	0.0	0.0	147.2	159.4	10	10
bc1	28.0	20.4	2.4	2.4	7.8	8.1	0.0	0.0	2.9	1.2	6.0	10	9
binkar10-1	6.5	6.8	1.0	1.0	6.2	1.7	0.0	0.0	1.2	0.1	0.2	10	10
dano3-3	0.0	0.0	0.0	0.0	12.3	12.8	0.0	0.0	47.2	36.7	103.2	10	10
dano3-5	0.0	0.1	0.0	0.0	21.4	34.1	0.0	0.7	315.4	72.1	341.1	10	10
leo1	34.0	17.8	17.6	11.9	8.6	5.7	0.0	0.0	12.0	0.7	1.9	10	10
leo2	28.9	15.3	15.8	2.7	8.3	6.4	0.0	0.0	15.6	1.4	3.7	10	10
neos1	37.5	11.0	35.0	5.0	12.4	11.6	0.0	0.0	100.2	0.1	1.3	10	10
neos5	4.7	5.5	0.0	0.0	5.6	6.3	0.0	0.0	1.3	0.0	0.6	10	10
neos9	0.5	0.4	0.0	0.0	12.0	11.3	0.0	0.0	9.7	24.3	62.2	10	10
neos11	0.0	0.0	0.0	0.0	8.1	6.2	0.0	0.0	5.0	1.1	1.7	10	10
neos12	15.7	35.7	7.1	7.1	4.9	6.9	0.0	0.0	0.0	6.2	10.6	10	10
neos13	14.5	30.3	0.3	19.4	28.0	26.8	0.0	0.0	1.1	32.8	63.6	10	10
neos14	46.0	45.1	35.5	27.7	7.3	9.3	0.0	0.0	331.7	0.0	1.1	10	10
neos15	52.0	46.1	39.0	30.5	7.5	11.9	0.0	0.0	341.4	0.0	1.2	10	10
neos-476283	0.1	0.2	0.1	0.1	10.6	12.4	0.0	0.0	8.4	110.5	194.1	10	10
neos-480878	2.4	1.3	0.3	0.3	13.0	9.8	0.0	0.0	15.2	0.5	2.1	10	10
neos-495307	5.7	5.7	5.7	5.7	7.1	7.2	0.0	0.0	1.4	0.3	1.8	10	10
neos-503737	49.2	45.7	18.9	30.2	3.9	3.2	0.0	0.0	0.0	0.6	0.6	10	10
neos-504674	77.4	64.6	24.5	31.7	29.6	32.2	0.0	0.1	213.8	0.2	1.3	10	10
neos-504815	68.4	40.8	42.0	10.7	28.0	27.9	0.0	0.0	113.5	0.1	1.1	10	10
neos-506428	176.0	135.0	160.0	40.0	25.3	33.8	0.0	0.0	2.8	1316.8	1482.7	5	4
neos-512201	62.7	67.2	20.8	14.1	28.6	27.8	0.0	0.0	169.6	0.2	1.1	10	10
neos-520729	0.3	0.4	0.2	0.0	4.4	4.9	0.0	0.0	0.0	3.7	27.4	10	10
neos-525149	2.6	2.6	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.1	0.2	10	10
neos-538867	177.6	164.2	65.0	81.3	15.5	13.3	0.0	0.0	11.3	0.1	0.4	10	10
neos-538916	276.1	172.5	151.1	47.4	20.4	16.8	0.0	0.0	6.0	0.2	0.5	10	10
neos-544324	11.3	10.6	6.3	6.3	15.2	18.9	0.0	0.0	0.0	5.9	22.0	10	10
neos-547911	8.6	7.1	7.1	7.1	5.2	5.2	0.0	0.0	0.0	1.3	1.7	10	10
neos-555694	22.7	23.5	6.2	6.2	5.2	9.2	0.0	0.0	0.0	0.2	0.7	10	10
neos-565815	0.0	0.0	0.0	0.0	3.8	4.4	0.0	0.0	2.2	1.2	1.6	10	10
neos-570431	13.0	17.0	10.0	10.0	6.8	7.3	0.0	0.0	4.4	0.3	0.7	10	10
neos-584146	0.0	0.0	0.0	0.0	15.0	14.1	0.0	0.0	6.8	0.2	0.9	10	10
neos-584851	26.7	45.4	0.0	33.3	17.8	75.9	0.0	5.4	12.1	0.3	2.2	10	9
neos-611135	187.8	122.7	50.8	50.1	14.1	10.5	0.0	0.0	15.0	3.3	7.6	10	10
neos-611838	0.7	0.5	0.7	0.0	10.5	6.8	0.0	0.0	618.2	0.4	2.0	10	10
neos-612125	0.5	0.5	0.5	0.0	10.8	10.8	0.0	0.1	582.0	0.4	2.8	10	10
neos-612143	0.7	0.5	0.7	0.0	10.7	9.1	0.0	0.0	629.6	0.4	2.8	10	10
neos-612162	0.7	0.7	0.7	0.1	10.7	8.6	0.0	0.0	644.5	0.4	2.7	10	10
neos-631784	8.6	4.3	3.9	2.0	16.8	15.8	0.0	0.0	113.5	3.5	13.1	10	10
neos-691058	56.4	0.4	0.0	0.3	14.1	10.0	0.0	0.0	0.0	2.8	4.0	10	10
neos-691073	21.3	0.6	0.3	0.3	10.3	8.3	0.0	0.0	0.0	3.0	3.4	10	10
neos-738098	0.0	0.0	0.0	0.0	23.5	20.3	0.0	0.0	2.0	14.7	26.9	10	10
neos-787933	311.9	158.4	283.9	71.0	19.0	19.0	0.0	0.0	789.7	3.7	10.0	10	10
neos-791021	48.1	26.2	37.5	6.3	29.7	27.4	0.0	0.0	453.5	1.0	8.5	10	10
neos-801834	6.6	6.6	6.0	6.6	1.0	1.0	0.0	0.0	0.0	0.4	0.4	10	10
neos-808214	0.0	0.0	0.0	0.0	110.7	111.8	1.2	0.0	10.5	13.0	14.6	10	10
neos-810286	8.3	5.2	3.9	0.5	18.7	15.0	0.0	0.0	5.4	21.1	19.2	10	10
neos-810326	20.0	29.7	7.0	2.4	16.1	21.9	0.0	0.0	1.9	4.4	8.0	10	10
neos-820879	14.1	13.5	8.0	7.1	9.5	7.1	0.0	0.0	0.9	0.9	1.0	10	10

number of cuts generated (only for FPC), the average time taken (in seconds) (i.e., $\bar{T}_i^{FP2.0}$ and \bar{T}_i^{FPC}), and the number of occasions (out of the 10 independent runs) that a feasible solution was found.

Acknowledgements

The authors would like to thank Prof. Matteo Fischetti and Dr. Domenico Salvagnin for kindly providing the original code for the FP 2.0 [18] which we modified in order to perform our numerical testing. Thanks again to Matteo Fischetti for many stimulating discussions on the FP and related algorithmic issues. This work was supported by ARC Discovery grant no. DP0987445.

name	GAP				ITERATIONS				MR	CUTS	TIME		SUCCESS	
	AVERAGE		BEST		FP2.0	FPC	FP2.0	FPC	FPC	FP2.0	FPC	FP2.0	FPC	
	FP2.0	FPC	FP2.0	FPC										
neos-824695	2.2	0.6	0.0	0.0	8.9	5.2	0.0	0.0	0.5	3.3	3.9	10	10	
neos-825075	22.3	12.1	0.0	0.0	6.8	5.5	0.0	0.0	0.8	0.1	0.1	10	10	
neos-826250	0.7	0.0	0.0	0.0	4.1	3.5	0.0	0.0	0.6	1.3	1.6	10	10	
neos-826812	0.5	0.2	0.0	0.0	4.7	2.9	0.0	0.0	0.5	2.4	2.6	10	10	
neos-827015	63.7	117.0	33.9	37.4	3.8	6.3	0.0	0.0	0.0	48.4	73.0	10	10	
neos-827175	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	2.5	2.3	10	10	
neos-829552	68.2	68.2	68.2	68.2	1.0	1.0	0.0	0.0	0.0	11.7	11.9	10	10	
neos-841664	18.4	3.9	3.1	0.0	18.9	9.8	0.3	0.0	235.0	4.9	6.8	10	10	
neos-847302	12.0	0.0	0.0	0.0	12.6	10.2	0.0	0.0	2.2	1.2	1.3	10	10	
neos-860300	36.3	36.1	36.2	35.7	3.0	3.0	0.0	0.0	0.0	0.6	0.9	10	10	
neos-885524					7.3	5.1	0.0	0.0	38.7	0.9	3.2	0	0	
neos-892255	13.3	12.0	13.3	0.0	6.0	5.6	0.0	0.0	0.0	0.3	0.5	10	10	
neos-906865	0.0	7.6	0.0	0.0	7.9	8.4	0.0	0.0	2.9	0.1	0.5	10	10	
neos-932816	0.0	0.0	0.0	0.0	3.1	2.5	0.0	0.0	0.0	4.1	3.9	9	10	
neos-933364	6.9	6.9	5.6	5.4	6.5	6.5	0.0	0.0	14.2	0.1	0.9	10	10	
neos-933562	20.5	26.8	0.0	0.0	16.5	15.5	0.0	0.0	0.0	1.9	2.0	10	10	
neos-933815	7.3	6.6	5.5	4.1	7.1	6.2	0.0	0.0	0.0	0.1	0.3	10	10	
neos-933966	226.1	50.8	13.0	10.1	9.8	5.0	0.0	0.0	1.0	7.2	7.1	10	10	
neos-934184	6.5	7.2	4.7	5.0	7.0	6.6	0.0	0.0	14.2	0.1	0.8	10	10	
neos-934278	72.0	145.0	25.1	26.5	4.2	6.3	0.0	0.0	0.0	7.8	9.5	10	10	
neos-934441	54.2	58.4	24.1	31.2	7.6	4.4	0.0	0.0	0.0	10.4	9.5	10	10	
neos-935234	42.6	52.7	20.0	22.9	4.9	5.4	0.0	0.0	0.0	10.0	10.3	10	10	
neos-935348	37.2	55.1	21.1	17.3	4.5	4.0	0.0	0.0	1.5	9.3	8.9	10	10	
neos-935674	2672.3	2427.7	1600.0	784.6	8.1	6.5	0.0	0.0	1.0	1.0	1.1	10	10	
neos-935769	5.5	4.4	2.4	0.8	7.4	6.8	0.0	0.0	6.9	17.9	18.8	10	10	
neos-941698	113.3	133.3	0.0	0.0	21.5	17.2	0.0	0.0	2.0	0.7	0.8	10	10	
neos-941782	18.9	23.7	5.3	0.0	18.9	19.6	0.0	0.0	8.8	1.0	1.5	10	10	
neos-942323	26.7	25.6	16.7	16.7	19.9	25.0	0.0	0.0	6.0	0.3	0.6	10	10	
neos-948268	0.0	0.0	0.0	0.0	3.5	3.6	0.0	0.0	0.0	10.6	10.7	10	10	
neos-955215	2.2	1.7	0.4	0.4	4.3	4.6	0.0	0.0	2.4	0.0	0.3	10	10	
neos-957323	0.2	0.3	0.0	0.0	5.3	4.6	0.0	0.0	0.5	5.7	19.1	10	10	
neos-1067731	0.6	5.4	0.4	0.4	5.1	5.8	0.0	0.0	0.3	2.4	4.4	10	10	
neos-1109824	55.2	43.3	37.7	8.2	24.7	18.4	0.7	0.2	45.0	0.7	1.8	10	10	
neos-1171448	1.1	0.6	0.3	0.0	14.8	13.2	0.0	0.0	20.5	2.0	18.2	10	10	
neos-1200887	12.3	8.8	5.3	2.7	11.4	13.6	0.0	0.0	12.8	0.1	1.0	10	10	
neos-1211578	4.5	2.7	2.6	1.3	13.6	10.9	0.0	0.0	12.2	0.0	0.7	10	10	

Table 3: Detailed statistics for FP2.0 and FPC for problems in class A

name	GAP				ITERATIONS				MR	CUTS	TIME		SUCCESS	
	AVERAGE		BEST		FP2.0	FPC	FP2.0	FPC	FPC	FP2.0	FPC	FP2.0	FPC	
	FP2.0	FPC	FP2.0	FPC										
10teams	10.9	9.7	4.5	3.2	109.0	89.3	2.0	0.0	6.7	7.3	6.6	10	10	
dano3mip	12.2	7.9	6.9	6.1	68.5	85.9	0.4	0.1	5994.9	330.2	1536.6	10	4	
markshare1	21280.0	13560.0	10950.0	6100.0	68.4	67.9	3.4	0.7	68.1	0.1	2.8	10	10	
markshare2	29555.0	19915.0	13350.0	12800.0	70.0	69.7	3.0	0.3	64.4	0.2	3.1	10	10	
opt1217	3.5	0.0	0.0	0.0	38.3	33.7	0.6	0.1	7.1	0.1	1.3	10	10	
pk1	526.7	105.5	250.0	58.3	57.6	278.8	1.4	0.6	294.3	0.2	13.1	10	9	
swath	163.8	130.3	65.1	35.5	162.5	370.5	5.0	0.1	606.2	10.1	231.5	10	6	
tr12-30	27.5	37.2	23.5	22.6	27.0	21.4	0.3	0.0	835.6	0.1	3.3	10	10	
vpm2	25.4	18.1	11.9	3.4	17.0	11.6	0.4	0.0	26.2	0.0	0.5	10	10	
22433	0.3	0.1	0.0	0.0	65.0	67.8	6.1	0.2	18.7	0.3	1.4	10	10	
23588	2.4	1.9	1.1	1.0	91.8	71.7	4.3	0.1	7.7	0.3	0.8	10	10	
bienst1	13.7	12.8	0.0	0.0	53.5	102.8	2.7	0.5	537.2	0.5	34.6	10	10	
bienst2	14.7	7.4	9.1	2.1	54.7	79.4	2.5	0.6	578.7	0.6	24.9	10	10	
mcf2	19.0	22.5	1.1	9.5	96.2	455.4	20.4	0.3	2403.2	2.7	449.7	10	7	
neos-522351	36.8	2.3	19.2	0.0	50.4	27.1	1.8	1.7	263.8	0.4	3.0	10	10	
neos-582605	0.0	0.0	0.0	0.0	67.5	45.5	2.6	0.4	80.5	0.8	8.8	10	10	
neos-583731	0.0	0.0	0.0	0.0	37.9	43.5	0.1	0.0	190.6	0.2	1.4	10	10	
neos-631164	11.6	11.9	7.0	8.9	75.8	70.0	1.2	0.2	64.5	0.4	1.0	10	10	
neos-631517	10.3	10.1	7.5	8.7	75.4	72.7	0.9	0.4	53.5	0.3	0.9	10	10	
neos-777800	0.0	0.0	0.0	0.0	21.5	32.6	0.7	0.4	2.6	9.3	16.3	10	10	
neos-831188	9.0	11.2	1.3	1.2	137.8	176.5	4.4	0.0	30.7	11.9	21.7	10	10	
neos-839838	33.9	22.4	12.9	9.3	56.9	234.4	6.7	15.0	124.9	3.6	51.8	10	8	
neos-839859	235.9	39.7	61.8	13.0	42.2	19.3	1.5	0.2	30.9	0.5	2.0	10	10	
neos-886822	44.9	38.7	24.2	22.4	114.0	112.2	3.3	0.5	27.2	2.0	5.0	10	10	
neos-911880	206.4	50.9	149.6	11.0	51.0	36.7	1.0	0.6	101.1	0.1	2.0	10	10	
neos-911970	225.9	43.9	112.9	11.2	50.2	39.3	0.9	0.8	126.0	0.2	1.6	10	10	
neos-912015	49.3	51.3	26.7	20.0	164.9	188.5	2.6	0.0	7.7	8.7	11.2	10	10	
neos-1053234	70.3	32.0	42.9	17.4	39.1	34.6	4.5	0.8	112.7	0.4	6.7	10	10	
neos-1120495	56.1	50.7	21.2	18.5	17.3	14.7	0.1	0.0	35.1	0.4	1.1	10	10	
neos-1121679	2634.7	1582.4	1417.6	658.8	68.3	69.7	2.2	0.5	74.9	0.1	2.9	10	10	

Table 4: Detailed statistics for FP2.0 and FPC for problems in class B

name	GAP				ITERATIONS		MR		CUTS	TIME		SUCCESS	
	AVERAGE		BEST		FP2.0	FPC	FP2.0	FPC	FPC	FP2.0	FPC	FP2.0	FPC
	FP2.0	FPC	FP2.0	FPC									
alc1s1	60.1	34.9	38.6	21.8	51.3	18.6	13.4	0.0	744.0	0.5	3.7	10	10
danoint	17.5	17.9	5.0	10.2	112.3	353.5	33.8	0.3	2039.5	2.6	185.0	10	8
glass4	294.6	141.9	138.9	83.3	218.3	119.1	101.9	4.1	1455.1	0.3	9.8	10	10
liu	202.8	142.6	169.4	92.9	152.2	120.1	12.9	2.1	3857.0	1.9	53.0	10	10
mas74	72.3	17.7	43.1	9.1	106.2	590.7	13.9	0.4	528.6	0.1	44.3	7	5
mas76	18.8	4.3	6.3	2.3	106.2	481.9	16.7	0.3	365.0	0.2	25.9	7	5
modglob	6.6×10^6	6.4×10^6	6.5×10^6	6.4×10^6	66.1	28.1	27.2	1.5	81.5	0.1	1.7	10	10
set1ch	67.2	19.9	47.7	9.0	35.6	13.0	6.5	0.0	136.9	0.1	0.4	10	10
aligninq	1.3	1.4	0.9	1.3	249.4	185.6	39.3	0.0	38.6	9.1	12.6	9	9
neos2		202.9		197.2	843.7	763.1	124.5	1.1	1304.7	9.5	438.0	0	3
neos4		7.4		5.1	614.6	119.7	285.9	0.2	31.6	40.7	79.5	0	10
neos17	57.0	49.1	44.8	20.0	689.5	144.8	659.1	32.5	131.6	1.6	18.8	7	10
neos-430149	581.7	512.6	477.0	161.8	1208.6	172.1	266.4	0.7	280.3	5.3	27.7	2	10
neos-598183	159.0	25.3	103.4	0.6	318.2	133.6	49.2	0.8	380.9	3.0	10.8	7	10
neos-603073	139.2	1.0	31.3	0.4	180.5	21.1	64.3	0.1	140.1	1.5	1.8	10	10
neos-631694		0.0		0.0	715.8	98.9	54.7	0.1	246.6	11.0	7.4	0	10
neos-803219	9.8	13.0	7.8	4.4	1110.4	368.4	79.3	12.7	221.9	3.1	38.8	2	10
neos-803220	10.9	17.4	4.6	4.4	263.9	109.7	32.7	1.2	114.9	0.6	9.9	10	10
neos-806323	51.1	37.9	44.5	8.7	958.6	245.7	168.1	3.8	223.9	5.3	15.5	2	10
neos-807639	7.2	9.1	2.7	6.7	180.5	86.3	97.8	12.4	16.5	0.9	4.1	10	9
neos-807705	18.1	21.0	8.8	15.7	129.9	56.0	34.3	0.8	72.0	0.7	4.5	10	10
neos-848589	109303.8	7316.6	67846.4	526.6	68.9	51.9	10.9	2.5	1013.2	117.3	846.5	10	10
neos-863472	14.2	8.4	10.2	0.7	82.9	52.7	12.9	0.1	150.5	0.3	2.3	10	10
neos-880324		0.4		0.0	1053.4	240.9	380.6	6.4	320.3	1.6	7.2	0	9
neos-912023	42.9	30.0	14.3	0.0	244.5	189.3	10.6	0.0	8.2	8.7	8.0	10	10
neos-913984	0.0	0.0	0.0	0.0	1051.6	18.2	10.6	0.2	34.5	322.2	14.3	3	10
neos-916792	26.4	16.4	10.5	7.0	227.7	121.1	24.9	0.3	40.7	22.8	68.7	9	10
neos-1053591	5.0	3.1	2.0	0.1	221.9	78.7	126.4	1.1	148.0	0.3	7.4	10	10
neos-1112782	3891.8	560.9	3757.8	52.4	160.4	39.3	80.4	1.9	165.1	1.7	7.5	10	10
neos-1112787	3512.3	242.8	3366.4	37.7	85.7	45.1	34.8	4.1	158.7	0.8	6.7	10	10
neos-1173026	43.9	23.7	24.7	14.3	54.0	45.9	18.5	4.5	70.9	0.2	5.2	10	10

Table 5: Detailed statistics for FP2.0 and FPC for problems in class C

References

- [1] T. Achterberg and T. Berthold, “Improving the Feasibility Pump”, *Discrete Optimization*, 4(2007):77–86.
- [2] E. Balas, S. Ceria, M. Dawande, F. Margot, and G. Pataki, “OCTANE: A new heuristic for pure 0-1 programs”, *Operations Research*, 49(2001):207–225.
- [3] E. Balas and C.H. Martin, “Pivot and complement-a heuristic for 0-1 programming”, *Management Science*, 26(1980):86–96.
- [4] E. Balas, S. Schmieta, and C. Wallace, “Pivot and shift—a mixed integer programming heuristic”, *Discrete Optimization*, 1(2004):3–12.
- [5] L. Bertacco, M. Fischetti and A. Lodi, “A Feasibility Pump heuristic for mixed-integer problems”, *Discrete Optimization*, 4(2007):63–76.
- [6] T. Berthold, *Primal heuristics for mixed integer programs*, Master’s thesis, Technische Universität Berlin, 2006.
- [7] D.P. Bertsekas, *Nonlinear Programming*, 2nd edn. Athena Scientific, Boston, 1999.
- [8] P. Bonami, G. Cornuejols, A. Lodi and F. Margot, “Feasibility Pump for mixed integer nonlinear programs”, *Mathematical Programming*, 119(2009):331–352.
- [9] R. S. Burachik and A.N. Iusem, *Set-Valued Mappings and Enlargements of Monotone Operators*, Springer Optimization and Its Applications, Vol 8, 2008.
- [10] C. D’Ambrosio, A. Frangioni, L. Liberti, and A. Lodi, “Experiments with a feasibility pump approach for nonconvex MINLPs”, in P. Festa, editor, *Symposium on Experimental Algorithms*, LNCS Vol. 6049, Heidelberg, Springer, (2010): 350–360.
- [11] C. D’Ambrosio, A. Frangioni, L. Liberti, and A. Lodi, “A storm of feasibility pumps for nonconvex MINLP”, 2010.
- [12] A. Danilidis and C. Lemarechal, “On a primal-proximal heuristic in discrete optimization”, *Mathematical Programming*, 104(2005):105–128.
- [13] E. Danna, E. Rothberg, and C.L. Pape, “Exploring relaxation induced neighborhoods to improve MIP solutions”, *Mathematical Programming*, 102(2005):71–90.

- [14] J. Eckstein and M. Nediak, “Pivot, cut and dive: a heuristic for 0-1 mixed integer programming”, *J. Heuristics*, 13(2007):471-503.
- [15] B.H. Faaland and F.S. Hillier, “Interior path methods for heuristic integer programming procedures”, *Operations Research*, 27(1979):1069–1087.
- [16] M. Fischetti and A. Lodi, “Local branching”, *Mathematical Programming*, 98(2003):23–47.
- [17] M. Fischetti, F. Glover and A. Lodi, “The Feasibility Pump”, *Mathematical Programming*, 104(2005):91–104.
- [18] M. Fischetti and D. Salvagnin, (2009), “Feasibility Pump 2.0”, *Mathematical Programming Computation*, 1(2009):201–222.
- [19] S. Ghosh and R. Hayward, “Pivot and Gomory Cut: a 0-1 mixed integer programming heuristic”, preprint available at <http://www.cs.ualberta.ca/~hayward/papers/pgc.ps>
- [20] F.S. Hillier, “Efficient heuristic procedures for integer linear programming with an interior”, *Operations Research*, 17(1969):600–637.
- [21] T. Ibaraki, T. Ohashi, and H. Mine, “A heuristic algorithm for mixed-integer programming problems”, *Mathematical Programming Studies*, 2(1974):115–136.
- [22] N. Karmarkar, M.G.C. Resende, and K.G. Ramakrishnan, “An interior point algorithm to solve computationally difficult set covering problems”, *Mathematical Programming*, 52(1991):597–618.
- [23] A. Lokketangen and F. Glover, “Solving zero-one mixed integer programming problems using tabu search”, *European Journal of Operational Research*, 106(1998):624–658.
- [24] S. Lucidi and F. Rinaldi, “Exact penalty functions for nonlinear integer programming problems”, *Journal of Optimization Theory and Applications*, 145(2010):479-488.
- [25] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley-Interscience series in Discrete Mathematics and Optimization, 1988.
- [26] R.T. Rockafellar, “Monotone operators and the proximal point algorithm”, *SIAM J. Control Optimization*, 14(1976):877-898.
- [27] M. De Santis, S. Lucidi and F. Rinaldi, “New concave functions for improving the Feasibility Pump”, *E-prints for the Optimization Community*, http://www.optimization-online.org/DB_HTML/2010/07/2667.html