RC23862 (W0602-029) February 2, 2006
Mathematics

IBM Research Report

A Feasibility Pump for Mixed Integer Nonlinear Programs

Pierre Bonami, Gérard Cornuéjols, Andrea Lodi*, François Margot
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA

*DEIS
University of Bologna
viale Risorgimento 2
40136 Bologna, Italy

(Part of this research was carried out when Andrea Lodi was Herman Goldstine Fellow of the IBM T. J. Watson Research Center)

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Pierre Bonami · Gérard Cornuéjols
Andrea Lodi · François Margot

A Feasibility Pump for Mixed Integer Nonlinear Programs

Abstract We present an algorithm for finding a feasible solution to a convex mixed integer nonlinear program. This algorithm, called Feasibility Pump, alternates between solving nonlinear programs and mixed integer linear programs. We also discuss how the algorithm can be iterated so as to improve the first solution it finds, as well as its integration within an outer approximation scheme. We report computational results.

1 Introduction

Finding a good feasible solution to a Mixed Integer Linear Program (MILP) can be difficult, and sometimes just finding a feasible solution is an issue. Fischetti, Glover and Lodi [6] developed a heuristic for the latter which they called Feasibility Pump. Here we propose a heuristic for finding a feasible solution for Mixed Integer NonLinear Programs

---

P. Bonami
Tepper School of Business, Carnegie Mellon University, Pittsburgh PA 15213, USA. Supported in part by a grant from IBM. E-mail: pbonami@andrew.cmu.edu

G. Cornuéjols
Tepper School of Business, Carnegie Mellon University, Pittsburgh PA 15213, USA; and LIF, Faculté des Sciences de Luminy, 13288 Marseille, France. Supported in part by NSF grant DMI-0352885 and ONR grant N00014-03-1-0188. E-mail: gc0v@andrew.cmu.edu

A. Lodi
DEIS, University of Bologna, viale Risorgimento 2, 40136 Bologna, Italy. Part of this research was carried out when Andrea Lodi was Herman Goldstine Fellow of the IBM T.J. Watson Research Center whose support is gratefully acknowledged. E-mail: alodi@deis.unibo.it

F. Margot
Tepper School of Business, Carnegie Mellon University, Pittsburgh PA 15213, USA. Supported in part by a grant from IBM and by ONR grant N00014-03-1-0188. E-mail: fmargot@andrew.cmu.edu

$$(MINLP) \quad \begin{cases} \min f(x,y) \\ \text{s.t. :} \\ g(x,y) \le b \\ x \in \mathbb{Z}^{n_1} \\ y \in \mathbb{R}^{n_2} \end{cases}$$

where f is a function from $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ to $\mathbb{R}$ and g is a function from $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ to $\mathbb{R}^m$. If a variable is nonnegative, the corresponding inequality is part of the constraints $g(x,y) \le b$. In this paper, all functions are assumed to be differentiable everywhere.

For MILP (when both f and g are linear functions), the basic principle of the Feasibility Pump consists in generating a sequence of points $(\bar{x}^0,\bar{y}^0),\dots,(\bar{x}^k,\bar{y}^k)$ that satisfy the continuous relaxation. Associated with the sequence $(\bar{x}^0,\bar{y}^0)$, ..., $(\bar{x}^k,\bar{y}^k)$ of integer infeasible points is a sequence $(\hat{x}^1,\hat{y}^1),\dots,(\hat{x}^{k+1},\hat{y}^{k+1})$ of points which are integer feasible but do not necessarily satisfy the other constraints of the problem. Specifically, each $\hat{x}^{i+1}$ is the componentwise rounding of $\bar{x}^i$ and $\hat{y}^{i+1} = \bar{y}^i$. The sequence $(\bar{x}^i,\bar{y}^i)$ is generated by solving a linear program whose objective function is to minimize the distance of x to $\hat{x}^i$ according to the L_1 norm. The two sequences have the property that at each iteration the distance between $\bar{x}^i$ and $\hat{x}^{i+1}$ is nonincreasing. This basic procedure may cycle and Fischetti, Glover and Lodi use randomization to restart the procedure.

For MINLP, we construct two sequences $(\bar{x}^0,\bar{y}^0),\dots,(\bar{x}^k,\bar{y}^k)$ and $(\hat{x}^1,\hat{y}^1),\dots,$ $(\hat{x}^{k+1},\hat{y}^{k+1})$ with the following properties. The points $(\bar{x}^i,\bar{y}^i)$ in the first sequence satisfy $g(\bar{x}^i,\bar{y}^i) \le b$ but $\bar{x}^i \notin \mathbb{Z}^{n_1}$. The points $(\hat{x}^i,\hat{y}^i)$ in the second sequence do not satisfy $g(\hat{x}^i,\hat{y}^i) \le b$ but they satisfy $\hat{x}^i \in \mathbb{Z}^{n_1}$. The sequence $(\bar{x}^i,\bar{y}^i)$ is generated by solving nonlinear programs (NLP) and the sequence $(\hat{x}^i,\hat{y}^i)$ is generated by solving MILPs. We call this procedure *Feasibility Pump for MINLP* and we present two versions, a basic version and an enhanced version, which we denote *basic FP* and *enhanced FP* respectively. Unlike the procedure of Fischetti, Glover and Lodi, the enhanced FP cannot cycle and it is finite when all the integer variables are bounded. The Feasibility Pump for MINLP is a heuristic in general, but when the region $S := \{(x,y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} : g(x,y) \le b\}$ is convex, the enhanced version is an exact algorithm: either it finds a feasible solution or it proves that none exists.

The paper is organized as follows. In Section 2 we outline two versions of the Feasibility Pump for MINLP assuming that the functions g_j are convex. We present the basic version of our algorithm as well as an enhanced version. In Section 3, we present the enhanced FP in the more general case where the region $g(x,y) \le b$ is convex. In Section 4, we study the convergence of these algorithms. When constraint qualification holds, we show that the basic Feasibility Pump cannot cycle. When constraint qualification does not hold, we give an example showing that the basic FP can cycle. On the other hand, we prove that the enhanced version never cycles. It follows that, when the region $g(x,y) \le b$ is convex and the integer variables x are bounded, the enhanced FP either finds a feasible solution or proves that none exists. In Section 5, we present computational results showing the effectiveness of the method. In Section 6, we discuss how the algorithm can be iterated so as to improve the first solution it finds and we report computational experiments for such an iterated Feasibility Pump algorithm. Finally, in Section 7, we discuss the integration of the Feasibility Pump within the Outer Approximation [4] approach and we report computational results.

2 Feasibility Pump When the Functions g_j Are Convex

In this section, we consider the case where each of the functions g_j is convex for $j = 1,\ldots,m$.

To construct the sequence $(\hat{x}^1,\hat{y}^1),\ldots,(\hat{x}^{k+1},\hat{y}^{k+1})$, we use an Outer Approximation of the region $g(x,y) \leq b$. This technique was first proposed by Duran and Grossmann [4]. It linearizes the constraints of the continuous relaxation of MINLP to build a mixed integer linear relaxation of MINLP.

Consider any feasible solution $(\overline{x},\overline{y})$ of the continuous relaxation of MINLP. By convexity of the functions g_j, the constraints

$$g_j(\overline{x},\overline{y}) + \nabla g_j(\overline{x},\overline{y})^T \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} \right) \leq b_j \quad j = 1,\ldots,m \tag{1}$$

are valid for MINLP. Therefore, given any set of points $\{(\overline{x}^0,\overline{y}^0),\ldots,(\overline{x}^{i-1},\overline{y}^{i-1})\}$, we can build a relaxation of the feasible set of MINLP

$$\begin{cases} g(\overline{x}^k,\overline{y}^k) + J_g(\overline{x}^k,\overline{y}^k) \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \overline{x}^k \\ \overline{y}^k \end{pmatrix} \right) \leq b \quad \forall k = 0,\ldots,i-1 \\ x \in \mathbb{Z}^{n_1} \\ y \in \mathbb{R}^{n_2} \end{cases}$$

where J_g denotes the Jacobian matrix of function g. Our basic algorithm generates $(\hat{x}^i,\hat{y}^i)$ using this relaxation.

The basic Feasibility Pump. Initially, we choose $(\overline{x}^0,\overline{y}^0)$ to be an optimal solution of the continuous relaxation of MINLP. More generally, when the objective $f(x,y)$ is not important, we could start from any feasible solution of this continuous relaxation. Then, for $i \geq 1$, we start by finding a point $(\hat{x}^i,\hat{y}^i)$ in the current outer approximation of the constraints that solves

$$(FOA)^i \quad \begin{cases} \min \|x - \overline{x}^{i-1}\|_1 \\ \text{s.t.}: \\ g(\overline{x}^k,\overline{y}^k) + J_g(\overline{x}^k,\overline{y}^k) \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \overline{x}^k \\ \overline{y}^k \end{pmatrix} \right) \leq b \quad \forall k = 0,\ldots,i-1 \\ x \in \mathbb{Z}^{n_1} \\ y \in \mathbb{R}^{n_2}. \end{cases}$$

We then compute $(\overline{x}^i,\overline{y}^i)$ by solving the NLP

$$(FP-NLP)^i \quad \begin{cases} \min \|x-\hat{x}^i\|_2 \\ \text{s.t. :} \\ g(x,y) \leq b \\ x \in \mathbb{R}^{n_1} \\ y \in \mathbb{R}^{n_2}. \end{cases}$$

The basic FP iterates between solving $(FOA)^i$ and $(FP-NLP)^i$ until either a feasible solution of MINLP is found or $(FOA)^i$ becomes infeasible. See Figure 1 for an illustration of the Feasibility Pump.

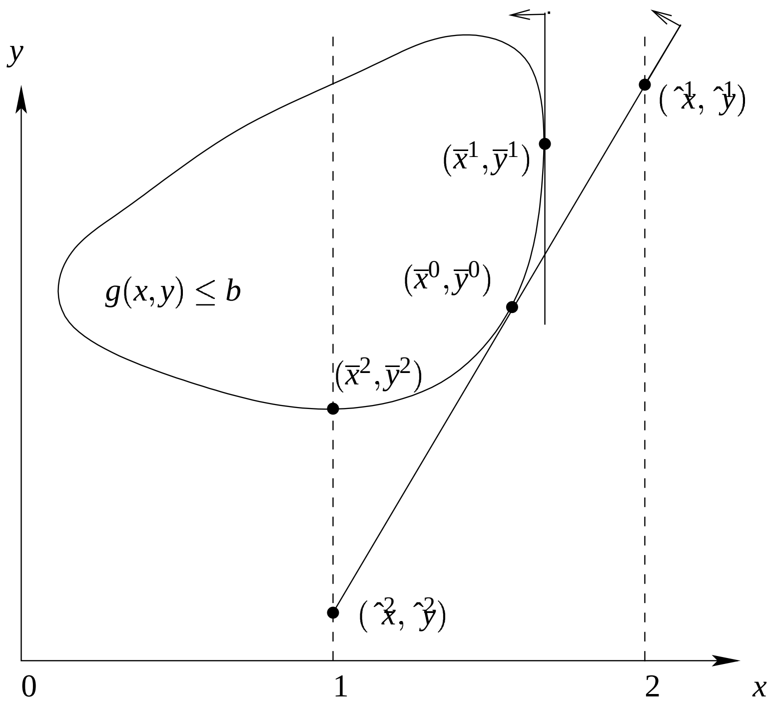

Fig. 1 Illustration of the Feasibility Pump (both basic and enhanced). The tangent lines to the feasible region represent outer approximation constraints (1) added in iterations 0 and 1.

The enhanced Feasibility Pump (case where the functions g_i are convex). In addition to the inequalities (1), other valid linear inequalities for MINLP may improve the outer approximation. For example, at iteration $k > 0$, we have a point $(\hat{x}^k, \hat{y}^k)$ outside the convex region $g(x,y) \leq b$ and a point $(\overline{x}^k, \overline{y}^k)$ on its boundary that minimizes $\|x-\hat{x}^k\|_2$. Then the inequality

$$(\overline{x}^k - \hat{x}^k)^T(x - \overline{x}^k) \geq 0 \tag{2}$$

is valid for MINLP. This is because the hyperplane that goes through $\overline{x}^k$ and is orthogonal to the vector $\overline{x}^k - \hat{x}^k$ is tangent at $\overline{x}^k$ to the projection of the convex region

$g(x,y) \leq b$ onto the x-space. Furthermore this hyperplane separates $(\hat{x}^k, \hat{y}^k)$ from the convex region $g(x,y) \leq b$. Therefore, we can add constraint (2) to $(FOA)^i$ for any $i > k$. We denote by $(SFOA)^i$ the resulting strengthening of the outer approximation $(FOA)^i$:

$$(SFOA)^i \quad \begin{cases} \min \|x - \overline{x}^{i-1}\|_1 \\ \text{s.t. :} \\ g(\overline{x}^k, \overline{y}^k) + J_g(\overline{x}^k, \overline{y}^k) \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \overline{x}^k \\ \overline{y}^k \end{pmatrix} \right) \leq b & \forall k = 0, \dots, i-1 \\ (\overline{x}^k - \hat{x}^k)^T (x - \overline{x}^k) \geq 0 & \forall k = 1, \dots, i-1 \\ x \in \mathbb{Z}^{n_1} \\ y \in \mathbb{R}^{n_2}. \end{cases}$$

Let $(\hat{x}^i, \hat{y}^i)$ denote the solution found by solving $(SFOA)^i$. The enhanced Feasibility Pump for MINLP when the functions g_i are convex iterates between solving $(SFOA)^i$ and $(FP-NLP)^i$ until either a feasible solution of MINLP is found or $(SFOA)^i$ becomes infeasible.

3 Feasibility Pump When the Region $g(x,y) \leq b$ is Convex

Let us now consider the case where the region $g(x,y) \leq b$ is convex but some of the functions defining it are nonconvex. Assume g_j is nonconvex. Then, constraint (1) may cut off part of the feasible region in general, unless $(\overline{x}, \overline{y})$ satisfies the constraint $g_j(x,y) \leq b_j$ with equality, namely $g_j(\overline{x}, \overline{y}) = b_j$, as proved in the next lemma.

Lemma 1 *Assume the region $S = \{z \in \mathbb{R}^n : g(z) \leq b\}$ is convex and let $z^* \in S$ such that $g_j(z^*) = b_j$. If g_j is differentiable at z^*, then*

$$\nabla g_j(z^*)^T (z - z^*) \leq 0$$

is valid for all $z \in S$.

Proof: Take any $\overline{z} \in S \subseteq \{z : g_j(z) \leq b_j\}$ with $\overline{z} \neq z^*$. By convexity of S, we have that, for any $\lambda \in [0,1]$, $z^* + \lambda(\overline{z} - z^*) \in S$. As $S \subseteq \{z : g_j(z) \leq b_j\}$ and $g_j(z^*) = b_j$, we get

$$g_j\left(z^* + \lambda(\overline{z} - z^*)\right) - g_j(z^*) \leq 0.$$

It follows that

$$\lim_{\lambda \to 0^+} \frac{g_j\left(z^* + \lambda(\overline{z} - z^*)\right) - g_j(z^*)}{\lambda} = \nabla g_j(z^*)^T (\overline{z} - z^*) \leq 0.$$

□

Lemma 1 shows that constraint (1) is valid for any point $(\overline{x},\overline{y})$ when g_j is convex, and for those points $(\overline{x},\overline{y})$ such that $g_j(\overline{x},\overline{y}) = b_j$ when g_j is nonconvex. Therefore for our most general version of the Feasibility Pump, we define the program $(FP-OA)^i$ obtained from $(SFOA)^i$ by using only the following subset of constraints of type (1)

$$g_j(\overline{x},\overline{y}) + \nabla g_j(\overline{x},\overline{y})^T \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} \right) \leq b_j \quad \forall j \in I(\overline{x},\overline{y}) \tag{3}$$

where $I(\overline{x},\overline{y}) = \{j : \text{either } g_j \text{ is convex or } g_j(\overline{x},\overline{y}) = b_j\} \subseteq \{1,\ldots,m\}$. Thus the program $(FP-OA)^i$ reads as follows:

$$(FP-OA)^i \begin{cases} \min \|x - \overline{x}^{i-1}\|_1 \\ \text{s.t. :} \\ g_j(\overline{x}^k,\overline{y}^k) + \nabla g_j(\overline{x}^k,\overline{y}^k)^T \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \overline{x}^k \\ \overline{y}^k \end{pmatrix} \right) \leq b_j \\ \qquad\qquad\qquad \forall j \in I(\overline{x}^k,\overline{y}^k), \forall k = 0,\ldots,i-1 \\ (\overline{x}^k - \hat{x}^k)^T (x - \overline{x}^k) \geq 0 \qquad \forall k = 1,\ldots,i-1 \\ x \in \mathbb{Z}^{n_1} \\ y \in \mathbb{R}^{n_2}. \end{cases}$$

This way of defining the constraints of $(FP-OA)^i$ gives a valid outer approximation of MINLP provided the region $g(x,y) \leq b$ is convex.

The enhanced Feasibility Pump (case where the region $g(x,y) \leq b$ is convex). The algorithm starts with a feasible solution $(\overline{x}^0,\overline{y}^0)$ of the continuous relaxation of MINLP and then iterates between solving $(FP-OA)^i$ and $(FP-NLP)^i$ for $i \geq 1$ until either a feasible solution of MINLP is found or $(FP-OA)^i$ becomes infeasible.

Note that in the case where the region $g(x,y) \leq b$ is nonconvex, the method can still be applied, but the outer approximation constraints (3) are not always valid. This may result in the problem $(FP-OA)^i$ being infeasible and the method failing while there exists some integer feasible solution to MINLP.

4 Convergence

Consider a point $(\overline{x},\overline{y})$ such that $g(\overline{x},\overline{y}) \leq b$. Let $\mathscr{I} \subseteq \{1,\ldots,m\}$ be the set of indices for which $g_j(x,y) \leq b_j$ is satisfied with equality by $(\overline{x},\overline{y})$. *The linear independence constraint qualification* (*constraint qualification* for short) is said to hold at $(\overline{x},\overline{y})$ if the vectors $\nabla g_j(\overline{x},\overline{y})$ for $i \in \mathscr{I}$ are linearly independent [5].

The next theorem shows that if the constraint qualification holds at each point $(\overline{x}^i,\overline{y}^i)$, then the basic FP cannot cycle.

Theorem 1 *In the basic FP, let $(\hat{x}^i,\hat{y}^i)$ be an optimal solution of $(FOA)^i$ and $(\overline{x}^i,\overline{y}^i)$ an optimal solution of $(FP-NLP)^i$. If the constraint qualification for $(FP-NLP)^i$ holds at $(\overline{x}^i,\overline{y}^i)$, then $\overline{x}^i \neq \overline{x}^k$ for all $k = 0,\ldots,i-1$.*

Proof:

Suppose that $\bar{x}^k = \bar{x}^i$ for some $k \leq i-1$ (i.e. $(\bar{x}^k, \bar{y}^k)$ is an optimal solution of $(FP-NLP)^i$).

Let $h(x) = \|x - \hat{x}^i\|_2$. By property of the norm, $\hat{x}^i$ satisfies

$$\nabla h(\bar{x}^k)^T (\hat{x}^i - \bar{x}^k) = -h(\bar{x}^k) < 0, \tag{4}$$

as the equality is derived from $h(x) = \sqrt{(x-\hat{x}^i)^T(x-\hat{x}^i)}$, implying $\nabla h(x) = \frac{x-\hat{x}}{h(x)}$.

Now, as a minimizer of $h(x)$ over $g(x,y) \leq b$ satisfying the constraint qualification, the point $(\bar{x}^k, \bar{y}^k)$ satisfies the KKT conditions

$$(\nabla h(\bar{x}^k)^T, 0) = -\lambda J_g(\bar{x}^k, \bar{y}^k) \tag{5}$$

$$\lambda (g(\bar{x}^k, \bar{y}^k) - b) = 0 \tag{6}$$

for some $\lambda \geq 0$. As a feasible solution of $(FOA)^i$, $(\hat{x}^i, \hat{y}^i)$ satisfies the outer-approximation the constraints in $(\bar{x}^k, \bar{y}^k)$ and therefore, since $\lambda \geq 0$:

$$\lambda g(\bar{x}^k, \bar{y}^k) + \lambda J_g(\bar{x}^k, \bar{y}^k) \left(\begin{pmatrix} \hat{x}^i \\ \hat{y}^i \end{pmatrix} - \begin{pmatrix} \bar{x}^k \\ \bar{y}^k \end{pmatrix} \right) \leq \lambda b.$$

Using (5) and (6), this implies that

$$\nabla h(\bar{x}^k)^T (\hat{x}^i - \bar{x}^k) \geq 0$$

which contradicts (4) and proves Theorem 1.

□

The proof of Theorem 1 shows that, when the constraint qualification holds, constraint (2) is implied by the outer approximations constraints at $(\bar{x}^k, \bar{y}^k)$.

Note that Theorem 1 still holds if we replace the L_2-norm by any L_p-norm in the objective of $(FP-NLP)^i$. Note also that if we replace the outer approximation constraints in $(FOA)^i$ by constraints (3), Theorem 1 still holds when the region $g(x,y) \leq b$ is convex.

Next we give an example showing that when the constraint qualification does not hold the basic algorithm may cycle.

Example 1 Consider the following constraint set for a 3-variable MINLP:

$$\begin{cases} (y_1 - \frac{1}{2})^2 + (y_2 - \frac{1}{2})^2 \leq \frac{1}{4} \\ x - y_1 \leq 0 \\ y_2 \leq 0 \\ x \in \{0,1\} \\ y \in \mathbb{R}^2. \end{cases} \tag{7}$$

The first and third constraints imply that $y_2 = 0$. Figure 2 illustrates the feasible region of the continuous relaxation, namely the line segment joining the points $(0, \frac{1}{2}, 0)$ and $(\frac{1}{2}, \frac{1}{2}, 0)$.

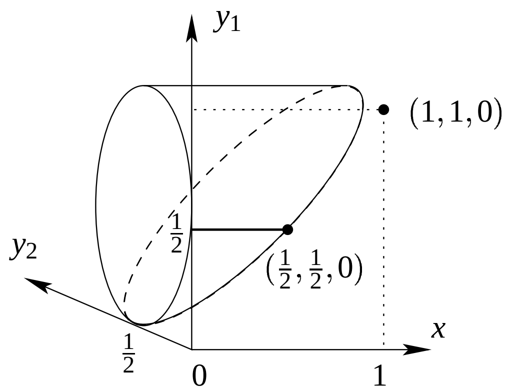

Fig. 2 Illustration of Example 1.

Starting from the point $(\hat{x},\hat{y}_1,\hat{y}_2)=(1,1,0)$, and solving $(FP-NLP)$, we get the point $(\bar{x},\bar{y}_1,\bar{y}_2)=(\frac{1}{2},\frac{1}{2},0)$. The Jacobian of g and $g(\bar{x},\bar{y}_1,\bar{y}_2)$ are given by

$$J_g(x,y_1,y_2)=\begin{pmatrix}0 & 2y_1-1 & 2y_2-1\\ 1 & -1 & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\\ -1 & 0 & 0\end{pmatrix}, \qquad g(\bar{x},\bar{y}_1,\bar{y}_2)=\begin{pmatrix}\frac{1}{4}\\ 0\\ 0\\ \frac{1}{2}\\ -\frac{1}{2}\end{pmatrix}.$$

Therefore the outer approximation constraints (1) for the point $(\bar{x},\bar{y}_1,\bar{y}_2)$ are given by

$$\begin{pmatrix}\frac{1}{4}\\ 0\\ 0\\ \frac{1}{2}\\ -\frac{1}{2}\end{pmatrix}+\begin{pmatrix}0 & 0 & -1\\ 1 & -1 & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\\ -1 & 0 & 0\end{pmatrix}\begin{pmatrix}x-\frac{1}{2}\\ y_1-\frac{1}{2}\\ y_2\end{pmatrix}\leq\begin{pmatrix}\frac{1}{4}\\ 0\\ 0\\ 1\\ 0\end{pmatrix}.$$

Among these constraints, the last four are linear constraints already present in (7), and after simplification the first one yields

$$y_2 \geq 0.$$

Since all these constraints are satisfied by $(\hat{x},\hat{y}_1,\hat{y}_2)=(1,1,0)$, this point is a feasible solution of the FOA and it is easy to verify that it is indeed optimum. Therefore we encounter a cycle. This happens since the constraint qualification does not hold at the point $(\bar{x},\bar{y}_1,\bar{y}_2)=(\frac{1}{2},\frac{1}{2},0)$. (Indeed, the first and third rows of A $J_g(\bar{x},\bar{y}_1,\bar{y}_2)$ are linearly dependent.)

□

In the next theorem, we consider the convergence of the enhanced Feasibility Pump for MINLP. In particular we prove that it cannot cycle. This is a difference with the Feasibility Pump of Fischetti, Glover and Lodi for MILP, where cycling can occur.

Theorem 2 *The enhanced Feasibility Pump cannot cycle. If the integer variables x are bounded, the enhanced FP terminates in a finite number of iterations. If, in addition, the region $g(x,y) \leq b$ is convex, the enhanced FP is an exact algorithm: either it finds a feasible solution of MINLP if one exists, or it proves that none exists.*

Proof: If for some $k \geq 0$, $(\bar{x}^k, \bar{y}^k)$ is integer feasible, the Feasibility Pump terminates. So we may assume that $(\bar{x}^k, \bar{y}^k)$ is not integer feasible. Since $(\hat{x}^k, \hat{y}^k) \neq (\bar{x}^k, \bar{y}^k)$, the point $(\hat{x}^k, \hat{y}^k)$ does not satisfy constraint (2) and therefore $\hat{x}^k$ cannot be repeated when solving $(FP-OA)^i$. Thus the enhanced Feasibility Pump cannot cycle. If the integer variables x are bounded, the enhanced Feasibility Pump is a finite algorithm, since there is only a finite number of possible different values for $\hat{x}^k$. The last part of the theorem follows from the fact that $(FP-OA)^i$ is a valid relaxation of MINLP when the region $g(x,y) \leq b$ is convex.

□

Example 2 We run the enhanced Feasibility Pump starting from the point where we were stuck with the basic algorithm in Example 1, namely $(\bar{x}^1, \bar{y}_1^1, \bar{y}_2^1) = (\frac{1}{2}, \frac{1}{2}, 0)$. The corresponding inequality (2), namely $x \leq \frac{1}{2}$, is added to $(FOA)^2$. Solving the resulting ILP $(FP-OA)^2$ yields $(\hat{x}^2, \hat{y}_1^2, \hat{y}_2^2) = (0,0,0)$. Solving $(FP-NLP)^2$, we get the point $(\bar{x}^2, \bar{y}_1^2, \bar{y}_2^2) = (0, \frac{1}{2}, 0)$, which is feasible for (7) and we stop.

□

Note that, although $\hat{x}^i$ cannot be repeated in the enhanced Feasibility Pump, the point $(\bar{x}^i, \bar{y}^i)$ could be repeated as shown by considering a slightly modified version of Example 1.

Example 3 Change the second constraint in Example 1 to $x - y_1 = 0$. Starting from the point $(\hat{x}^1, \hat{y}_1^1, \hat{y}_2^1) = (1,1,0)$, and solving $(FP-NLP)^1$, we get the point $(\bar{x}^1, \bar{y}_1^1, \bar{y}_2^1) = (\frac{1}{2}, \frac{1}{2}, 0)$. As in Example 1, the constraint qualification does not hold at this point. Here inequality (2) is $x \leq \frac{1}{2}$. We add this inequality to $(FOA)^2$ to get $(FP-OA)^2$. Solving this integer program yields $(\hat{x}^2, \hat{y}_1^2, \hat{y}_2^2) = (0,0,0)$. Solving $(FP-NLP)^2$, we get the point $(\bar{x}^2, \bar{y}_1^2, \bar{y}_2^2) = (\frac{1}{2}, \frac{1}{2}, 0)$, which is the same as $(\bar{x}^1, \bar{y}_1^1, \bar{y}_2^1)$. Although a point $(\bar{x}, \bar{y})$ is repeated, the enhanced Feasibility Pump does not cycle. Indeed, now inequality (2) is $x \geq \frac{1}{2}$. Adding it to $(FOA)^3$ yields $(FP-OA)^3$ which is infeasible. This proves that the starting MINLP is infeasible.

□

5 Computational Results

The Feasibility Pump for MINLP has been implemented in the COIN infrastructure [2] using a new framework for MINLP [1]. Our implementation uses `Ipopt3.0` to solve the nonlinear programs and `Cplex9.0` to solve the mixed integer linear programs. All the tests were performed on an IBM IntellistationZ Pro with an Intel Xeon 3.2GHz CPU, 2 gigabytes of RAM and running Linux Fedora Core 3.

We tested the Feasibility Pump on a set of 66 convex MINLP instances gathered from different sources, and featuring applications from operations research and chemical engineering. Those instances are discussed in [1,8]. In these instances, the objective function and all the functions g_j are convex.

The basic Feasibility Pump never cycles on the instances in our test set. This means that using the enhanced FP is not necessary for these instances. Therefore all the results reported in this paper are obtained with the basic Feasibility Pump.

		FP			OA		
Name	previous best	value	time	# iter	value	time	# iter
BatchS101006M	769440*	786499	0	1	782384	1	1
BatchS121208M	1241125*	1364991	0	1	1243886	3	1
BatchS151208M	1543472*	1692878	0	1	1545222	3	1
BatchS201210M	2295349*	2401369	1	1	2311640	10	1
CLay0304M	40262.40*	59269.10	2	10	40262.40	25	14
CLay0304H	40262.40*	65209.10	0	10	40262.40	7	13
CLay0305M	8092.50*	9006.76	0	3	8278.46	35	2
CLay0305H	8092.50*	8646.44	0	2	8278.46	4	3
FLay04M	54.41*	54.41	0	1	54.41	0	1
FLay04H	54.41*	54.41	0	1	54.41	0	1
FLay05M	64.50*	64.50	0	1	64.50	2	1
FLay05H	64.50*	64.50	0	1	64.50	0	1
FLay06M	66.93*	66.93	2	1	66.93	41	1
FLay06H	66.93*	66.93	0	1	66.93	19	1
fo7_2	17.75	17.75	1	2	17.75	19	1
fo7	20.73	29.94	0	1	20.73	23	1
fo8	22.38	38.01	0	1	23.91	123	1
fo9	23.46	49.80	3	4	24.00	1916	1
o7_2	116.94	159.38	0	1	118.85	5650	4
o7	131.64	171.51	3	5	—	>7200	—
RSyn0830H	-510.07*	-509.49	1	1	-510.07	1	1
RSyn0830M	-510.07*	-491.53	0	1	-497.87	0	1
RSyn0830M02H	-730.51*	-727.22	0	1	-728.23	0	1
RSyn0830M02M	-730.51*	-663.55	0	1	-712.45	365	1
RSyn0830M03H	-1543.06*	-1538.02	1	1	-1535.46	1	1
RSyn0830M03M	-1543.06*	-981.98	0	1	-1532.09	442	1
RSyn0830M04H	-2529.07*	-2519.03	2	1	-2512.04	2	1
RSyn0830M04M	-2529.07*	-2436.44	1	1	-2502.39	2584	1
RSyn0840H	-325.55*	-317.50	0	1	-325.55	1	1
RSyn0840M	-325.55*	-321.42	0	1	-325.55	0	1
RSyn0840M02H	-734.98*	-732.31	0	1	-732.31	0	1
RSyn0840M02M	-734.98*	-599.57	0	1	-721.98	209	1
RSyn0840M03H	-2742.65*	-2719.53	1	1	-2732.53	1	1
RSyn0840M03M	-2742.65*	-2525.19	3	3	-2701.81	695	1
RSyn0840M04H	-2564.50*	-2538.83	2	1	-2544.04	2	1
RSyn0840M04M	-2563.50*	-2478.67	5	3	-2488.87	7200	1

Table 1 FP vs. first solution found by OA (on the first 36 instances of the test set). Column labeled “previous best” gives the best known solution obtained using `Dicopt` and `Sbb` and ”*” indicates that the value is known to be optimal; columns labeled “value” report the objective value of the solution found, where “−” indicates that no solution is found; columns labeled “time” show the CPU time in seconds rounded to the closest integer (with a maximum of 2 hours of CPU time allowed); columns labeled “# iter” give the number of iterations.

In order to guarantee convergence to an optimal solution in Theorems 1 and 2 it is important to find an optimum solution $(\bar{x}^i, \bar{y}^i)$ of $(FP-NLP)^i$. On the other hand, it is not necessary to obtain an optimum solution of $(FP-OA)^i$. In our

		FP			OA		
Name	previous best	value	time	# iter	value	time	# iter
SLay07H	64749*	66223	0	1	69509	0	1
SLay07M	64749*	65254	0	1	65287	0	1
SLay08H	84960*	93425	0	1	115041	0	1
SLay08M	84960*	91849	0	1	91849	0	1
SLay09H	107806*	120858	0	1	115989	0	1
SLay09M	107806*	115881	0	1	117250	0	1
SLay10H	129580*	156882	0	1	156490	2	1
SLay10M	129580*	136402	0	1	163371	0	1
Syn30H	-138.16*	-111.86	0	1	-111.86	0	1
Syn30M	-138.16*	-125.19	0	1	-125.19	0	1
Syn30M02H	-399.68*	-387.37	0	1	-387.37	0	1
Syn30M02M	-399.68*	-386.25	0	1	-386.25	0	1
Syn30M03H	-654.15*	-641.84	0	1	-641.84	0	1
Syn30M03M	-654.15*	-646.05	0	1	-646.05	1	1
Syn30M04H	-865.72*	-818.12	0	1	-818.12	0	1
Syn30M04M	-865.72*	-825.75	0	1	-856.05	2	1
Syn40H	-67.71*	-61.19	0	1	-61.19	0	1
Syn40M	-67.71*	-55.71	0	1	-55.71	0	1
Syn40M02H	-388.77*	-387.04	0	1	-388.77	0	1
Syn40M02M	-388.77*	-371.48	0	1	-376.48	1	1
Syn40M03H	-395.15*	-318.64	4	1	-318.64	0	1
Syn40M03M	-395.15*	-331.69	0	1	-354.69	14	1
Syn40M04H	-901.75*	-827.71	0	1	-837.71	0	1
Syn40M04M	-901.75*	-765.20	0	1	-805.70	17	1
trimloss2	5.3*	5.3	0	3	5.3	0	6
trimloss4	8.3*	11.7	1	11	8.3	893	72
trimloss5	—	13.0	12	23	—	>7200	—
trimloss6	—	16.7	14	24	—	>7200	—
trimloss7	—	23.2	553	111	—	>7200	—
trimloss12	—	221.7	4523	243	—	>7200	—

Table 2 FP vs. first solution found by OA (on the remaining 30 instances). Table 2 is the continuation of Table 1. Symbol ”*” indicates that the value is known to be optimal, “−” indicates that no solution is found.

implementation of the Feasibility Pump, we do not insist on solving the MILP $(FP-OA)^i$ to optimality. Once a feasible solution has been found and has not been improved for 5000 nodes of the `Cplex` branch-and-cut algorithm, we use this solution as our point $(\hat{x}^i, \hat{y}^i)$. This reduces the amount of time spent solving MILPs and it improves the overall computing time of the Feasibility Pump.

In a first experiment, we compare the solution obtained with the Feasibility Pump to the first solution obtained by the Outer Approximation algorithm (*OA* for short) as implemented in [1] and using `Cplex9.0` and `Ipopt3.0` as subsolvers. Tables 1 and 2 summarize this comparison.

The following comments can be made about the results of tables 1 and 2. The Feasibility Pump finds a feasible solution in less than a second in most cases. Overall, FP is much faster than OA. Although on the `CLay` instances both FP and OA require several iterations to find a feasible solution, FP is roughly ten times faster. The `fo`, `o` and `trimloss` instances are particularly challenging for OA, which fails to find a feasible solution within the 2-hour time limit for 5 of these 12 instances. By contrast, FP finds a feasible solution to each of them. The cases of `trimloss5`, `trimloss6`, `trimloss7` and `trimloss12` are noteworthy since no feasible solution was known prior to this work. The column "previous best" contains the best known solution from `Dicopt` [3] and `Sbb`[9]. `Dicopt` is an MINLP solver based on the outer approximation technique whereas `Sbb` is a solver based on branch-and-bound.

6 Iterating the Feasibility Pump for MINLP

In the next two sections, we assume that we have a convex MINLP, that is we assume that both the region $g(x,y) \leq b$ and the objective function f are convex. This section investigates the heuristic obtained by iterating the FP, i.e. calling several time in a row FP, each time trying to find a solution strictly better than the last solution found.

More precisely, to take into account the cost function $f(x,y)$ of MINLP, we add to $(FOA)^i$ a new variable α and the constraint $f(x,y) \leq \alpha$. Initially, the variable α is unbounded. Each time a new feasible solution with value z^U to MINLP is found, the upper bound on α is decreased to $z^U - \delta$ for some small $\delta > 0$. As a result, the current best known feasible solution becomes infeasible and it is possible to restart FP from the optimal solution of the relaxation of MINLP. Note that (1) is used to generate outer approximations of the convex constraint $f(x,y) \leq \alpha$.

If executed long enough, this algorithm will ultimately find the optimal solution of MINLP and prove its optimality by application of Theorem 2 under the assumption that the integer variables are bounded and δ is small enough. Here, we do not use it as an exact algorithm but instead we just run it for a limited time. We call this heuristic *Iterated Feasibility Pump* for MINLP (or *IFP* for short).

Table 3 compares the best solutions found by iterated FP and by OA with a time limit of 1 minute of CPU time. In our experiments, we use $\delta = 10^{-4}$.

The following comments can be made about the results of Table 3. IFP produces good feasible solutions for all but 2 of the instances within the 1-minute time limit, whereas OA fails to find a feasible solution for 15 of the instances. In terms of the quality of solutions found, OA finds a strictly better solution than IFP in 9 cases while IFP is the winner in 20 cases. OA can prove optimality of its solution in 36 instances and IFP in 30 instances.

7 Application to Outer Approximation Decomposition

We now present a new variation of the Outer Approximation Decomposition algorithm of Duran and Grossmann [4] which integrates the Feasibility Pump algorithm. Duran and Grossmann assume that all the functions are convex and that the constraint qualification holds at all optimal points. Our variation of outer approximation does not need the assumption that all functions are convex, provided that the region $g(x,y) \leq b$ is a convex set and that $f(x,y)$ is convex.

In this algorithm we alternate between solving four different problems. The first one is a linear outer approximation of MINLP with the original convex ob-

Name	IFP value	OA value
BatchS101006M	769440 *	769440 *
BatchS121208M	1241125 *	1241125 *
BatchS151208M	1543472 *	1543472 *
BatchS201210M	2297282	2295349
CLay0304H	40791.00	40262.40 *
CLay0304M	41359.60	40262.40 *
CLay0305H	8092.50 *	8092.50 *
CLay0305M	8092.50 *	8278.46
FLay04H	54.41 *	54.41 *
FLay04M	54.41 *	54.41 *
FLay05H	64.50	64.50
FLay05M	64.50	64.50
FLay06H	66.93	66.93
FLay06M	66.93	66.93
fo7_2	17.75 *	17.75
fo7	20.73	20.73
fo8	22.38	—
fo9	28.95	—
o7_2	124.83	—
o7	139.87	—
RSyn0830H	-510.07 *	-510.07 *
RSyn0830M	-510.07 *	-510.07 *
RSyn0830M02H	-730.51	-730.51 *
RSyn0830M02M	-730.51	—
RSyn0830M03H	-1541.84	-1543.06 *
RSyn0830M03M	-1543.06	—
RSyn0830M04H	-2527.86	-2529.07 *
RSyn0830M04M	-2520.88	—
RSyn0840H	-325.55 *	-325.55 *
RSyn0840M	-325.55 *	-325.55 *
RSyn0840M02H	-734.98	-734.98 *
RSyn0840M02M	-734.98	—
RSyn0840M03H	-2742.65	-2742.65 *
RSyn0840M03M	-2741.65	—
RSyn0840M04H	-2561.37	-2564.50 *
RSyn0840M04M	-2499.29	—

Name	IFP value	OA value
SLay07H	64749 *	64749 *
SLay07M	64749 *	64749 *
SLay08H	84960 *	84960 *
SLay08M	84960 *	84960 *
SLay09H	107983	108910
SLay09M	107806	108125
SLay10H	132562	136401
SLay10M	129814	130447
Syn30H	-138.16 *	-138.16 *
Syn30M	-138.16 *	-138.16 *
Syn30M02H	-399.68 *	-399.68 *
Syn30M02M	-399.68 *	-399.68 *
Syn30M03H	-654.15 *	-654.16 *
Syn30M03M	-654.15 *	-654.15 *
Syn30M04H	-865.72 *	-865.72 *
Syn30M04M	-865.72 *	-865.72
Syn40H	-67.71 *	-67.71 *
Syn40M	-67.71 *	-67.71 *
Syn40M02H	-388.77 *	-388.77 *
Syn40M02M	-388.77 *	-388.77 *
Syn40M03H	-395.15 *	-395.15 *
Syn40M03M	-395.06	-395.15
Syn40M04H	-827.71	-901.75 *
Syn40M04M	-864.76	-901.75
trimloss2	5.3 *	5.3 *
trimloss4	8.3	—
trimloss5	11.8	—
trimloss6	16.7	—
trimloss7	—	—
trimloss12	—	—

Table 3 IFP vs. OA (at most 1 minute of CPU time). Columns labeled "value" report the objective value of the solution found; symbol "*" denotes proven optimality and "−" indicates that no solution is found.

jective function $f(x,y)$ being linearized as well:

$$
(OA)^i \qquad \begin{cases} \min \alpha \\ \text{s.t.}: \\ \nabla f(\overline{x}^k,\overline{y}^k)^T \begin{pmatrix} x-\overline{x}^k \\ y-\overline{y}^k \end{pmatrix} - \alpha \le -f(\overline{x}^k,\overline{y}^k) \\ \qquad\qquad\qquad\qquad\qquad\qquad k=0,\dots,i-1 \\ \nabla g_j(\overline{x}^k,\overline{y}^k)^T \begin{pmatrix} x-\overline{x}^k \\ y-\overline{y}^k \end{pmatrix} \le b_j - g_j(\overline{x}^k,\overline{y}^k) \\ \qquad\qquad\qquad\qquad\qquad\qquad \forall j \in I(\overline{x}^k,\overline{y}^k),\ k=0,\dots,i-1 \\ (\overline{x}^k-\hat{x}^k)^T(x-\overline{x}^k) \ge 0 \qquad\qquad\qquad\qquad \forall k \in K \\ x \in \mathbb{Z}^{n_1},\ y \in \mathbb{R}^{n_2},\ \alpha \in \mathbb{R} \end{cases}
$$

where $I(\overline{x},\overline{y}) = \{j : \text{either } g_j \text{ is convex or } g_j(\overline{x},\overline{y}) = b_j\}$ as defined in Section 3 and $K \subseteq \{1,\dots,i-1\}$ is the subset of iterations where $(\overline{x}^k,\overline{y}^k)$ is obtained by solving $(FP-NLP)^k$ and $\overline{x}^k \neq \hat{x}^k$. Note that $(OA)^i$ is an MILP.
The second problem is MINLP with x fixed:

$$(NLP)^i \quad \begin{cases} \min f(\hat{x}^i,y) \\ \text{s.t. :} \\ g(\hat{x}^i,y) \le b \\ y \in \mathbb{R}^{n_2}. \end{cases}$$

Note that $(NLP)^i$ is a nonlinear program.
The third one is $(FP-NLP)^i$ as defined in Section 2. Recall that this is a nonlinear program.
The fourth one is the following MILP which looks for a better solution than the best found so far:

$$(FP-OA)^i \quad \begin{cases} \min \|x-\overline{x}^{i-1}\|_1 \\ \text{s.t. :} \\ \nabla f(\overline{x}^k,\overline{y}^k)^T \begin{pmatrix} x-\overline{x}^k \\ y-\overline{y}^k \end{pmatrix} \le -f(\overline{x}^k,\overline{y}^k) + z^U - \delta \\ \qquad\qquad\qquad\qquad\qquad\qquad k = 0,\dots,i-1 \\ \nabla g_j(\overline{x}^k,\overline{y}^k)^T \begin{pmatrix} x-\overline{x}^k \\ y-\overline{y}^k \end{pmatrix} \le b_j - g_j(\overline{x}^k,\overline{y}^k) \\ \qquad\qquad\qquad\qquad \forall j \in I(\overline{x}^k,\overline{y}^k),\ k = 0,\dots,i-1 \\ (\overline{x}^k - \hat{x}^k)^T(x-\overline{x}^k) \ge 0 \qquad\qquad \forall k \in K \\ x \in \mathbb{Z}^{n_1},\ y \in \mathbb{R}^{n_2} \end{cases}$$

where z^U is the current upper bound on the value of MINLP and $\delta > 0$ is a small value indicating the desired improvement in objective function value. As in $(OA)^k$, K denotes the subset of iterations where $(\overline{x}^k,\overline{y}^k)$ is obtained by solving $(FP-NLP)^k$ and $\overline{x}^k \neq \hat{x}^k$.

The overall algorithm is given in Figure 3. A cursory description of its steps is as follows: Initially, we solve the continuous relaxation of MINLP to optimality to obtain a starting point $(\overline{x}^0,\overline{y}^0)$. Then, we compute the optimum $(\hat{x}^1,\hat{y}^1)$ of $(OA)^1$ where $K = \emptyset$. Similarly, at subsequent iterations, we obtain $(\hat{x}^i,\hat{y}^i)$ and a lower bound $\hat{\alpha}^i$ on the value of MINLP by solving $(OA)^i$. We then solve $(NLP)^i$ with x fixed at $\hat{x}^i$. If $(NLP)^i$ is feasible, then $(\overline{x}^i,\overline{y}^i)$ given by $\overline{x}^i = \hat{x}^i$ and $\overline{y}^i$ is the optimal solution of $(NLP)^i$. Moreover, $f(\overline{x}^i,\overline{y}^i)$ is an upper bound on the optimal solution of MINLP. Otherwise, $(NLP)^i$ is infeasible and we perform at least one iteration of the Feasibility Pump. More precisely, FP starts by solving $(FP-NLP)^i$, obtaining the point $(\overline{x}^i,\overline{y}^i)$. Then $(\hat{x}^{i+1},\hat{y}^{i+1})$ is obtained by solving $(FP-OA)^{i+1}$. Additional iterations of the Feasibility Pump are possibly performed, solving alternatively $(FP-NLP)$ and $(FP-OA)$ until either a better feasible solution of MINLP is found, or a proof is obtained that no such solution exists, or some iteration or time limit is reached (five iterations and at most two minutes CPU time are used in the experiments). If no better feasible solution of MINLP exists, the algorithm terminates. In the other two cases, a sequence of points $(\bar{x}^k, \bar{y}^k)$ is generated for $k = i, \dots, l-1$. In the case where a better feasible solution $(\bar{x}^{l-1}, \bar{y}^{l-1})$ is found by $(FP-NLP)^{l-1}$, i.e. $(FP-NLP)^{l-1}$ has objective value zero, we solve $(NLP)^{l-1}$ to check whether there exists a better solution than $(\bar{x}^{l-1}, \bar{y}^{l-1})$ with respect to the original objective function f. If we find such an improved solution we replace $\bar{y}^{l-1}$ by this solution. In both cases, the algorithm reverts to solving $(OA)^l$ with all the outer approximation constraints generated for the points $\{(\bar{x}^i, \bar{y}^i), \dots, (\bar{x}^{l-1}, \bar{y}^{l-1})\}$. The algorithm continues iterating between the four problems as described above. The algorithm terminates when the lower bound given by (OA) and the best upper bound found are equal within a specified tolerance ε.

$z^U := +\infty$;
$z^L := -\infty$;
$(\bar{x}^0, \bar{y}^0) :=$ optimal solution of the continuous relaxation of $MINLP$;
$K := \emptyset$;
$i := 1$;
Choose convergence parameters ε and δ
while $z^U - z^L > \varepsilon$ **do**
  Let $(\hat{\alpha}^i, \hat{x}^i, \hat{y}^i)$ be the optimal solution of $(OA)^i$;
  $z^L := \hat{\alpha}^i$;
  if $(NLP)^i$ is feasible;
   then
    Let $\bar{x}^i := \hat{x}^i$ and $\bar{y}^i$ be the optimal solution to $(NLP)^i$;
    $z^U := \min(z^U, f(\bar{x}^i, \bar{y}^i))$;
    $i := i+1$;
   else
    Let $(\bar{x}^i, \bar{y}^i)$ be the optimal solution of $(FP-NLP)^i$;
    $l := i+1$
    while $\bar{x}^{l-1} \neq \hat{x}^{l-1} \wedge l \leq i+5 \wedge$
     time in this FP < 2 minutes **do**
     if $(FP-OA)^l$ is feasible
      then
       Let $(\hat{x}^l, \hat{y}^l)$ be the optimal solution of $(FP-OA)^l$;
       Let $(\bar{x}^l, \bar{y}^l)$ be the optimal solution of $(FP-NLP)^l$;
       if $\bar{x}^l = \hat{x}^l$
        then
         replace $\bar{y}^l$ by the optimal solution of $(NLP)^l$;
         $z^U := \min(z^U, f(\bar{x}^l, \bar{y}^l))$;
        else
         $K := K \cup \{l\}$;
       fi
       $l := l+1$;
      else z^U is optimal, **exit**;
     fi
    od
    $i := l$
  fi
od

Fig. 3 Enhanced Outer Approximation algorithm.

The integration of Feasibility Pump into the Outer Approximation algorithm enhances the behavior of OA for instances with convex feasible sets defined by nonconvex constraints. This is shown by the following example (see also Figure 4).

Example 4

$$\begin{cases} \min -y \\ \text{s.t. :} \\ y - \sin\left(\frac{5\pi}{3}x\right) \leq 0 \\ -y - \sin\left(\frac{5\pi}{3}x\right) \leq 0 \\ x \in \{0,1\} \\ y \in \mathbb{R}. \end{cases}$$

Although the constraints of the above problem are nonconvex, the feasible region

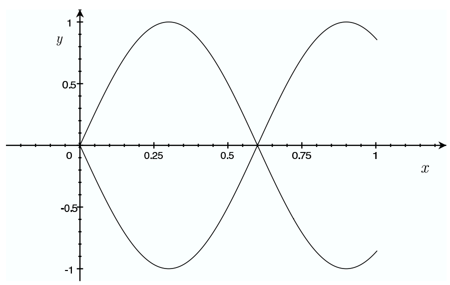

Fig. 4 Illustration of Example 4.

of the continuous relaxation is convex. Solving the continuous relaxation gives the point $(\bar{x}^0, \bar{y}^0) = (0.3, 1)$. The first outer approximation problem, namely $(OA)^1 = \max\{y : y \leq 1,\ x \in \{0,1\}\}$, has solution $(\hat{x}^1, \hat{y}^1) = (1,1)$. At this point, the classical OA tries to minimize the violation of the constraints over the line $x = 1$ selecting the point $(1,0)$. Then it adds the following two constraints

$$-\frac{5\pi}{6}x + y \leq -\frac{5\pi}{6} - \frac{\sqrt{3}}{2} \qquad \text{and} \qquad -\frac{5\pi}{6}x - y \leq -\frac{5\pi}{6} - \frac{\sqrt{3}}{2}$$

which make the corresponding MILP infeasible. This occurs because the point $(1,0)$ is taken outside of the feasible region of the continuous relaxation and, since the constraints are not defined by convex functions, generating the supporting hyperplanes at this point induces invalid constraints. On the other hand, the OA enhanced by $FP - NLP$ warm starting from $(\hat{x}^1, \hat{y}^1)$ solves $(FP - NLP)^1$ finding the closest NLP feasible point $(\bar{x}^1, \bar{y}^1) = (0.6, 0)$ to $\hat{x}^1$ in L_2-norm. Starting from such a point $(FP - OA)^2$ converges to $(\hat{x}^2, \hat{y}^2) = (0,0)$ which is integer and NLP feasible (as proved by the next $FP - NLP$ iteration).

□

The behavior outlined by the previous example is generalized by the following theorem.

Theorem 3 *Consider an MINLP with convex objective function and convex region $g(x,y) \leq b$. Assume that the integer variables are bounded. If the constraint qualification holds at every optimal solution of $(NLP)^i$, the modified version of OA converges to an optimal solution or proves that none exists even when the convex region $g(x,y) \leq b$ is defined by nonconvex constraints.*

Proof: If $\hat{x}^i \neq \hat{x}^k$ for all i,k with $i \neq k$ then the algorithm terminates in a finite number of iterations since the integer variables are bounded. Furthermore the algorithm finds an optimum solution of MINLP or proves that none exists since the constraints of $(OA)^i$ and $(FP-OA)^i$ are valid outer approximations of MINLP.

Now suppose that $(\hat{x}^i,\hat{y}^i)$ satisfies $\hat{x}^i = \hat{x}^k$ for some $k < i$. We consider two cases. First suppose that $\hat{x}^k$ was obtained by solving $(OA)^k$. Then we claim that $(NLP)^k$ is feasible, for suppose not. Then the algorithm calls the Feasibility Pump. Since $(NLP)^k$ is infeasible, the first iteration of FP finds a point $\overline{x}^k \neq \hat{x}^k$. Therefore, $k \in K$ and constraint $(\overline{x}^k - \hat{x}^k)^T(x-\hat{x}^k) \geq 0$ has been added. This implies that $\hat{x}^k$ can never be repeated, a contradiction. This proves the claim. Second, suppose that $\hat{x}^k$ was obtained by solving $(FP-OA)^k$. Then $\overline{x}^k$ is obtained by solving $(FP-NLP)^k$. If $\overline{x}^k \neq \hat{x}^k$ then, by the argument used above, $\hat{x}^k$ cannot be repeated, a contradiction. Therefore $\overline{x}^k = \hat{x}^k$ which implies that $(NLP)^k$ is feasible. Thus in both cases $(NLP)^k$ is feasible, $\overline{x}^k = \hat{x}^k$ and $\overline{y}^k$ is the optimum solution of $(NLP)^k$.

Since $\overline{y}^k$ is optimal for $(NLP)^k$ and the constraint qualification holds, $\overline{y}^k$ satisfies the KKT conditions:

$$
\begin{aligned}
&\nabla_y f(\overline{x}^k,\overline{y}^k) = -\sum_{j=1}^{m} \lambda_j \nabla_y g_j(\overline{x}^k,\overline{y}^k) \\
&\lambda_j \left(g_j(\overline{x}^k,\overline{y}^k) - b_j\right) = 0 \qquad\qquad \forall j = 1,\ldots,m \\
&\lambda \geq 0
\end{aligned}
$$

where $\nabla_y g_j$ denotes the gradient of g_j with respect to the y variables.
Since $i > k$ and $(\hat{x}^i,\hat{y}^i)$ is feasible for $(OA)^i$ or $(FP-OA)^i$ the following outer approximation constraints are satisfied by $(\hat{x}^i,\hat{y}^i)$:

$$
\nabla g_j(\overline{x}^k,\overline{y}^k)^T \begin{pmatrix} \hat{x}^i - \overline{x}^k \\ \hat{y}^i - \overline{y}^k \end{pmatrix} \leq b_j - g_j(\overline{x}^k,\overline{y}^k) \quad \forall j \in I(\overline{x}^k,\overline{y}^k).
$$

Since $\hat{x}^i = \overline{x}^k$ we have

$$
\nabla_y g_j(\overline{x}^k,\overline{y}^k)^T \left(\hat{y}^i - \overline{y}^k\right) \leq b_j - g_j(\overline{x}^k,\overline{y}^k) \quad \forall j \in I(\overline{x}^k,\overline{y}^k).
$$

Using the above KKT conditions, this implies that

$$
\nabla_y f(\overline{x}^k,\overline{y}^k)^T \left(\hat{y}^i - \overline{y}^k\right) \leq 0. \tag{8}
$$

There are now two possible cases. First, if $(\hat{x}^i,\hat{y}^i)$ is a solution of $(OA)^i$, it satisfies the inequality

$$
\nabla f(\overline{x}^k,\overline{y}^k)^T \begin{pmatrix} 0 \\ \hat{y}^i - \overline{y}^k \end{pmatrix} - \hat{\alpha}^i \leq -f(\overline{x}^k,\overline{y}^k).
$$

Then equation (8) implies that $\alpha^i \geq f(\bar{x}^k, \bar{y}^k)$ showing that $(\bar{x}^k, \bar{y}^k)$ is an optimum solution, and therefore the algorithm terminates with the correct answer.

In the second case, $(\hat{x}^i, \hat{y}^i)$ is a solution of $(FP-OA)^i$. Then it satisfies

$$\nabla f(\bar{x}^k, \bar{y}^k)^T \begin{pmatrix} 0 \\ \hat{y}^i - \bar{y}^k \end{pmatrix} \leq -f(\bar{x}^k, \bar{y}^k) + z^U - \delta$$

and equation (8) implies that $z^U - \delta \geq f(\bar{x}^k, \bar{y}^k)$. This is a contradiction to the facts that $(\bar{x}^k, \bar{y}^k)$ is feasible to MINLP and that z^U is the best known upper bound.

□

Computational experiments. In Table 4 we report computational results comparing the OA with the enhanced OA coupled with FP (*OA+FP*). The latter is implemented as a modification of the OA algorithm implemented in [1], which is used as the OA code. Our procedure is set as follows. We start by performing one minute of iterated Feasibility Pump in order to find a good solution. We then start the enhanced OA algorithm. Since the primary goal of the FP in the enhanced OA is to quickly find improved feasible solutions, we put a limit of two minutes and five iterations for each call to the FP inside the enhanced OA.

	OA+FP				OA			
Name	ub	tub	lb	tlb	ub	tub	lb	tlb
CLay0304M	40262.4	79	*	82	40262.4	12	*	14
CLay0305H	8092.5	4	*	32	8092.5	24	*	24
CLay0305M	8092.5	4	*	24	8092.5	75	*	75
fo7_2	17.75	4	*	103	17.75	20	*	128
fo7	20.73	260	*	260	20.73	24	*	197
fo8	22.38	573	*	835	22.38	727	*	906
fo9	23.46	1160	*	2613	23.46	5235	*	6024
o7_2	116.94	189	*	2312	118.86	5651	114.08	7200
o7	131.64	5	*	6055	—	—	122.79	7200
RSyn0830M02M	-730.51	12	*	178	-730.51	3837	*	5272
RSyn0830M03M	-1543.06	52	*	1018	-1538.91	5933	-1548.46	7200
RSyn0830M04M	-2520.88	46	-3067.54	7200	-2502.39	5697	-3216.91	7200
RSyn0840M02M	-734.98	1383	*	1383	-734.98	1846	*	1846
RSyn0840M03M	-2742.65	3418	*	3418	-2734.53	7200	-2789.93	7200
RSyn0840M04M	-2556.60	42	-2638.63	7200	-2488.87	7200	-3599.77	7200
SLay10M	129580	1778	*	3421	129580	336	128531	7200
trimloss4	8.3	10	*	423	8.3	785	*	785
trimloss5	10.7	485	3.3	7200	—	—	5.9	7200
trimloss6	16.5	2040	3.5	7200	—	—	6.5	7200
trimloss7	27.5	387	2.6	7200	—	—	3.3	7200
trimloss12	—	—	5.4	7200	—	—	9.6	7200

Table 4 Comparison between OA and its enhanced version on a subset of diffi cult instances. Columns labeled "ub" and "lb" report the upper and lower bound values; columns labeled "tub" and "tlb" give the CPU time in seconds for obtaining those upper and lower bounds; symbol "*" denotes proven optimality and "−" indicates that no solution is found.

The results of Table 4 show that OA+FP can solve 15 instances whereas the classical OA algorithm solves only 10 within the 2-hour time limit. Furthermore, OA+FP finds a feasible solution in all but one instance, whereas the classical OA algorithm fails to find a feasible solution in 5 cases. In addition to being more robust, OA+FP is also competitive in terms of computing time, on most instances.

Acknowledgements We would like to thank all the members of the IBM-CMU working group on MINLP (Larry Biegler, Andy Conn, Ignacio Grossmann, Carl Laird, Jon Lee, Nick Sawaya and Andreas Waechter) for numerous discussions on this topic. Special thanks to Andy Conn for his helpful comments on an earlier draft of the paper.

References

1. P. Bonami, A. Wächter, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot and N.W. Sawaya. An algorithmic framework for convex mixed integer nonlinear programs. *Technical Report* RC23771, IBM T.J. Watson Research Center, 2005.
2. COIN-OR. `www.coin-or.org`
3. `Dicopt`. `http://www.gams.com/solvers/dicopt/main.htm`
4. M. Duran and I.E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.
5. A.V. Fiacco, G.P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques, *Wiley & Sons*, 1968; republished 1990, SIAM Philadelphia.
6. M. Fischetti, F. Glover and A. Lodi. The Feasibility Pump. *Mathematical Programming*, 104:91–104, 2004.
7. R. Fletcher and S. Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66:327–349, 1994.
8. N.W. Sawaya, C.D. Laird and P. Bonami. A novel library of nonlinear mixed-integer and generalized disjunctive programming problems. In preparation, 2006.
9. `Sbb`. `http://www.conopt.com/sbb/SBB_announcement.htm`