Penalty Alternating Direction Methods for Mixed-Integer Optimization: A New View on Feasibility Pumps

Björn Geissler[1], Antonio Morsi[1], Lars Schewe[1], Martin Schmidt[2]

Abstract. Feasibility pumps are highly effective primal heuristics for mixed-integer linear and nonlinear optimization. However, despite their success in practice there are only few works considering their theoretical properties. We show that feasibility pumps can be seen as alternating direction methods applied to special reformulations of the original problem, inheriting the convergence theory of these methods. Moreover, we propose a novel penalty framework that encompasses this alternating direction method, which allows us to refrain from random perturbations that are applied in standard versions of feasibility pumps in case of failure. We present a convergence theory for the new penalty based alternating direction method and compare the new variant of the feasibility pump with existing versions in an extensive numerical study for mixed-integer linear and nonlinear problems.

Due to their practical relevance, mixed-integer nonlinear problems (MINLPs) form a very important class of optimization problems. One important part of successful algorithms for the solution of such problems is finding feasible solutions quickly. For this, typically heuristics are employed. These can be roughly divided into heuristics that improve known feasible solutions (e.g., local branching [24] or RINS [16]) and heuristics that construct feasible solutions from scratch. This article discusses a heuristic of the latter type: The algorithm of interest in this article is the so-called *feasibility pump* that has originally been proposed by Fischetti et al. in [23] for MIPs and that has been extended by many other researchers, e.g., in [1–3, 6, 7, 17, 18, 25, 33]. In addition, feasibility pumps have also been applied to MINLPs during the last years; see, e.g., [4, 8, 9, 14, 15, 38, 39]. A more detailed review of the literature about feasibility pumps is given in Section 1. For a comprehensive overview over primal heuristics for mixed-integer linear and nonlinear problems in general, we refer the interested reader to Berthold [4, 5] and the references therein.

In a nutshell, feasibility pumps work as follows: given an optimal solution of the continuous relaxation of the problem, the methods construct two sequences. The first one contains integer-feasible points, the second one contains points that are feasible w.r.t. the continuous relaxation. Thus, one has found an overall feasible point if these sequences converge to a common point. To escape from situations where the construction of the sequences gets stuck and thus do not converge to a common point, feasibility pumps usually incorporate randomized restarts.

The feasibility pumps described in the literature are difficult to analyze theoretically due to the use of random perturbations. These random perturbations are, however, crucial to the practical performance of the methods. The main object of the existing theoretical analysis is the *idealized feasibility pump*, i.e., the method without random perturbations. This is the method analyzed in the publications [17]

Date: April 7, 2016.

2010 *Mathematics Subject Classification.* 65K05, 90-08, 90C10, 90C11, 90C59.

Key words and phrases. Mixed-Integer Nonlinear Optimization, Mixed-Integer Linear Optimization, Feasibility Pump, Alternating Direction Methods, Penalty Methods.

and [6]. To be more specific, De Santis et al. show in [17] that idealized feasibility pumps are a special case of the Frank–Wolfe algorithm applied to a suitable chosen concave and nonsmooth merit function and Boland et al. showed in [6] that idealized feasibility pumps can be seen as discrete versions of the proximal point algorithm.

Our approach is similar to these publications. We show that the idealized variant can be seen as a so-called *alternating direction method* (ADM) applied to a special reformulation of the mixed-integer problem at hand. To this end, we extend the known theory on feasibility pumps by applying the convergence theory of general ADMs. We then go one step further: The necessity to use random perturbations comes from the need to escape from undesired points. We replace these random perturbations of the original feasibility pump by a penalty framework. This allows us to view feasibility pumps as penalty based alternating direction methods—a new class of optimization methods for which we also present convergence theory. In summary, we are able to give a convergence theory for a class of feasibility pumps that incorporates deterministic restart rules. Another advantage is that our method can be presented in a quite generic way that comprises both the case of mixed-integer linear and nonlinear problems.

We further give extensive computational results to show that our replacement of the random restarts does not lead to a degradation in performance. Our method compares favorably with published variants of feasibility pumps for MIPs and MINLPs.

The paper is organized as follows: In Section 1 we review the main ingredients of feasibility pumps and give a more detailed literature survey. Afterward, we discuss general ADMs in Section 2 and show that idealized feasibility pumps can be seen as ADMs applied to certain equivalent reformulations of the original problem. In Section 3, we then present a penalty ADM, prove convergence results, and show how this new method can be used to obtain a novel feasibility pump algorithm that replaces random restarts with penalty parameter updates. In Section 4 we discuss important implementation issues and Section 5 finally presents an extensive computational study both for MIPs and MINLPs.

1. Feasibility Pumps

In this section we give an overview over feasibility pump algorithms for mixed-integer linear problems (MIPs) as well as for mixed-integer nonlinear problems (MINLPs). We start with the MIP case in Section 1.1 and afterward discuss generalizations for nonlinear problems in Section 1.2. General surveys on this topic can be found in Berthold [4] and Bonami et al. [10].

1.1. Feasibility Pumps for Mixed-Integer Linear Problems.

The feasibility pump has been introduced by Fischetti et al. [23] for binary MIPs and has been extended to general MIPs by Bertacco et al. [3]. The goal of the feasibility pump is to find a feasible point of a mixed-integer linear problem of the general form

$$\min_x \quad c^\top x \tag{1a}$$

$$\text{s.t.} \quad Ax \geq b, \tag{1b}$$

$$x_i \in \mathbb{Z} \quad \text{for all } i \in I, \tag{1c}$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, and $\emptyset \neq I \subseteq \{1, \dots, n\}$. Moreover, we assume that variable bounds $l \leq x \leq u$ with $-\infty < l_i \leq u_i < \infty$ for all $i \in I$ are part of $Ax \geq b$. We refer to the polyhedron of the LP relaxation of (1) by P, i.e., $P = \{x \in \mathbb{R}^n : Ax \geq b\}$. Throughout the paper we assume that $P \neq \emptyset$. The main idea of the feasibility pump is to create two sequences $(\bar{x}^k)$ and $(\tilde{x}^k)$ such that $\bar{x}^k \in P$ and $\tilde{x}^k$ is integer feasible, i.e., $\tilde{x}_i^k \in \mathbb{Z}$ for all $i \in I$. In addition, the

Algorithm 1 The basic feasibility pump for 0-1-MIPs

```
1: Compute x̄^0 ∈ argmin{c^⊤x : x ∈ P}.
2: if x̄^0 is integer feasible then
3:     return x̄^0
4: end if
5: Set x̃^0 = ⌈x̄^0⌋ and k ← 0.
6: while not termination condition do
7:     Compute x̄^{k+1} ∈ argmin{‖x_I − x̃_I^k‖_1 : x ∈ P}.
8:     if x̄^{k+1} is integer feasible then
9:         return x̄^{k+1}
10:    end if
11:    Set x̃^{k+1} = ⌈x̄^{k+1}⌋.
12:    if algorithm stalls or cycles then
13:        perturb x̃^{k+1}
14:    end if
15:    Set k ← k + 1.
16: end while
```

construction of the sequences is tailored to minimize the distance of the pairs $\bar{x}^k$ and $\tilde{x}^k$. The algorithm terminates after a given time, after an iteration limit has been reached, or if the distance is zero, i.e., $\bar{x}^k = \tilde{x}^k$. In the latter case, the algorithm terminates with an MIP-feasible point, i.e., a point that is both in P and integer feasible. We now describe the method in detail for the case of 0-1-MIPs, i.e., we replace $x_i \in \mathbb{Z}$ by $x_i \in \{0,1\}$ in (1c). The initial point $\bar{x}^0$ is computed to be an optimal solution of the LP relaxation of (1), i.e., $\bar{x}^0 \in \operatorname{argmin}\{c^\top x : x \in P\}$, and the initial point $\tilde{x}^0$ of the other sequence is the rounding $\lceil \bar{x}^0 \rfloor$ of the integer components of $\bar{x}^0$. Note that the rounding operator $\lceil \cdot \rfloor$ only rounds integer components, i.e., $\lceil x_i \rfloor = x_i$ for all $i \notin I$. From then on, in each iteration k the new iterate $\bar{x}^{k+1}$ is the nearest point (w.r.t. the integer components) to $\tilde{x}^k$ in the ℓ_1 norm, i.e.,

$$\bar{x}^{k+1} \in \operatorname{argmin}\{\|x_I - \tilde{x}_I^k\|_1 : x \in P\}$$

and $\tilde{x}^{k+1} := \lceil \bar{x}^{k+1} \rfloor$. Here and in what follows, x_I denotes the sub-vector of x only consisting of the components indicated by the index set I. After every rounding step a cycle and a stalling test decides whether a random perturbation of the integer part of $\tilde{x}^k$ is applied. The details can be found in Bertacco et al. [3] and Fischetti et al. [23]. A formal listing of the basic feasibility pump for binary MIPs is given in Algorithm 1. In what follows, Line 7 of Algorithm 1 is referred to as the *projection step* and Line 11 is called the *rounding step*. Note that the projection step can be written as a linear program by reformulating the ℓ_1 norm objective as

$$\|x_I - \tilde{x}_I\|_1 = \sum_{i \in I : \tilde{x}_i = 0} x_i + \sum_{i \in I : \tilde{x}_i = 1} (1 - x_i).$$

This is also possible for general MIPs but then requires the introduction of auxiliary variables and constraints; see Bertacco et al. [3] for the details. The original feasibility pump is very successful in quickly finding feasible solutions. However, these solutions are often of minor quality. Thus, improvements of the original version with the goal of developing variants of the feasibility pump that are comparable in run time as well as success rate and provide solutions of better quality are studied in many publications. Fischetti and Salvagnin [25] improved the rounding step by replacing the simple rounding $\tilde{x}^k = \lceil \bar{x}^k \rfloor$ with a procedure based on constraint propagation. Other strategies of improving the rounding step are given in Baena and Castro [2], where simple rounding is replaced with rounding of different candidate points on a line segment between the solution of the projection step and the analytic center of the LP polyhedron, and in Boland et al. [7], where an integer line search is applied to find integer feasible points that are closer to P than the points achieved by simple rounding. In contrast to these approaches, Achterberg and Berthold [1] improved the projection step in order to achieve feasible solutions of better quality by replacing the ℓ_1 norm objective $\|x_I - \tilde{x}_I\|_1$ with a convex combination of this distance measure and the original objective function:

$$(1-\alpha)\|x_I - \tilde{x}_I\|_1 + \alpha\beta c^\top x.$$

Here, $\alpha \in [0,1]$ is the convex combination parameter and $\beta \in \mathbb{R}_{>0}$ is a problem data depending scaling parameter; see [1] for the details. None of the papers cited so far contains any theoretical results on the feasibility pump. This situation changed with a remark in Eckstein and Nediak [22] noticing that the idealized feasibility pump for binary MIPs can be interpreted as a Frank–Wolfe algorithm (see Frank and Wolfe [26]) applied to the minimization of a concave and nonsmooth objective function over a polyhedron. De Santis et al. [17] seized this idea and proved this correspondence, yielding the first theoretical result for the feasibility pump. To be more precise, they used the result shown by Mangasarian in [35] that the Frank–Wolfe algorithm applied to the above given situation terminates after a finite number of iterations and returns a so-called *vertex stationary point* of the problem. We discuss the relation of this result with our results in more detail in Section 3. In [18], De Santis et al. generalized their results to general MIPs. Another theoretical result is presented by Boland et al. in [6], where it is shown that the idealized feasibility pump can be interpreted as a discrete version of the proximal point algorithm. We remark that both cited theoretical investigations only consider the case of idealized feasibility pumps, i.e., the variants of feasibility pumps without random perturbations used to handle cycling or stalling issues.

1.2. **Feasibility Pumps for Mixed-Integer Nonlinear Problems.** We now turn to feasibility pumps for mixed-integer nonlinear problems of the form

$$\min_x \quad f(x) \tag{2a}$$
$$\text{s.t.} \quad h(x) \geq 0, \tag{2b}$$
$$x_i \in \mathbb{Z} \cap [l_i, u_i] \quad \text{for all } i \in I, \tag{2c}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the constraints function $h : \mathbb{R}^n \to \mathbb{R}^p$ are continuous. The feasible set of the NLP relaxation is denoted by $\Omega^\mathrm{r} := \{x \in \mathbb{R}^n : h(x) \geq 0\}$. Again, we assume that $\Omega^\mathrm{r} \neq \emptyset$ holds and that Ω^r is compact. Lastly, we assume that all bounds on the discrete variables are finite, i.e., $-\infty < l_i \leq u_i < \infty$ for all $i \in I$. The MINLP (2) is said to be convex if f and $-h$ are convex.

For convex problems, the direct generalization of the feasibility pump for MIPs to MINLPs is given in Bonami and Gonçalves [9]: the projection step LP is replaced by an NLP and the rounding step stays the same, i.e., $\tilde{x}^{k+1} := \lceil \bar{x}^{k+1} \rfloor$. Cycling and stalling issues are again handled by random perturbations of the integer components of x. Variants of this method are then deduced by modifying the rounding step and by replacing the ℓ_1 with the ℓ_2 norm in the projection step. The feasibility pump for convex MINLP proposed in Bonami et al. [8] significantly differs from the version in [9] because it replaces the simple rounding step by a MIP relaxation of (2) that is successively tightened by adding outer approximation cuts (see Duran and Grossmann [21]) based on the NLP feasible solutions obtained by solving the ℓ_2 norm projection steps. Bonami et al. study two versions of their algorithm; a basic and an enhanced one. For their basic version it is shown that it cannot cycle if the LICQ holds for the NLP relaxation. For their enhanced version it is shown that it cannot cycle and that it is an exact method for solving convex MINLPs if all integer variables are bounded.

For nonconvex MINLPs the convergence results of Bonami et al. [8] do not hold because the outer approximation cuts cannot be applied to the nonconvex NLP relaxation and because the projection step is now a nonconvex problem, which is too hard to be solved to global optimality in general. The article D'Ambrosio et al. [15] is the first presentation of a feasibility pumps for nonconvex MINLPs. The above mentioned issues are "resolved" by solving the nonconvex projection step NLP via a multistart heuristic using local NLP solvers and the rounding step is realized by a MIP using outer approximation cuts if they are globally valid for the nonconvex problem. In the subsequent paper [14], D'Ambrosio et al. interpreted feasibility pumps for general nonconvex MINLPs as variants of successive projection methods (SPMs). Despite their strong similarity, the authors observe that typical feasibility pumps do not fall exactly into the class of SPMs, which is why their convergence theory is not applicable.

Finally, a generalization of the objective feasibility pump of Achterberg and Berthold [1] is given in [38, 39] for convex MINLP and Berthold [4] discusses some new algorithmic ideas for nonconvex MINLPs.

2. Alternating Direction Methods

In this section, we first briefly review classical alternating direction methods (ADMs) and afterward prove that idealized feasibility pumps, i.e., the basic feasibility pump algorithm 1 without random perturbations, can be seen as a special case of alternating direction methods. This gives new theoretical insights since the complete theory of ADMs can be applied to idealized feasibility pumps.

To this end, we consider the general problem

$$\min_{x,y} \quad f(x,y) \tag{3a}$$

$$\text{s.t.} \quad g(x,y) = 0, \quad h(x,y) \geq 0, \tag{3b}$$

$$x \in X, \quad y \in Y, \tag{3c}$$

for which we make the following assumption:

Assumption 1. *The objective function $f : \mathbb{R}^{n_x+n_y} \to \mathbb{R}$ and the constraint functions $g : \mathbb{R}^{n_x+n_y} \to \mathbb{R}^m$, $h : \mathbb{R}^{n_x+n_y} \to \mathbb{R}^p$ are continuous and the sets X and Y are non-empty and compact.*

The feasible set is denoted by Ω, i.e.,

$$\Omega = \{(x,y) \in X \times Y : g(x,y) = 0,\, h(x,y) \geq 0\} \subseteq X \times Y,$$

and the corresponding projections onto X and Y are denoted by Ω_X and Ω_Y, respectively. Classical alternating direction methods are extensions of Lagrangian methods and have been originally proposed in Gabay and Mercier [27] and Glowinski and Marroco [29]. More recently, ADM-type methods have seen a resurgence; see, e.g., [11] for a general overview and [28] for an application of ADMs to nonconvex MINLPs from gas transport including heat power constraints. The latter application also provided the motivation for this article. ADMs solve Problem (3) by solving two simpler problems: Given an iterate (x^k, y^k) they solve Problem (3) for y fixed to y^k into the direction of x, yielding a new x-iterate x^{k+1}. Afterward, x is fixed to x^{k+1} and Problem (3) is solved into the direction of y, yielding a new y-iterate y^{k+1}. A formal listing is given in Algorithm 2. If the optimization problem in Line 3 or Line 4 of Algorithm 2 has a unique solution for all k, it is known that ADMs

Algorithm 2 A Standard Alternating Direction Method

1: Choose initial values $(x^0, y^0) \in X \times Y$.
2: **for** $k = 0, 1, \dots$ **do**
3: Compute
$$x^{k+1} \in \operatorname*{argmin}_x \{f(x, y^k) : g(x, y^k) = 0,\ h(x, y^k) \geq 0, x \in X\}.$$
4: Compute
$$y^{k+1} \in \operatorname*{argmin}_y \{f(x^{k+1}, y) : g(x^{k+1}, y) = 0,\ h(x^{k+1}, y) \geq 0, y \in Y\}.$$
5: Set $k \leftarrow k + 1$.
6: **end for**

converge to so-called *partial minima* of Problem (3), i.e., to points $(x^*, y^*) \in \Omega$ for which

$$\begin{aligned} f(x^*, y^*) &\leq f(x, y^*) \quad \text{for all } (x, y^*) \in \Omega, \\ f(x^*, y^*) &\leq f(x^*, y) \quad \text{for all } (x^*, y) \in \Omega \end{aligned}$$

holds; see Gorski et al. [30] for the following result:

Theorem 2.1. *Let* $\left\{(x^i, y^i)\right\}_{i=0}^\infty$ *be a sequence with* $(x^{i+1}, y^{i+1}) \in \Theta(x^i, y^i)$*, where*

$$\Theta(x^i, y^i) := \{(x^*, y^*) : \forall x \in X. f(x^*, y^i) \leq f(x, y^i);\ \forall y \in Y. f(x^*, y^*) \leq f(x^*, y)\}.$$

Suppose that Assumption 1 holds and that the solution of the first optimization problem is always unique. Then every convergent subsequence of $\left\{(x^i, y^i)\right\}_{i=0}^\infty$ *converges to a partial minimum. For two limit points* z, z' *of such subsequences it holds that* $f(z) = f(z')$*.*

Stronger results can be obtained if additional assumptions are made on f and $X \times Y$: If f is continuously differentiable, Algorithm 2 converges to a stationary point (in the classical sense of nonlinear optimization). If, in addition, f and $X \times Y$ are convex it is easy to show that partial minimizers are also global minimizers of Problem (3). For more details on the convergence theory of classical ADMs, see Gorski et al. [30] as well as Wendell and Hurter [40].

2.1. **Feasibility Pumps as ADMs.** Recall that the basic feasibility pump algorithm 1 tries to find a feasible solution for the binary variant of MIP (1). We now consider the idealized feasibility pump, i.e., we omit the perturbation step in Line 13 of Algorithm 1, and show that the idealized feasibility pump is a special case of the ADM algorithm 2 applied to a certain reformulation of MIP (1). To this end, we duplicate the variables x_I using the new variable vector $y \in \{0, 1\}^I$, yielding

$$\begin{aligned} \min_{x,y} \quad & c^\top x \\ \text{s.t.} \quad & x \in X := \{x \in \mathbb{R}^n : Ax \geq b, x_I \in [0, 1]^I\}, \\ & y \in Y := \{0, 1\}^I, \quad g(x, y) = x_I - y = 0, \end{aligned}$$

which is obviously equivalent to the original MIP. Note that, compared to the general problem (3), we do not explicitly require the inequality constraints vector h. The feasibility pump is only interested in feasibility and thus ignores the objective function. By deleting the objective from the reformulated model and instead moving an ℓ_1 penalty term of the coupling condition $y = x_I$ into the objective, we obtain

$$\min_{x,y} \quad \|x_I - y\|_1 \tag{4a}$$

$$\text{s.t.} \quad x \in X := \{x \in \mathbb{R}^n : Ax \geq b, x_I \in [0, 1]^I\}, \quad y \in Y := \{0, 1\}^I. \tag{4b}$$

If we define initial values (x^0, y^0) by $x^0 := \operatorname{argmin}\{c^\top x : x \in X\}$ and $y^0 := \lceil x^0 \rfloor$, it can be easily seen that solving Problem (4) with the ADM algorithm 2 exactly corresponds to the idealized feasibility pump algorithm. To be more precise, finding the new x-iterate within the ADM coincides with the projection step and finding the new y-iterate corresponds to the rounding step.

In the context of Algorithm 2, we say that the sequence of iterates z^k cycles if there exists an iteration k and an $l \geq 2$ with $z^k = z^{k+l}$. Next, we prove that the ADM cannot cycle and thus terminates after a finite number of iterations. To this end, we make the following observations: First, X and Y in (4) are non-empty and compact sets. Second, we can assume uniqueness of the rounding step by resolving tie-breaks choosing lexicographically minimal solutions. Thus, by using that norms are continuous, we have the following result.

Lemma 2.2. *Algorithm 2 does not cycle.*

Proof. Assume the contrary, i.e., there exists an iteration k and an $l \geq 2$ such that $z^k, z^{k+1}, \dots, z^{k+l} = z^k$. Since $f(x^{k+1}, y^{k+1}) \leq f(x^{k+1}, y^k) \leq f(x^k, y^k)$ holds for all iterations k we directly see that $f(z^k) = f(z^{k+1}) = \cdots = f(z^{k+l-1})$ holds. This, however, implies that z^k is already a partial minimum at which the algorithm stops. □

We note that this lemma is equivalent to Proposition 1 of De Santis et al. [17]. There, the authors show that the idealized feasibility pump for binary MIPs is equivalent to the Frank–Wolfe algorithm (using an unitary stepsize) applied to the problem

$$\min_{x \in P} \quad \sum_{i \in I} \min\{x_i, 1 - x_i\}, \tag{5}$$

where the objective function is a concave and nonsmooth merit function for measuring integrality. Applying the convergence theory from [35] then also yields finite termination at so-called *vertex stationary points*. Since we have now proven that the ADM algorithm 2 applied to (4) is equivalent to the above mentioned special case of the Frank–Wolfe method, we have also shown that partial minima of Problem (4) are exactly the vertex stationary points of Problem (5).

From the theory reported above and the last lemma we can directly deduce the following convergence theorem for the idealized feasibility pump.

Theorem 2.3. *The idealized feasibility pump terminates at a partial minimum (x^*, y^*) of Problem* (4) *after a finite number of iterations. If the partial minimum (x^*, y^*) has objective value $\|x_I^* - y^*\|_1 = 0$, the point (x^*, y^*) is feasible for the MIP* (1).

This theorem also gives us a new view of the random perturbation steps of feasibility pump algorithms: They can be interpreted as an attempt to escape from non-integral partial optima of Problem (4).

So far, we have only discussed the case of binary MIPs. However, our theory is still applicable as long as the optimization problems in Line 3 and 4 of Algorithm 2 are solved to global optimality. This is a realistic assumption for convex MINLPs of type (2). The suitable generalization of Problem (4) for this problem class reads

$$\min_{x,y} \quad \|x_I - y\|_1 \tag{6a}$$

$$\text{s.t.} \quad x \in X := \{x \in \mathbb{R}^n : h(x) \geq 0, x_I \in [0,1]^I\}, \quad y \in Y := \{0,1\}^I. \tag{6b}$$

We note that the resulting ADM is exactly the method presented in Bonami and Gonçalves [9]. Using the same techniques as above, we get the following convergence theorem for the idealized feasibility pump for convex MINLP.

Theorem 2.4. *The idealized feasibility pump for convex MINLP* (2) *is equivalent to the ADM algorithm 2 applied to Problem* (6)*. Thus, it terminates at a partial minimum* (x^*, y^*) *of Problem* (6) *after a finite number of iterations. If this partial minimum has objective value* $\|x_I^* - y^*\|_1 = 0$*, the point* (x^*, y^*) *is feasible for the convex MINLP* (2)*.*

We close this section with two remarks. First, we note that we presented the results in this section for binary MI(NL)Ps only for improving readability. The extension to general mixed-integer problems is straightforward. Second, we again want to highlight that the theoretical results presented in this section only hold if the optimization problems in Line 3 and 4 of Algorithm 2 are solved to global optimality. Since this is typically not possible for general nonconvex MINLPs, the results are only practically valid for convex mixed-integer problems.

3. The Penalty Alternating Direction Method

In this section we first present a new penalty alternating direction method in Section 3.1 and afterward prove the convergence results in Section 3.2. Finally, in Section 3.3 we show how the new method can be used to obtain a novel feasibility pump algorithm for general mixed-integer optimization. This new penalty alternating direction method based feasibility pump replaces random perturbations with a theoretically analyzable penalty framework for escaping undesired intermediate points. Thus, the complete theory presented for the new method also applies to the new feasibility pump variant.

3.1. The Algorithm.

We now present the novel weighted ℓ_1 penalty method based on the classical ADM framework given in Section 2. To this end, we define the ℓ_1 penalty function

$$\phi_1(x, y; \mu, \rho) := f(x, y) + \sum_{i=1}^{m} \mu_i \, |g_i(x, y)| + \sum_{i=1}^{p} \rho_i [h_i(x, y)]^-,$$

where

$$[\alpha]^- := \begin{cases} 0, & \text{if } \alpha \ge 0, \\ -\alpha, & \text{if } \alpha < 0, \end{cases}$$

and $\mu = (\mu_i)_{i=1}^m, \rho = (\rho_i)_{i=1}^p \ge 0$ are the penalty parameters for the equality and inequality constraints. Note that we allow for different penalty parameters for the constraints instead of a single penalty parameter as it is often the case for penalty methods.

The penalty ADM now proceeds as follows. Given a starting point and initial values for all penalty parameters, the alternating direction method of Algorithm 2 is used to compute a partial minimum of the penalty problem

$$\min_{x,y} \quad \phi_1(x, y; \mu, \rho) \quad \text{s.t.} \quad x \in X, y \in Y. \tag{7}$$

Afterward, the penalty parameters are updated and the next penalty problem is solved to partial minimality. Thus, the algorithm produces a sequence of partial minima of a sequence of penalty problems of type (7). More formally, the method is specified in Algorithm 3.

3.2. Convergence Theory.

We now present the convergence results for the penalty ADM algorithm 3. We start by proving that partial minima of the penalty problems are partial minima of the original problem if they are feasible.

Lemma 3.1. *Assume that* (x^*, y^*) *is a partial minimum of* $\phi_1(x, y; \mu, \rho)$ *for arbitrary but fixed* $\mu, \rho \ge 0$ *and let* (x^*, y^*) *be feasible for Problem* (3)*. Then* (x^*, y^*) *is a partial minimum of Problem* (3)*.*

Algorithm 3 The ℓ_1 Penalty Alternating Direction Method

1: Choose initial values $(x^{0,0}, y^{0,0}) \in X \times Y$ and penalty parameters $\mu^0, \rho^0 \geq 0$.
2: **for** $k = 0, 1, \dots$ **do**
3: Set $l = 0$.
4: **while** $(x^{k,l}, y^{k,l})$ is not a partial minimum of (7) with $\mu = \mu^k$ and $\rho = \rho^k$ **do**
5: Compute

$$x^{k,l+1} \in \operatorname*{argmin}_x \{\phi_1(x, y^{k,l}; \mu^k, \rho^k) : x \in X\}.$$

6: Compute

$$y^{k,l+1} \in \operatorname*{argmin}_y \{\phi_1(x^{k,l+1}, y; \mu^k, \rho^k) : y \in Y\}.$$

7: Set $l \leftarrow l + 1$.
8: **end while**
9: Choose new penalty parameters $\mu^{k+1} \geq \mu^k$ and $\rho^{k+1} \geq \rho^k$.
10: **end for**

Proof. Let $x \in X$ such that (x, y^*) is feasible for Problem (3). Then it holds

$$\begin{aligned} f(x, y^*) &= f(x, y^*) + \sum_{i=1}^m \mu_i \, |g_i(x, y^*)| + \sum_{i=1}^p \rho_i [h_i(x, y^*)]^- \\ &= \phi_1(x, y^*; \mu, \rho) \geq \phi_1(x^*, y^*; \mu, \rho) \\ &= f(x^*, y^*) + \sum_{i=1}^m \mu_i \, |g_i(x^*, y^*)| + \sum_{i=1}^p \rho_i [h_i(x^*, y^*)]^- \\ &= f(x^*, y^*). \end{aligned}$$

The analogous inequality holds for all $y \in Y$ such that (x^*, y) is feasible. Thus, (x^*, y^*) is a partial minimum of Problem (3). □

For the next theorem we need some more notation. Let χ be the ℓ_1 feasibility measure of Problem (3), which we define as

$$\chi(x, y) := \sum_{i=1}^m |g_i(x, y)| + \sum_{i=1}^p [h_i(x, y)]^-.$$

Obviously, $\chi(x, y) \geq 0$ holds and $\chi(x, y) = 0$ if and only if (x, y) is feasible w.r.t. g and h. Moreover, we define the weighted ℓ_1 feasibility measure as

$$\chi_{\mu,\rho}(x, y) := \sum_{i=1}^m \mu_i \, |g_i(x, y)| + \sum_{i=1}^p \rho_i [h_i(x, y)]^-,$$

i.e., our ℓ_1 penalty function can be stated as

$$\phi_1(x, y; \mu, \rho) = f(x, y) + \chi_{\mu,\rho}(x, y).$$

The next theorem states that the sequence of partial minima of the iteratively solved penalty problems converges to a partial minimum of $\chi_{\mu,\rho}$.

Lemma 3.2. *Suppose that Assumption 1 holds and that $\mu_i^k \nearrow \infty$ for all $i = 1, \dots, m$ and $\rho_i^k \nearrow \infty$ for all $i = 1, \dots, p$. Moreover, let (x^k, y^k) be a sequence of partial minima of (7) (for $\mu = \mu^k$ and $\rho = \rho^k$) generated by Algorithm 3 with $(x^k, y^k) \to (x^*, y^*)$. Then there exist weights $\bar{\mu}, \bar{\rho} \geq 0$ such that (x^*, y^*) is a partial minimizer of the feasibility measure $\chi_{\bar{\mu},\bar{\rho}}$.*

Proof. Let (x^k, y^k) be a partial minimizer of $\phi_1(x, y; \mu^k, \rho^k)$, i.e.,

$$\begin{aligned} \phi_1(x, y^k; \mu^k, \rho^k) &\geq \phi_1(x^k, y^k; \mu^k, \rho^k) \quad \text{for all } x \in X, \\ \phi_1(x^k, y; \mu^k, \rho^k) &\geq \phi_1(x^k, y^k; \mu^k, \rho^k) \quad \text{for all } y \in Y, \end{aligned}$$

which is equivalent to

$$\begin{aligned} & f(x, y^k) + \sum_{i=1}^{m} \mu_i^k \left| g_i(x, y^k) \right| + \sum_{i=1}^{p} \rho_i^k [h_i(x, y^k)]^- \\ \geq\ & f(x^k, y^k) + \sum_{i=1}^{m} \mu_i^k \left| g_i(x^k, y^k) \right| + \sum_{i=1}^{p} \rho_i^k [h_i(x^k, y^k)]^- \end{aligned} \tag{8}$$

for all $x \in X$ and

$$\begin{aligned} & f(x^k, y) + \sum_{i=1}^{m} \mu_i^k \left| g_i(x^k, y) \right| + \sum_{i=1}^{p} \rho_i^k [h_i(x^k, y)]^- \\ \geq\ & f(x^k, y^k) + \sum_{i=1}^{m} \mu_i^k \left| g_i(x^k, y^k) \right| + \sum_{i=1}^{p} \rho_i^k [h_i(x^k, y^k)]^- \end{aligned} \tag{9}$$

for all $y \in Y$. The sequence $(\mu^k, \rho^k) \subseteq \mathbb{R}^{m+p}$ is unbounded but the normed sequence

$$\frac{(\mu^k, \rho^k)}{\|(\mu^k, \rho^k)\|} \subseteq \mathbb{R}^{m+p},$$

is bounded. Thus, there exists a subsequence (indexed by l) of the normed sequence such that

$$\frac{(\mu^l, \rho^l)}{\|(\mu^l, \rho^l)\|} \to (\bar{\mu}, \bar{\rho}) \quad \text{for } l \to \infty.$$

For l sufficiently large, division of (8) and (9) by $\|(\mu^l, \rho^l)\|$ yields

$$\begin{aligned} & \frac{1}{\|(\mu^l, \rho^l)\|} f(x, y^l) + \sum_{i=1}^{m} \frac{\mu_i^l}{\|(\mu^l, \rho^l)\|} \left| g_i(x, y^l) \right| + \sum_{i=1}^{p} \frac{\rho_i^l}{\|(\mu^l, \rho^l)\|} [h_i(x, y^l)]^- \\ \geq\ & \frac{1}{\|(\mu^l, \rho^l)\|} f(x^l, y^l) + \sum_{i=1}^{m} \frac{\mu_i^l}{\|(\mu^l, \rho^l)\|} \left| g_i(x^l, y^l) \right| + \sum_{i=1}^{p} \frac{\rho_i^l}{\|(\mu^l, \rho^l)\|} [h_i(x^l, y^l)]^- \end{aligned}$$

for all $x \in X$ and

$$\begin{aligned} & \frac{1}{\|(\mu^l, \rho^l)\|} f(x^l, y) + \sum_{i=1}^{m} \frac{\mu_i^l}{\|(\mu^l, \rho^l)\|} \left| g_i(x^l, y) \right| + \sum_{i=1}^{p} \frac{\rho_i^l}{\|(\mu^l, \rho^l)\|} [h_i(x^l, y)]^- \\ \geq\ & \frac{1}{\|(\mu^l, \rho^l)\|} f(x^l, y^l) + \sum_{i=1}^{m} \frac{\mu_i^l}{\|(\mu^l, \rho^l)\|} \left| g_i(x^l, y^l) \right| + \sum_{i=1}^{p} \frac{\rho_i^l}{\|(\mu^l, \rho^l)\|} [h_i(x^l, y^l)]^- \end{aligned}$$

for all $y \in Y$. Finally, by using that the limit preserves non-strict inequalities and the linearity of the limit, for $l \to \infty$ we obtain

$$\sum_{i=1}^{m} \bar{\mu}_i \left| g_i(x, y^*) \right| + \sum_{i=1}^{p} \bar{\rho}_i [h_i(x, y^*)]^- \geq \sum_{i=1}^{m} \bar{\mu}_i \left| g_i(x^*, y^*) \right| + \sum_{i=1}^{p} \bar{\rho}_i [h_i(x^*, y^*)]^-$$

for all $x \in X$ and

$$\sum_{i=1}^{m} \bar{\mu}_i \left| g_i(x^*, y) \right| + \sum_{i=1}^{p} \bar{\rho}_i [h_i(x^*, y)]^- \geq \sum_{i=1}^{m} \bar{\mu}_i \left| g_i(x^*, y^*) \right| + \sum_{i=1}^{p} \bar{\rho}_i [h_i(x^*, y^*)]^-$$

for all $y \in Y$. This is equivalent to

$$\begin{aligned} \chi_{\bar{\mu}, \bar{\rho}}(x, y^*) &\geq \chi_{\bar{\mu}, \bar{\rho}}(x^*, y^*) \quad \text{for all } x \in X, \\ \chi_{\bar{\mu}, \bar{\rho}}(x^*, y) &\geq \chi_{\bar{\mu}, \bar{\rho}}(x^*, y^*) \quad \text{for all } y \in Y \end{aligned}$$

and thus completes the proof. □

The two preceding lemmas now enable us to characterize the overall convergence behavior of the penalty ADM algorithm 3.

Theorem 3.3. *Suppose that Assumption 1 holds and that $\mu_i^k \nearrow \infty$ for all $i = 1, \dots, m$ and $\rho_i^k \nearrow \infty$ for all $i = 1, \dots, p$. Moreover, let (x^k, y^k) be a sequence of partial minima of* (7) *(for $\mu = \mu^k$ and $\rho = \rho^k$) generated by Algorithm 3 with $(x^k, y^k) \to (x^*, y^*)$. Then*

a) (x^*, y^*) is a partial minimum of the original problem (3) *or*

b) there exist weights $\bar{\mu}, \bar{\rho} \geq 0$ such that (x^*, y^*) is a partial minimizer of the feasibility measure $\chi_{\bar{\mu},\bar{\rho}}$.

Proof. The second statement always holds by Lemma 3.2. If, in addition, the obtained partial minimum of $\chi_{\bar{\mu},\bar{\rho}}$ satisfies $\chi_{\bar{\mu},\bar{\rho}}(x^*, y^*) = 0$, we can apply Lemma 3.1 and obtain a partial minimum of the original problem (3). □

Corollary 3.4. Suppose the assumptions of Theorem (3.3) hold.

a) If f is additionally differentiable, then (x^*, y^*) is a stationary point of the original problem (3) or there exist weights $\bar{\mu}, \bar{\rho} \geq 0$ such that (x^*, y^*) is a partial minimizer of the feasibility measure $\chi_{\bar{\mu},\bar{\rho}}$.

b) If the assumption of a) holds and if the feasible set Ω as well f over Ω are additionally convex, then (x^*, y^*) is a global optimum of the original problem (3) or there exist weights $\bar{\mu}, \bar{\rho} \geq 0$ such that (x^*, y^*) is a partial minimizer of the feasibility measure $\chi_{\bar{\mu},\bar{\rho}}$.

In the last theorem we generalize the classical result on the exactness of the ℓ_1 penalty function (see, e.g., [32, 36]) to the setting of partial minima. For the ease of presentation, we state and prove this result only for the case without inequality constraints. However, the result can also be applied to problems including inequality constraints by using standard reformulation techniques to translate inequality constrained to equality constrained problems. Beforehand, we need two assumptions:

Assumption 2. *The objective function $f : X \times Y \to \mathbb{R}$ of Problem* (3) *is locally Lipschitz continuous in the direction of x and of y, i.e., for every $(x^*, y^*) \in \Omega$ there exists an open set $N(x^*, y^*)$ containing (x^*, y^*) and a constant $L \geq 0$ such that*

$$|f(x, y^*) - f(x^*, y^*)| \leq L\|x - x^*\| \quad \text{for all } x \text{ with } (x, y^*) \in N(x^*, y^*),$$
$$|f(x^*, y) - f(x^*, y^*)| \leq L\|y - y^*\| \quad \text{for all } y \text{ with } (x^*, y) \in N(x^*, y^*).$$

Assumption 3. *For every constraint $g_i, i = 1, \dots, m$, there exists a constant $l_i > 0$ such that*

$$l_i\|x - x^*\| \leq |g_i(x, y^*) - g_i(x^*, y^*)| \quad \text{for all } x \text{ with } (x, y^*) \in N(x^*, y^*),$$
$$l_i\|y - y^*\| \leq |g_i(x^*, y) - g_i(x^*, y^*)| \quad \text{for all } y \text{ with } (x^*, y) \in N(x^*, y^*).$$

Note that in the case of existing directional derivatives of g_i, the latter assumption states that the directional derivatives of the g_i both in the direction of x and of y are bounded away from zero. Before we state and proof the exactness theorem we briefly discuss the latter assumption. In the context of ADMs, the constraints $g(x, y) = 0$ are mostly so-called copy constraints of the type

$$g(x, y) = A(x - y) = 0$$

that are used to decompose the genuine problem formulation such that it fits into the framework of Problem (3); see, e.g., Nowak [37]. If the matrix A is square and has full rank—as it is typically the case for copy constraints—the constraints g are bi-Lipschitz and thus fulfill Assumption 3. Now, we are ready to state and prove the theorem on exactness of the ℓ_1 penalty function w.r.t. partial minima.

Theorem 3.5. *Let (x^*, y^*) be a partial minimizer of*

$$\min_{x,y} \quad f(x,y) \quad s.t. \quad g(x,y) = 0, x \in X, y \in Y, \tag{10}$$

and suppose that Assumptions 2 and 3 hold. Then there exists a constant $\bar{\mu} > 0$ such that (x^*, y^*) is a partial minimizer of

$$\min_{x,y} \quad \phi_1(x,y;\mu) \quad s.t. \quad x \in X, y \in Y$$

for all $\mu \geq \bar{\mu}$ and

$$\phi_1(x,y;\mu) := f(x,y) + \sum_{i=1}^m \mu_i \left| g_i(x,y) \right|.$$

Proof. Since (x^*, y^*) is a partial minimizer of Problem (10), it holds that

$$f(x, y^*) \geq f(x^*, y^*) \quad \text{for all } (x, y^*) \in \Omega, \tag{11a}$$

$$f(x^*, y) \geq f(x^*, y^*) \quad \text{for all } (x^*, y) \in \Omega, \tag{11b}$$

where Ω is the feasible region of Problem (10). First, assume that (x, y^*) is feasible for Problem (10). Using (11) we obtain

$$\begin{aligned}
\phi_1(x^*, y^*; \mu) &= f(x^*, y^*) + \sum_{i=1}^m \mu_i \left| g_i(x^*, y^*) \right| \\
&= f(x^*, y^*) \leq f(x, y^*) \\
&= f(x, y^*) + \sum_{i=1}^m \mu_i \left| g_i(x, y^*) \right| \\
&= \phi_1(x, y^*; \mu)
\end{aligned}$$

for all μ. The inequality

$$\phi_1(x^*, y^*; \mu) \leq \phi_1(x^*, y; \mu)$$

can be shown analogously.

We now consider the case that (x, y^*) is not feasible for Problem (10). We set $\bar{\mu} := L/(m\bar{l})e > 0$, where $\bar{l} := \min_{i=1,\dots,m} \{l_i\}, e = (1, \dots, 1)^\top \in \mathbb{R}^m$, and show that for all $\mu \geq \bar{\mu}$ the inequality

$$f(x, y^*) + \sum_{i=1}^m \mu_i \left| g_i(x, y^*) \right| \geq f(x^*, y^*) + \sum_{i=1}^m \mu_i \left| g_i(x^*, y^*) \right| \tag{12}$$

holds for all $x \in X$. Since (x^*, y^*) is feasible for Problem (10), Inequality (12) is equivalent to

$$\sum_{i=1}^m \mu_i \left| g_i(x, y^*) \right| \geq f(x^*, y^*) - f(x, y^*).$$

With the definition of $\bar{\mu}$ we obtain

$$\begin{aligned}
&\sum_{i=1}^m \mu_i \left| g_i(x, y^*) \right| \geq \bar{\mu} \sum_{i=1}^m \left| g_i(x, y^*) \right| = \bar{\mu} \sum_{i=1}^m \left| g_i(x, y^*) - g_i(x^*, y^*) \right| \\
&\geq \bar{\mu} \sum_{i=1}^m l_i \|x - x^*\| \geq \bar{\mu} m \bar{l} \|x - x^*\| = L \|x - x^*\| \\
&\geq \left| f(x^*, y^*) - f(x, y^*) \right| \geq f(x^*, y^*) - f(x, y^*).
\end{aligned}$$

Thus, (12) holds for all $\mu \geq \bar{\mu}$. Analogously, it can be shown that the inequality

$$f(x^*, y) + \sum_{i=1}^{m} \mu_i \, |g_i(x^*, y)| \geq f(x^*, y^*) + \sum_{i=1}^{m} \mu_i \, |g_i(x^*, y^*)|$$

holds for all $y \in Y$. □

3.3. **The Penalty ADM as a Feasibility Pump.** In this section we discuss the application of the proposed penalty alternating direction method as a new variant of a feasibility pump algorithm for convex MINLPs. The motivation is the following: We have seen in the last section that idealized, i.e., perturbation-free, feasibility pumps for convex MINLPs terminate at partial minima after a finite number of iterations. However, it is possible that the obtained partial minimum is not integer feasible. Feasibility pumps typically try to resolve this problem by applying a random perturbation of the integer components. This procedure has the significant drawback that it renders a convergence theory of the overall method (almost) impossible. In contrast to these random perturbations, the method we propose uses a theoretically analyzable penalty framework to escape integer infeasible partial minima.

We start by rewriting Problem (2) by again duplicating the integer components x_I of x and obtain

$$\min_{x,y} \quad f(x) \quad \text{s.t.} \quad h(x) \geq 0, \quad x_I = y, \quad y \in \mathbb{Z}^I \cap [l_I, u_I]. \tag{13}$$

With the compact sets

$$X := \{x : h(x) \geq 0\}, \quad Y := \mathbb{Z}^I \cap [l_I, u_I]$$

and the additional equality constraints

$$g(x, y) = x_I - y = 0$$

we can apply Algorithm 3 to Problem (13). Note that the problem interfaced to the penalty ADM does not contain any inequality constraints explicitly since we moved them to the set X. This also simplifies the ℓ_1 penalty function $\phi_1(x, y; \mu, \rho)$ to $\phi_1(x, y; \mu)$. In the lth ADM iteration of the kth penalty iteration of Algorithm 3 the two subproblems being solved are

$$\min_{x \in X} \quad \phi_1(x, y^{k,l}; \mu^k),$$

which can be written as

$$\min_{x} \quad f(x) + \sum_{i \in I} \mu_i^k \left| x_i - y_i^{k,l} \right| \quad \text{s.t.} \quad h(x) \geq 0, \tag{14}$$

and

$$\min_{y \in Y} \quad \phi_1(x^{k,l+1}, y; \mu^k),$$

which can be written as

$$\min_{y} \quad f(x^{k,l+1}) + \sum_{i \in I} \mu_i^k \left| x_i^{k,l+1} - y_i \right| \quad \text{s.t.} \quad y \in \mathbb{Z}^I \cap [l_I, u_I]. \tag{15}$$

Note that Problem (14) is the NLP relaxation of (2), where the original objective function is augmented by a weighted ℓ_1 penalty term. Note further that solving Problem (15) simply means to apply a weighted rounding of the variables $y_i, i \in I$.

4. Implementation Issues

In this section we comment on important implementation issues. First, we rewrite Problem (13) by replacing the coupling equality $x_I = y$ by inequality constraints in order to be able to penalize a coupling error $x_i > y_i, i \in I$, different than the error $y_i > x_i$. In addition, we also explicitly state all variable bounds from now on. Thus, we obtain

$$\begin{aligned} \min_{x,y} \quad & f(x) \\ \text{s.t.} \quad & h(x) \geq 0, \quad x \in [l, u], \\ & x_I \geq y, \quad y \geq x_I, \quad y \in \mathbb{Z}^I \cap [l_I, u_I]. \end{aligned}$$

In other words, we slightly modified the compact and non-empty constraint sets to

$$X := \{x \in [l, u] : h(x) \geq 0\}, \quad Y := \mathbb{Z}^I \cap [l_I, u_I]$$

and replaced the coupling equalities by coupling inequalities. The two subproblems (14) and (15) that are solved within the lth ADM iteration of the kth penalty iteration are now given by

$$\min_{x} \quad f(x) + \sum_{i \in I} \left(\underline{\rho}_i^k [x_i - y_i^{k,l}]^- + \bar{\rho}_i^k [y_i^{k,l} - x_i]^- \right) \tag{16a}$$

$$\text{s.t.} \quad h(x) \geq 0, \quad x \in [l, u], \tag{16b}$$

and

$$\min_{y} \quad f(x^{k,l+1}) + \sum_{i \in I} \left(\underline{\rho}_i^k [x_i^{k,l+1} - y_i]^- + \bar{\rho}_i^k [y_i - x_i^{k,l+1}]^- \right) \tag{17a}$$

$$\text{s.t.} \quad y \in \mathbb{Z}^I \cap [l_I, u_I], \tag{17b}$$

i.e., we also replaced the single penalty parameters μ_i for the equality coupling constraints in Problem (13) by two new penalty parameters $\underline{\rho}_i$ and $\bar{\rho}_i$ for the lower and upper violation of the coupling.

In order to actually implement the penalty ADM based feasibility pump for MINLPs, we follow Achterberg and Berthold [1] and scale the objective function of Problem (16) such that the impact of the ℓ_1 penalty terms and the original objective function f can be balanced. This balancing between feasibility and optimality is done using the parameter $\alpha^k \in [0, 1]$. Additionally, we again rewrite the ℓ_1 penalty terms in the objective function. To this end, we denote the set of indices of binary variables by $B \subseteq I$, introduce the variables $d_i^+, d_i^- \geq 0$ for all $i \in I \setminus B$, and rewrite Problem (16) as

$$\begin{aligned} \min_{x,d} \quad & \alpha^k \frac{\sqrt{|I|}}{\|f(x^{(0,0)})\|} f(x) + (1 - \alpha^k) \tilde{\chi}(x_B, d_{I \setminus B}; \rho_I^{\pm}) \\ \text{s.t.} \quad & h(x) \geq 0, \quad x \in [l, u], \\ & d_i^+ \geq x_i - y_i^{k,l} \quad \text{for all } i \in I \setminus B, \\ & d_i^- \geq y_i^{k,l} - x_i \quad \text{for all } i \in I \setminus B, \\ & d_i^-, d_i^+ \geq 0 \end{aligned} \tag{18}$$

with

$$\tilde{\chi}(x_B, d_{I \setminus B}; \rho_I^{\pm}) := \sum_{i \in B_0} \underline{\rho}_i^k x_i + \sum_{i \in B_1} \bar{\rho}_i^k (1 - x_i) + \sum_{i \in I \setminus B} \left(\underline{\rho}_i^k d_i^+ + \bar{\rho}_i^k d_i^- \right),$$

where $B_0 := \{i \in B : y_i^{k,l} = 0\}$ and $B_1 := \{i \in B : y_i^{k,l} = 1\}$. For binary MINLPs, i.e., $I = B$, only the objective function of Problem (18) may change from one iteration to the next. This is of special importance for mixed-integer linear problems since then the optimal simplex basis obtained in iteration k yields a primal feasible starting basis for iteration $k+1$. However, when $I \neq B$, the optimal basis obtained from iteration k is generally (primal and dual) infeasible for the LP that has to be solved in iteration $k+1$.

Since $f(x^{k,l+1})$ is constant and $\alpha^k \in [0,1]$, solving Problem (17) is equivalent to solving the $|I|$ independent problems

$$y_i^{k,l+1} := \operatorname*{argmin}_{y_i} \left\{ \underline{\rho}_i^k [x_i^{k,l+1} - y_i]^- + \bar{\rho}_i^k [y_i - x_i^{k,l+1}]^- \;\middle|\; y_i \in \mathbb{Z} \cap [l_i, u_i] \right\}, \quad i \in I.$$

The solutions to these problems can be stated explicitly:

$$y_i^{k,l+1} = \begin{cases} \lceil x_i^{k,l+1} \rceil, & \text{if } \bar{\rho}_i^k (\lceil x_i^{k,l+1} \rceil - x_i^{k,l+1}) \leq \underline{\rho}_i^k (x_i^{k,l+1} - \lfloor x_i^{k,l+1} \rfloor), \\ \lfloor x_i^{k,l+1} \rfloor, & \text{otherwise.} \end{cases}$$

Finally, we have a look on the update of the penalty parameters. Assume that the kth penalty iteration is finished after performing l ADM iterations. Then we set

$$\underline{\rho}_i^{k+1} = \begin{cases} \operatorname{inc}(\underline{\rho}_i^k), & \text{if } y_i^{k,l+1} = \lceil x_i^{k,l+1} \rceil, \\ \underline{\rho}_i^k, & \text{otherwise,} \end{cases}$$

$$\bar{\rho}_i^{k+1} = \begin{cases} \operatorname{inc}(\bar{\rho}_i^k), & \text{if } y_i^{k,l+1} = \lfloor x_i^{k,l+1} \rfloor, \\ \bar{\rho}_i^k, & \text{otherwise,} \end{cases}$$

where the penalty parameter update operator $\operatorname{inc}(a)$ may be any function with $\operatorname{inc}(a) > a$, e.g., $\operatorname{inc}(a) = a + 1$ or $\operatorname{inc}(a) = 10a$ are used in our computational study. This way, unsuccessful rounding down of the same variable is eventually followed by rounding up of this variable due to increasingly penalizing rounding down and vice versa. Similarly, the conventional feasibility pump algorithm tries to escape from repeated rounding in the "wrong" direction by randomly switching the rounding direction from time to time; see Fischetti et al. [23]. Finally, we set $\alpha^{k+1} = \lambda \alpha^k$ with $\lambda \in (0,1)$ whenever the penalty parameters are updated.

We note that choosing $\alpha_0 = 0$ in our penalty alternating direction method based feasibility pump is similar to the feasibility pump presented in Bertacco et al. [3] and Fischetti et al. [23], while choosing $\alpha_0 = 1$ yields an algorithm the behaves similar to the objective feasibility pump algorithm presented by Achterberg and Berthold [1].

Lastly, we note that for $\alpha_0 = 0$, we also have $\alpha_k = 0$ for all $k > 0$. In this case the first term of the objective function of Problem (18) vanishes for all k, l. In any other case, we can divide the entire objective function by α_k to be conformal to the theoretical setting presented in previous sections.

5. Computational Results

In this section we present extensive numerical results for the penalty ADM based feasibility pump introduced in Section 3 and 4. Since our method is completely generic in terms of the problem type to which it is applied, we present computational results both for MIPs in Section 5.1 and for (convex as well as nonconvex) MINLPs in Section 5.2.

Throughout this section we use standard performance profiles as proposed by Dolan and Moré [19] to compare running times and solution quality. Following [34], solution quality is measured by the primal-dual gap defined by

$$\text{gap} = \frac{b_\text{p} - b_\text{d}}{\inf\{|z| : z \in [b_\text{d}, b_\text{p}]\}}, \tag{19}$$

where b_p is the primal and b_d is the dual bound, respectively. Additionally, we set $\text{gap} = \infty$ whenever $b_\text{d} < 0 \leq b_\text{p}$ and $\text{gap} = 0$ if $b_\text{d} = b_\text{p} = 0$.

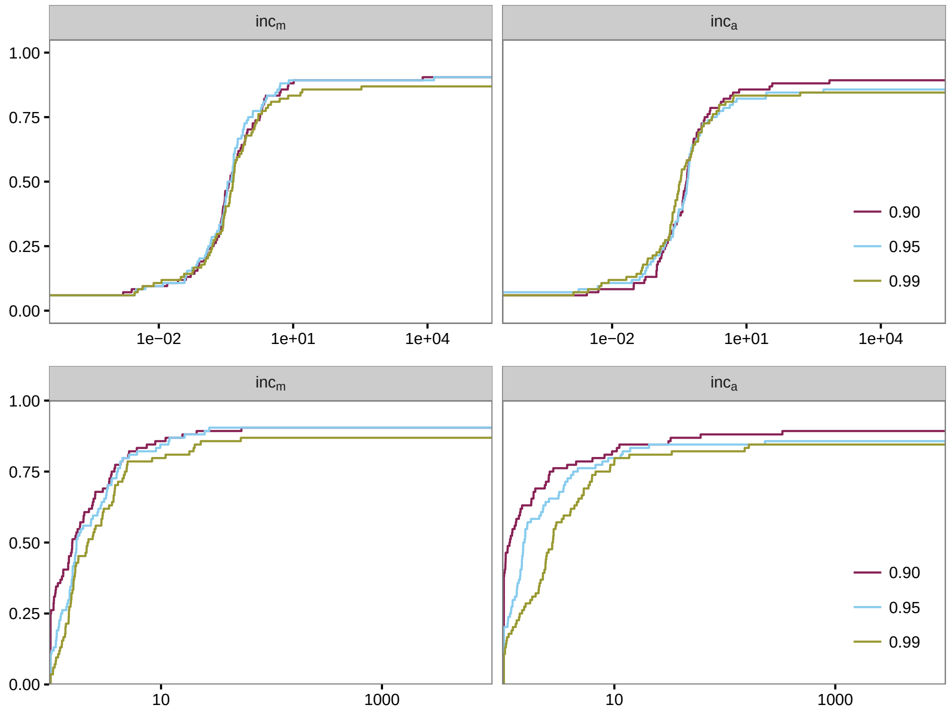

FIGURE 1. Non-dominated parameterizations $\lambda \in \{0.9, 0.95, 0.99\}$, $\text{inc} \in \{\text{inc}_\text{m}, \text{inc}_\text{a}\}$ (all with activated Gurobi presolve) for the penalty ADM based feasibility pump for MIPs. Top: primal-dual gap. Bottom: running times.

All computational experiments have been executed on a 12 core Xeon 5650 "Westmere" chip running at 2.66 GHz with 12 MB shared cache per chip and 24 GB of DDR3-1333 RAM. The time limit is set to $t^+ = 1\,\text{h}$ without any limit on the number of iterations for the outer penalty and the inner ADM loop. Additional information about the computational setup and the implementation details are given in the respective sections.

5.1. **Mixed-Integer Linear Problems.** We start with discussing the results of our algorithm applied to mixed-integer linear problems. For MIPs, our algorithm is implemented in C++ and uses Gurobi 6.5.0 [31] for solving the LP subproblems. We use Gurobi's option `deterministic concurrent` for the first LP and solve all succeeding LPs using the primal simplex method; see Section 4. The C++ code has been compiled with gcc 4.8.4 using the optimization flag o3. First, we present a parameter study. Our penalty ADM based feasibility pump can be instantiated using different choices for certain algorithmic parameters: The initial convex combination parameter α^0 for weighting the objective function and the distance function $\tilde{\chi}$ (see Problem (18)) is always set to $\alpha^0 = 1$, emulating the objective feasibility pump of Achterberg and Berthold [1]. The parameter λ for updating the convex combination parameter is varied in the set $\{0.9, 0.95, 0.99\}$ in our parameter study and the penalty parameter update operator can be chosen to be the additive variant $\text{inc}_\text{a}(x) = x + 1$ or the multiplicative variant $\text{inc}_\text{m}(x) = 10x$. In addition, we also tested the impact of (de)activating the MIP presolve of Gurobi before applying our method. Thus, combining the different choices for λ, inc, and the (de)activation of Gurobi's presolve leads to 12 parameter combinations. We applied

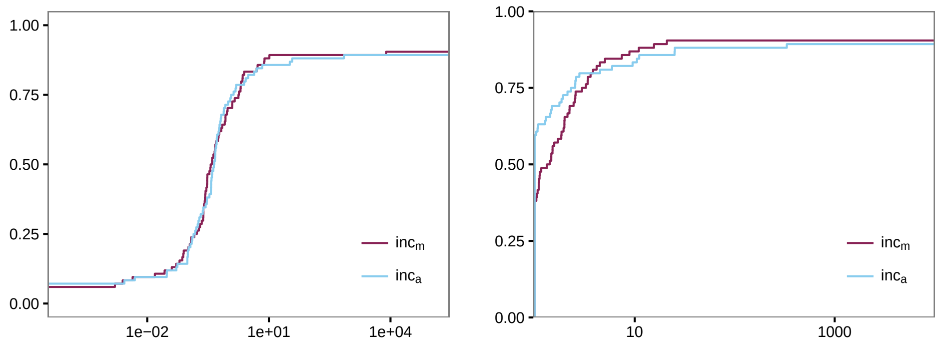

FIGURE 2. Performance profiles of primal-dual gap (left) and running times (right) for the winner parameterizations $\lambda = 0.9$, activated Gurobi presolve, and $\text{inc} \in \{\text{inc}_\text{m}, \text{inc}_\text{a}\}$ for MIPs

every of these 12 variants of our method to solve the MIPLIB 2010 benchmark test set excluding the infeasible instances sh608gpia-3col, enlight14, and ns1766074. This yields a test set of 84 instances; see Koch et al. [34]. In order to determine the best parameterization, we compare all 12 variants using performance profiles, where the performance measure is chosen as defined in (19). We then exclude a parameterization p if a another parameterization p' exists that dominates p. Here, domination is defined by a performance profile completely left-above the other one. This yields the exclusion of deactivating Gurobi's presolve and, thus, 6 remaining parameterizations; $\lambda \in \{0.9, 0.95, 0.99\}$ and $\text{inc} \in \{\text{inc}_\text{m}, \text{inc}_\text{a}\}$. The corresponding performance profiles are given in Figure 1. It can be seen that lower values for λ yield more robust instantiations of the algorithm, i.e., the number of instances for which a feasible solution can be found is larger. Additionally, all tested variants solve 5 out of 84 instances to global optimality, except for the variant with $\lambda = 0.95$ and inc_a penalty parameter update rule, which solves 6 instances to global optimality. Altogether, the six parameter choices are quite comparable. Turning to running times, it can be clearly seen that smaller values of λ also lead to shorter running times. Thus, our parameter study suggests to activate the MIP presolve of Gurobi, to choose $\lambda = 0.9$, and to leave the choice of the penalty parameter update rule $\text{inc} \in \{\text{inc}_\text{m}, \text{inc}_\text{a}\}$ as an option for the user. Figure 2 shows the performance profiles for solution quality (left) and running times (right) for these "winning" parameterizations. We again see that some instances are solved to global optimality[1] and that both parameterizations of our algorithm find a feasible solution for approximately 90 % of the MIPLIB 2010 benchmark instances (75 out of 84 instances for the inc_a update rule and 76 for the multiplicative rule inc_m). Moreover, the multiplicative update operator inc_m yields a slightly more robust algorithm, i.e., it finds a feasible solution for a few more instances than the additive version inc_a. The right part of Figure 2 compares the two winner instantiations w.r.t. running times. It can be seen that the additive inc_a update operator tends to result in a faster algorithm for significantly more instances than the multiplicative version (59.5 % vs. 38.1 %).

Next, we compare our penalty ADM based feasibility pump with the objective feasibility pump of Achterberg and Berthold [1]. In order to achieve a fair comparison, we extend our method with a local branching strategy with an additional k-opt

[1]The instances triptim1, pigeon-10, enlight13, ex9, and ns1758913 are solved to global optimality using the inc_m penalty update rule and ns1208400, triptim1, acc-tight5, enlight13, ex9, and ns1758913 are solved to global optimality using inc_a.

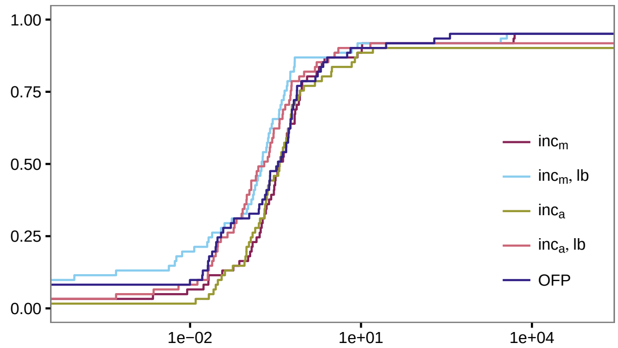

FIGURE 3. The two winner parameterizations (see Figure 2) with and without local branching compared to the objective feasibility pump (OFP) by Achterberg and Berthold [1]

neighborhood constraint with $k = \min\{20, \lfloor |I|/10 \rfloor\}$ and a time limit $t^+ - t^*$; see Fischetti and Lodi [24]. Here t^* denotes the time spent in the penalty ADM based feasibility pump itself. Thus, the local branching stage serves as an improvement heuristic as it is also the case in [1]. As test instances we use all MIPLIB 2010 and MIPLIB 2003 instances that have been used in [1] as well. Figure 3 shows the primal-dual gap performance profiles of the two winner parameterizations ($\lambda = 0.9$ and $\text{inc} \in \{\text{inc}_\text{m}, \text{inc}_\text{a}\}$) with and without local branching applied as an additional improvement heuristic as well as the corresponding performance profile curve based on the results reported by Achterberg and Berthold [1]. First of all, it can be seen that all five methods find a feasible solution for at least 90.2 % of the tested instances, which underpins the strength of feasibility pumps in general. Comparing only the different parameterization of our method we see that the inc_m update rule with local branching outperforms the version without local branching and both variants using the additive penalty update rule. The latter also performs similar independent of whether local branching is used or not, whereas the local branching stage significantly improves the solution quality when the inc_m rule is used (in which case we find a global optimal solution for 11.5 %; compared to 8.2 % for the objective feasibility pump of Achterberg and Berthold [1]). One sees that the multiplicative update rule together with local branching slightly outperforms the objective feasibility pump of Achterberg and Berthold [1]. Thus, our method does not only allow for a full theoretical analysis but can also compete with state-of-the-art implementations of feasibility pumps for MIPs.

We close this section with an exemplary discussion of the course of integer (in)feasibility during the iterations of our method. Figure 4 shows the approximately 50 last iterations of our method applied to the MIPLIB 2010 instance rococoC10-001000. Dots correspond to numbers of fractional integer components and the solid line represents the course of the total fractionality measure

$$\sum_{i \in I} |x_i - \lfloor x_i + 0.5 \rfloor|$$

of solutions x of the continuous subproblems over the subsequent ADM iterations. The small black lines on top of the ADM iteration axis denote iterations at which the penalty parameters are updated.

First of all, we see that penalty parameter updates are applied whenever the ADM of the inner loop stalls, i.e., whenever the ADM of the inner loop entered an undesired integer infeasible partial minimum. The method stops after 295 ADM

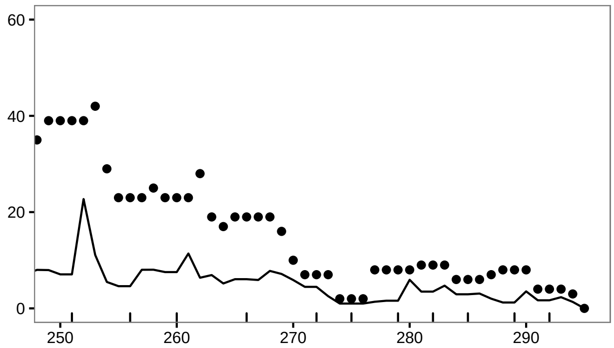

FIGURE 4. Number of fractional integer components (dots) and total fractionality (solid line) vs. ADM iterations. Penalty parameter updates are marked with small black vertical lines on top of the ADM iteration axis.

iterations with an integer feasible partial minimum. As expected, we typically see a sawtooth phenomenon: The total fractionality decreases between two consecutive penalty parameter updates and increases after a penalty parameter update. The number of fractional integer components follows this behavior qualitatively. The number of ADM iterations between two consecutive penalty parameter updates varies between 3 to 6 iterations. Thus, convergence to partial minima does not seem to be challenging for this specific instance.

5.2. **Mixed-Integer Nonlinear Problems.** We now turn to mixed-integer nonlinear programs. The penalty based ADM for this class of models has been implemented in C++ using the so-called GAMS Expert-level API with GAMS 24.5.4 [13]. The continuous relaxation models are solved with CONOPT 3.17A [20]. According to the results from Section 5.1 we choose the parameters $\alpha^0 = 1$ and $\lambda = 0.9$. The penalty parameter update rule is chosen to be $\text{inc}_{\text{a}}(x) = x + 1$ since this variant turned out to be favorable for MINLPs. We set the time limit to 1 h as for the MIP experiments and we do not incorporate any iteration limits for the inner ADM and the outer penalty loop. Throughout this section we declare an MINLP instance as solved to feasibility if CONOPT finds a feasible solution (w.r.t. to its default tolerances) with all integer components fixed to integral values.

We compare the results of our method with other recently published numerical results concerning feasibility pump algorithms for convex and nonconvex MINLPs. Most of the results from the literature that we use for these comparisons are based on the first and second version of the MINLPLib; see Bussieck et al. [12]. Since a reasonable comparison of running times is not possible due to differences in the used hardware, we focus on the comparison of success rates and solution quality.

We also compared the obtained results separately for convex and nonconvex instances. As expected, the results of our method are slightly better for the convex case. However, the results are qualitatively rather similar and we thus present the following analysis of our computational results without distinguishing between convex and nonconvex instances.

First, we start with a comparison of different feasibility pump versions presented by D'Ambrosio et al. in [14] and our method on selected MINLPLib instances. D'Ambrosio et al. test their method on 65 MINLPLib instances. However, it turned out in the meantime that the used test set contained 4 duplicate instances. Thus, the following comparison is carried out on 61 MINLPLib instances. Figure 5 displays the performance profiles using the primal-dual gap as the performance measure; see (19).

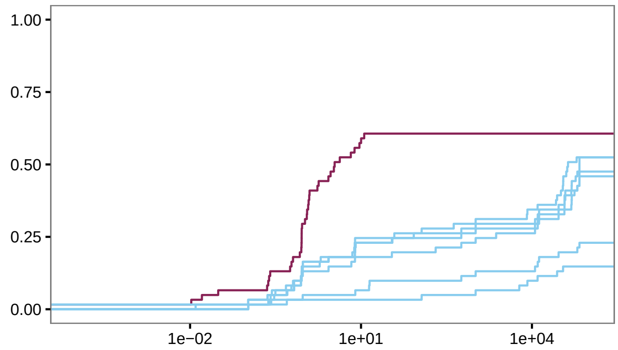

FIGURE 5. Performance profiles for the primal-dual gap of the penalty ADM based feasibility pump (red) and all six feasibility pump variants (blue) for MINLPs presented in [14] on subset of 61 MINLPLib instances.

It can be clearly seen that the penalty ADM based feasibility pump outperforms all feasibility pump variants presented in [14]. Although the number of instances solved to optimality is comparably low for all algorithms, the overall solution quality of our penalty ADM based algorithm is significantly higher than the quality of solutions obtained in D'Ambrosio et al. [14]. Additionally, the number of instances for which we found a feasible solution is 60.7 % whereas the percentage of instances solved to feasibility in D'Ambrosio et al. [14] ranges from 54.1 % to only 16.4 %.

Complementing this comparison on the MINLPLib test set we also tested our method on 889 out of 1385 instances of the more recent MINLPLib2 test set. Here, we neglected all instances containing only continuous variables. Again, our method behaves quite satisfactory. It computes a feasible solution for 642 instances (72.2 %), leaving 247 instances unsolved. Moreover, the penalty ADM based feasibility pump computes solutions with a vanishing primal-dual gap for 83 instances; i.e., for 9.3 % of all instances of the test set.

Finally, we discuss the most recent (at least to the best of our knowledge) results on a new variant of feasibility pumps for nonconvex mixed-integer nonlinear problems that are published by Berthold in his PhD thesis [4]. The test set used by Berthold is neither a sub- nor a superset of the current MINLPLib. Thus, we compare our method with the algorithms proposed by Berthold on all instances of his test set that are also part of the current MINLPLib version, yielding a test set of 154 instances. Again, we compare the methods by performance profiles of the primal-dual gap, see Figure 6. The penalty ADM based feasibility pump significantly outperforms all variants of the feasibility pump for MINLPs proposed by Berthold. First, the methods of Berthold do not solve any instance to optimality, whereas we close the primal-dual gap for 15.6 % of the instances. Second, our method finds a feasible solution for 79.9 % of the tested instances, whereas the methods of Berthold solve approximately 43.5 % to 64.9 % to feasibility.

6. SUMMARY

In this paper we have shown that idealized feasibility pumps, i.e., feasibility pumps without random perturbations, can be seen as alternating direction methods applied to a special reformulation of the original mixed-integer problem. This yields that idealized feasibility pumps converge to a partial minimum of the reformulated problem. If this partial minimum is not an integer feasible point, feasibility pumps apply a random perturbation to escape this undesired point. We replace this

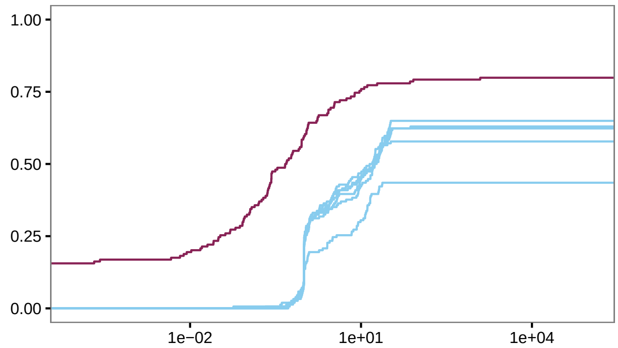

Figure 6. Performance profiles for the primal-dual gap of the penalty ADM based feasibility pump (red) and different feasibility pump variants (blue) proposed by Berthold [4]

random restart with a penalty framework that encompasses the alternating direction method in the inner loop and that replaces random perturbations by tailored penalty parameter updates. This way it is possible for the first time to perform a theoretical study for a variant of the feasibility pump including restarts. The resulting penalty based alternating direction method can be applied to both MIPs and MINLPs. Our numerical results indicate that this new version of the feasibility pump is comparable (w.r.t. to most recent publications) for the case of MIPs and clearly outperforms other feasibility pump algorithms on MINLPs.

Acknowledgements

We acknowledge funding through the DFG SFB/Transregio 154, Subprojects A05 and B07. This research has been performed as part of the Energie Campus Nürnberg and supported by funding through the "Aufbruch Bayern (Bavaria on the move)" initiative of the state of Bavaria.

References

[1] T. Achterberg and T. Berthold. "Improving the feasibility pump." In: *Discrete Optimization* 4.1 (2007). Mixed Integer Programming IMA Special Workshop on Mixed-Integer Programming, pp. 77–86. doi: 10.1016/j.disopt.2006.10.004.

[2] D. Baena and J. Castro. "Using the analytic center in the feasibility pump." In: *Operations Research Letters* 39.5 (2011), pp. 310–317. doi: 10.1016/j.orl.2011.07.005.

[3] L. Bertacco, M. Fischetti, and A. Lodi. "A feasibility pump heuristic for general mixed-integer problems." In: *Discrete Optimization* 4.1 (2007). Mixed Integer Programming IMA Special Workshop on Mixed-Integer Programming, pp. 63–76. doi: 10.1016/j.disopt.2006.10.001.

[4] T. Berthold. "Heuristic algorithms in global MINLP solvers." PhD thesis. Technische Universität Berlin, 2014.

[5] T. Berthold. "Primal Heuristics for Mixed Integer Programs." Diploma Thesis. TU Berlin, Sept. 2006.

[6] N. L. Boland, A. C. Eberhard, F. Engineer, and A. Tsoukalas. "A New Approach to the Feasibility Pump in Mixed Integer Programming." In: *SIAM Journal on Optimization* 22.3 (2012), pp. 831–861. doi: 10.1137/110823596.

[7] N. L. Boland, A. C. Eberhard, F. G. Engineer, M. Fischetti, M. W. P. Savelsbergh, and A. Tsoukalas. "Boosting the feasibility pump." In: *Mathematical Programming Computation* 6.3 (2014), pp. 255–279. DOI: 10.1007/s12532-014-0068-9.

[8] P. Bonami, G. Cornuéjols, A. Lodi, and F. Margot. "A Feasibility Pump for mixed integer nonlinear programs." In: *Mathematical Programming* 119.2 (2009), pp. 331–352. DOI: 10.1007/s10107-008-0212-2.

[9] P. Bonami and J. P. M. Gonçalves. "Heuristics for convex mixed integer nonlinear programs." In: *Computational Optimization and Applications* 51.2 (2012), pp. 729–747. DOI: 10.1007/s10589-010-9350-6.

[10] P. Bonami, M. Kilinç, and J. Linderoth. "Algorithms and Software for Convex Mixed Integer Nonlinear Programs." In: *Mixed Integer Nonlinear Programming*. Ed. by J. Lee and S. Leyffer. Vol. 154. The IMA Volumes in Mathematics and its Applications. Springer New York, 2012, pp. 1–39. DOI: 10.1007/978-1-4614-1927-3_1.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers." In: *Found. Trends Mach. Learn.* 3.1 (Jan. 2011), pp. 1–122. DOI: 10.1561/2200000016.

[12] M. R. Bussieck, A. S. Drud, and A. Meeraus. "MINLPLib - A Collection of Test Models for Mixed-Integer Nonlinear Programming." In: *INFORMS Journal on Computing* 15.1 (2003), pp. 114–119. DOI: 10.1287/ijoc.15.1.114.15159.

[13] G. D. Corporation. *General Algebraic Modeling System (GAMS) Release 24.5.4*. Washington, DC, USA. 2015. URL: http://www.gams.com.

[14] C. D'Ambrosio, A. Frangioni, L. Liberti, and A. Lodi. "A storm of feasibility pumps for nonconvex MINLP." In: *Mathematical Programming* 136.2 (2012), pp. 375–402. DOI: 10.1007/s10107-012-0608-x.

[15] C. D'Ambrosio, A. Frangioni, L. Liberti, and A. Lodi. "Experiments with a Feasibility Pump Approach for Nonconvex MINLPs." In: *Experimental Algorithms*. Ed. by P. Festa. Vol. 6049. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 350–360. DOI: 10.1007/978-3-642-13193-6_30.

[16] E. Danna, E. Rothberg, and C. L. Pape. "Exploring relaxation induced neighborhoods to improve MIP solutions." In: *Mathematical Programming* 102.1 (2005), pp. 71–90. DOI: 10.1007/s10107-004-0518-7.

[17] M. De Santis, S. Lucidi, and F. Rinaldi. "A New Class of Functions for Measuring Solution Integrality in the Feasibility Pump Approach." In: *SIAM Journal on Optimization* 23.3 (2013), pp. 1575–1606. DOI: 10.1137/110855351.

[18] M. De Santis, S. Lucidi, and F. Rinaldi. "Feasibility Pump-like heuristics for mixed integer problems." In: *Discrete Applied Mathematics* 165 (2014). 10th Cologne/Twente Workshop on Graphs and Combinatorial Optimization (CTW 2011), pp. 152–167. DOI: 10.1016/j.dam.2013.06.018.

[19] E. D. Dolan and J. J. Moré. "Benchmarking Optimization Software with Performance Profiles." In: *Mathematical Programming* 91 (2 2002), pp. 201–213. DOI: 10.1007/s101070100263.

[20] A. Drud. "CONOPT—A Large-Scale GRG Code." In: *ORSA Journal on Computing* 6.2 (1994), pp. 207–216.

[21] M. A. Duran and I. E. Grossmann. "An outer-approximation algorithm for a class of mixed-integer nonlinear programs." In: *Mathematical Programming* 36.3 (1986), pp. 307–339. DOI: 10.1007/BF02592064.

[22] J. Eckstein and M. Nediak. "Pivot, Cut, and Dive: a heuristic for 0-1 mixed integer programming." In: *Journal of Heuristics* 13.5 (2007), pp. 471–503. DOI: 10.1007/s10732-007-9021-7.

[23] M. Fischetti, F. Glover, and A. Lodi. "The feasibility pump." In: *Mathematical Programming* 104.1 (2005), pp. 91–104. DOI: 10.1007/s10107-004-0570-3.

[24] M. Fischetti and A. Lodi. "Local branching." In: *Mathematical Programming* 98.1-3 (2003), pp. 23–47. DOI: 10.1007/s10107-003-0395-5.

[25] M. Fischetti and D. Salvagnin. "Feasibility pump 2.0." In: *Mathematical Programming Computation* 1.2-3 (2009), pp. 201–222. DOI: 10.1007/s12532-009-0007-3.

[26] M. Frank and P. Wolfe. "An algorithm for quadratic programming." In: *Naval Research Logistics Quarterly* 3.1-2 (1956), pp. 95–110. DOI: 10.1002/nav.3800030109.

[27] D. Gabay and B. Mercier. "A dual algorithm for the solution of nonlinear variational problems via finite element approximation." In: *Computers & Mathematics with Applications* 2.1 (1976), pp. 17–40. DOI: 10.1016/0898-1221(76)90003-1.

[28] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt. "Solving power-constrained gas transportation problems using an MIP-based alternating direction method." In: *Computers & Chemical Engineering* 82 (2015), pp. 303–317. DOI: 10.1016/j.compchemeng.2015.07.005.

[29] R. Glowinski and A. Marroco. "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires." In: *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique* 9.R2 (1975), pp. 41–76. URL: http://eudml.org/doc/193269.

[30] J. Gorski, F. Pfeuffer, and K. Klamroth. "Biconvex sets and optimization with biconvex functions: a survey and extensions." In: *Math. Methods Oper. Res.* 66.3 (2007), pp. 373–407. DOI: 10.1007/s00186-007-0161-1.

[31] Gurobi Optimization, Inc. *Gurobi Optimizer Reference Manual, Version 6.5.* 2015.

[32] S.-P. Han and O. L. Mangasarian. "Exact penalty functions in nonlinear programming." In: *Mathematical Programming* 17.1 (1979), pp. 251–269. DOI: 10.1007/BF01588250.

[33] S. Hanafi, J. Lazić, and N. Mladenović. "Variable Neighbourhood Pump Heuristic for 0-1 Mixed Integer Programming Feasibility." In: *Electronic Notes in Discrete Mathematics* 36 (2010). ISCO 2010 - International Symposium on Combinatorial Optimization, pp. 759–766. DOI: 10.1016/j.endm.2010.05.096.

[34] T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, E. Danna, G. Gamrath, A. M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D. E. Steffy, and K. Wolter. "MIPLIB 2010." In: *Mathematical Programming Computation* 3.2 (2011), pp. 103–163. DOI: 10.1007/s12532-011-0025-9.

[35] O. L. Mangasarian. "Solution of general linear complementarity problems via nondifferentiable concave minimization." In: *Acta Mathematica Vietnamica* 22.1 (1997), pp. 199–205.

[36] J. Nocedal and S. J. Wright. *Numerical Optimization.* 2nd ed. Springer Series in Operations Research and Financial Engineering. New York: Springer Verlag, 2006. DOI: 10.1007/978-0-387-40065-5.

[37] I. Nowak. *Relaxation and decomposition methods for mixed integer nonlinear programming.* 2005. URL: urn:nbn:de:kobv:11-10038479.

[38] S. Sharma. "Mixed-integer nonlinear programming heuristics applied to a shale gas production optimization problem." Master's thesis. Norwegian University of Science and Technology, 2013.

[39] S. Sharma, B. R. Knudsen, and B. Grimstad. "Towards an objective feasibility pump for convex MINLPs." In: *Computational Optimization and Applications* 63.3 (2015), pp. 737–753. DOI: 10.1007/s10589-015-9792-y.

[40] R. E. Wendell and A. P. Hurter Jr. "Minimization of a non-separable objective function subject to disjoint constraints." In: *Operations Research* 24.4 (1976), pp. 643–657. DOI: 10.1287/opre.24.4.643.

Appendix A. Detailed Results for the MIPLIB 2010

Table 1: Detailed numerical results for the penalty ADM based feasibility pump on all 84 feasible MIPLIB 2010 instances; see Section 5.1. Objective function value, running times (s), outer penalty iterations (#Pen.), and inner ADM iterations (#ADM) for the method with $\lambda = 0.9$, multiplicative penalty parameter update rule $\mathrm{inc_m}$, activated Gurobi presolve, and deactivated local branching.

Instance	Objective	Time	#Pen.	#ADM
ran16x16	4734.00	0.12	52	144
rmatr100-p10	494.00	2.49	12	41
neos-1396125	—	—	—	—
netdiversion	—	—	—	—
msc98-ip	25797700.00	146.92	48	307
ns1208400	—	—	—	—
noswot	-38.00	0.17	61	240
satellites1-25	33.00	44.26	24	98
rococoC10-001000	34598.00	1.43	68	295
rmine6	-239.11	3.21	32	93
iis-pima-cov	74.00	3.23	12	44
csched010	—	—	—	—
mine-90-10	-558839000.00	1.69	58	173
triptim1	22.87	317.56	7	16
roll3000	18404.00	2.83	48	207
tanglegram2	1445.00	2.51	16	34
pigeon-10	-9000.00	2.51	1091	2732
neos-1337307	-201818.00	2.99	29	100
neos-1109824	687.00	3.93	93	295
sp98ir	244287000.00	4.73	49	168
neos13	-28.04	5.54	23	91
binkar10_1	7263.57	0.23	32	112
pw-myciel4	13.00	3.99	20	130
lectsched-4-obj	9.00	3.46	66	287
opm2-z7-s2	-1519.00	35.30	13	37
n3div36	170600.00	13.36	103	321
timtab1	1415540.00	0.27	41	153
rail507	183.00	30.66	35	158
pg5_34	-12628.50	0.76	65	195
acc-tight5	—	—	—	—
zib54-UUE	13164400.00	0.44	27	75
map20	-371.00	159.76	127	306

Table 1: Continued.

Instance	Objective	Time	#Pen.	#ADM
mik-250-1-100-1	284980.00	0.48	71	202
sp98ic	558066000.00	12.55	84	313
glass4	3390030000.00	0.28	103	353
danoint	78.00	1.70	24	101
bnatt350	—	—	—	—
mspp16	407.00	1399.15	91	249
mzzv11	-16438.00	94.84	102	440
vpphard	44.00	85.31	38	242
30n20b8	906.00	40.42	892	3828
cov1075	120.00	0.17	12	24
dfn-gwin-UUM	125512.00	0.28	66	166
n3seq24	53600.00	271.77	71	315
air04	71018.00	60.42	41	190
iis-100-0-cov	100.00	0.27	12	24
rmatr100-p5	1327.00	5.18	11	39
gmu-35-40	-2159090.00	1.89	298	1073
reblock67	-18739300.00	1.14	51	169
net12	337.00	29.06	53	217
map18	-280.00	250.17	120	317
bab5	-78587.20	34.25	178	540
neos18	17.00	0.87	21	86
neos-934278	264.00	1091.80	57	225
app1-2	-30.00	3189.86	4133	15567
neos-849702	—	—	—	—
tanglegram1	6478.00	114.05	16	35
m100n500k4r1	-20.00	0.76	30	203
eilB101	1513.00	0.94	17	66
bienst2	73.25	0.21	20	76
enlight13	0.00	0.00	1	1
neos-916792	44.56	5.64	100	335
mine-166-5	-22751900.00	1.16	22	54
neos-686190	16410.00	1.87	47	143
eil33-2	1373.60	0.74	19	64
mcsched	228737.00	1.37	13	56
neos-1601936	23409.00	266.56	66	629
n4-3	13980.00	0.28	13	35
biella1	3743460.00	65.50	26	238
ns1688347	35.00	6.14	49	227
rocII-4-11	-0.52	18.74	292	1162
neos-476283	407.01	66.98	78	226
ex9	0.00	0.00	1	1
unitcal_7	20426600.00	40.69	112	400
iis-bupa-cov	100.00	2.16	12	38
core2536-691	692.00	31.37	19	106
newdano	89.75	0.44	19	77
ns1758913	-1454.67	18.37	5	15
beasleyC3	863.00	0.19	24	67
qiu	1235.01	0.44	19	51
bley_xl1	285.00	4.62	37	149

Table 1: Continued.

Instance	Objective	Time	#Pen.	#ADM
ns1830653	—	—	—	—
aflow40b	3565.00	2.48	146	437
macrophage	522.00	0.32	14	30

Appendix B. Detailed Results for the MINLPLib

Table 2: Detailed numerical results for the penalty ADM based feasibility pump on all 263 MINLPLib instances with integer variables. Objective function value, running times (s), outer penalty iterations (#Pen.), and inner ADM iterations (#ADM) for the method with $\lambda = 0.9$, additive penalty parameter update rule inc_a, and deactivated local branching.

Instance	Objective	Time	#Pen.	#ADM
ghg_2veh	7.78	0.00	1	1
fo9	52.78	7.00	60	109
tln6	20.50	9.00	177	209
tls12	—	—	39937	48574
deb9	176.08	2.00	3	4
pb351575	6457260.00	13.00	15	21
watersbp	—	—	—	—
super2	—	—	—	—
nvs19	-1070.00	1.00	10	17
prob02	112235.00	1.00	10	13
st_miqp3	—	—	—	—
prob03	11.00	1.00	18	31
ex1266a	16.30	1.00	7	13
st_testgr1	-12.76	1.00	11	15
tln12	315.60	186.00	2860	3350
fo8_ar4_1	38.66	47.00	384	529
st_testgr3	-20.50	1.00	19	29
pb302055	4087020.00	156.00	155	200
nuclear14	-1.13	3.00	4	5
ex1223a	5.81	1.00	5	8
saa_2	12.75	35.00	50	78
st_e40	—	—	—	—
fuel	8566.12	2.00	16	42
gear4	120.67	4.00	70	87
ex1252a	143555.00	3.00	23	54
st_miqp4	—	—	—	—
nvs21	-4.27	9.00	172	174
o8_ar4_1	345.47	26.00	130	256
parallel	—	—	65803	65803
tln4	9.30	5.00	92	103
nous2	0.63	0.00	3	4
nvs08	23.83	1.00	10	14
qap	411560.00	6.00	13	28
m7_ar2_1	—	—	64246	64522

Table 2: Continued.

Instance	Objective	Time	#Pen.	#ADM
st_test6	664.00	4.00	46	75
fo7_ar4_1	41.66	1477.00	25281	25479
ex1263a	38.60	55.00	1101	1142
ex1225	31.00	1.00	6	9
nuclearvb	-1.02	0.00	1	1
pb302035	4052260.00	27.00	25	34
contvar	829318.00	7.00	8	35
deb8	176.08	3.00	3	4
gasnet	6999380.00	13.00	97	194
st_e31	—	—	—	—
nuclear49	-1.15	29.00	13	17
ex1224	-0.88	1.00	12	19
feedtray2	—	—	—	—
ex1264a	10.00	5.00	90	108
gear	0.00	1.00	3	4
gkocis	-1.41	0.00	7	9
ex1263	69.60	87.00	1665	1773
nuclear14a	-1.13	1.00	1	1
nvs13	-585.20	0.00	3	5
ghg_1veh	7.78	0.00	1	1
ex1252	—	—	—	—
netmod_dol2	-0.49	36.00	16	44
ravempb	269590.00	1.00	10	23
m3	55.80	2.00	21	32
netmod_dol1	-0.01	16.00	22	40
nuclear25a	-1.12	5.00	22	29
no7_ar3_1	149.22	13.00	142	201
st_test3	—	—	—	—
waterful2	—	—	—	—
ex1265	22.30	17.00	231	343
product	-2075.33	16.00	56	103
var_con5	285.87	1.00	3	4
ex1222	1.08	1.00	6	10
tls4	22.00	475.00	9538	9781
nvs22	6.06	0.00	4	6
ex3pb	103.58	2.00	19	37
sep1	-510.08	0.00	5	7
ghg_3veh	7.77	0.00	4	5
super1	—	—	—	—
prob10	3.45	0.00	3	4
o7	190.62	23.00	374	418
pb351555	5256790.00	26.00	21	41
water4	1323.97	68.00	1294	1381
nvs14	-40358.20	2.00	35	38
nuclear14b	-1.09	12.00	51	73
waterx	934.86	0.00	7	12
m7_ar4_1	450.97	122.00	1993	2166
st_test5	-110.00	2.00	28	46
tltr	54.60	1.00	23	32

Table 2: Continued.

Instance	Objective	Time	#Pen.	#ADM
st_test8	-29605.00	0.00	4	5
hda	-4322.55	5.00	18	39
st_testph4	-80.50	0.00	3	5
stockcycle	260786.00	12.00	147	218
fo9_ar25_1	—	—	57500	57912
windfac	0.25	2.00	26	28
st_e29	-0.88	1.00	12	19
water3	—	—	—	—
nvs23	-1078.40	1.00	3	5
beuster	128512.00	17.00	99	254
tloss	24.30	3.00	41	62
ex1244	95046.40	1.00	13	17
cecil_13	-115564.00	8.00	65	92
pb302075	4624370.00	63.00	49	77
nvs12	-477.00	0.00	3	5
nvs04	—	—	—	—
enpro48pb	188887.00	1.00	10	23
product2	—	—	—	—
var_con10	452.69	1.00	5	7
fo8_ar3_1	45.34	36.00	469	590
m7	—	—	65916	66203
m7_ar25_1	—	—	63515	63812
eg_disc_s	6.03	6.00	7	15
tls6	36.10	191.00	2418	2859
nous1	1.57	1.00	3	4
nuclearvc	-0.98	8.00	49	59
o7_2	161.38	17.00	147	281
ex1264	31.60	11.00	185	222
tls5	15.50	81.00	1319	1590
chp_partload	—	—	25022	25231
fac3	34789500.00	3.00	23	53
nvs18	-778.40	1.00	3	5
nuclearvd	-1.04	6.00	63	75
fo8	47.40	6.00	66	99
oaer	-1.92	0.00	4	5
lop97icx	4590.48	3.00	13	31
uselinear	—	—	—	—
eg_disc2_s	5.68	3.00	3	4
ex1243	135552.00	2.00	18	47
watersym1	—	—	—	—
space25	919.99	283.00	3235	4234
csched1a	-29903.30	0.00	7	13
st_e35	71468.10	1.00	3	4
ex1221	7.67	0.00	1	1
o7_ar4_1	182.29	69.00	971	1138
synthes2	80.29	1.00	17	34
watersym2	—	—	—	—
nuclearva	-1.01	4.00	33	45
st_e14	5.81	0.00	6	10

Table 2: Continued.

Instance	Objective	Time	#Pen.	#ADM
nuclear10a	—	—	1744	2063
pb351535	5474850.00	180.00	240	306
st_miqp2	2.00	1.00	12	16
eniplac	-120713.00	42.00	636	764
super3t	—	—	—	—
nuclear49b	-1.13	131.00	25	74
nvs20	230.92	1.00	7	13
gbd	2.20	0.00	1	1
st_e38	7197.73	0.00	1	1
nvs09	-43.13	0.00	1	1
fo7_ar3_1	34.49	13.00	97	212
nuclear49a	-1.15	25.00	25	33
no7_ar4_1	188.94	129.00	2040	2246
deb10	209.43	1.00	9	13
ex1223b	5.81	0.00	6	10
nvs02	5.96	2.00	35	38
fo8_ar2_1	—	—	63425	63800
nuclear10b	-1.15	1627.00	31	124
st_e32	-0.72	14.00	243	289
lop97ic	4535.18	9.00	10	31
util	999.84	2.00	34	39
nvs01	13.10	1.00	5	7
netmod_kar2	0.00	2.00	17	31
csched2	-160668.00	2.00	9	20
oil2	-0.73	0.00	3	4
o7_ar2_1	—	—	66427	66695
st_test4	—	—	—	—
meanvarxsc	—	—	—	—
hmittelman	16.00	2.00	22	39
tln7	39.00	165.00	3173	3239
minlphix	345.51	3.00	53	63
st_test1	0.00	1.00	6	11
risk2bpb	-55.48	2.00	17	21
nvs17	-1078.20	3.00	52	57
synthes3	82.37	2.00	14	28
o7_ar25_1	160.47	165.00	2936	3100
gastrans	—	—	73821	73824
fo9_ar5_1	51.17	11.00	72	131
m7_ar3_1	—	—	67526	67807
fo7_ar5_1	38.09	7.00	53	110
ex4	659.57	4.00	46	77
st_e13	—	—	—	—
fo9_ar2_1	—	—	59417	59843
m7_ar5_1	511.21	58.00	857	1025
batchdes	185769.00	1.00	7	13
ex1233	201540.00	3.00	21	49
fo8_ar5_1	51.84	352.00	5769	5963
spectra2	14.04	3.00	25	53
waste	1169.08	21.00	82	139

Table 2: Continued.

Instance	Objective	Time	#Pen.	#ADM
nvs11	-416.40	3.00	69	74
dosemin3d	1.32	30.00	7	13
blendgap	-0.00	1.00	1	1
st_e27	2.00	0.00	1	1
ex1265a	15.10	5.00	60	87
gear3	0.00	0.00	3	4
nvs15	1.00	0.00	3	4
nuclear25	-1.12	8.00	22	29
space960	7985000.00	104.00	39	111
nvs16	14.20	5.00	55	108
nuclearve	-1.02	1.00	7	9
o7_ar3_1	162.83	33.00	446	562
st_e36	—	—	8503	53584
elf	1.68	0.00	1	1
fac1	172954000.00	10.00	85	185
feedtray	-13.41	0.00	1	1
spring	1.35	2.00	19	27
pb351595	11847500.00	907.00	919	1524
batch	309205.00	2.00	22	43
waters	—	—	—	—
procsel	-1.41	0.00	5	7
super3	—	—	—	—
st_miqp5	-333.89	0.00	1	1
johnall	-222.37	4.00	54	57
mbtd	5.58	65.00	13	41
qapw	395664.00	46.00	106	345
nvs24	-1001.00	0.00	7	14
4stufen	118114.00	9.00	74	169
no7_ar2_1	—	—	53460	53715
nvs07	4.00	0.00	5	6
st_e15	—	—	—	—
fo7_2	40.43	13.00	171	236
nvs06	1.86	1.00	3	4
st_test2	—	—	—	—
o9_ar4_1	341.57	52.00	497	691
nvs10	-308.40	0.00	4	7
dosemin2d	173.98	11.00	5	11
nvs03	16.00	0.00	11	14
oil	-0.87	1.00	3	4
nuclear104	—	—	—	—
no7_ar5_1	153.84	172.00	2440	2576
csched2a	-162047.00	7.00	44	120
alan	3.00	1.00	5	7
ex1223	5.81	0.00	6	10
pb302095	—	—	3127	4614
deb7	176.08	2.00	3	4
fuzzy	—	—	1	10716
nuclear25b	-1.09	30.00	63	153
ex1266	—	—	72715	74554

Table 2: Continued.

Instance	Objective	Time	#Pen.	#ADM
tls2	11.30	211.00	3385	3543
st_miqp1	380.50	1.00	15	28
enpro56pb	266762.00	3.00	24	56
o7_ar5_1	166.99	80.00	1274	1443
gear2	0.01	0.00	4	6
fo7	25.60	87.00	1643	1691
tln2	23.30	16.00	284	313
nuclearvf	—	—	—	—
du-opt	5.34	0.00	3	4
deb6	251.66	1.00	3	4
fac2	407585000.00	5.00	35	97
ex1226	-17.00	0.00	1	1
eg_all_s	8.67	13.00	20	44
synheat	219858.00	2.00	18	43
synthes1	7.09	1.00	8	14
fo7_ar2_1	55.38	307.00	5449	5627
tln5	16.50	4.00	47	65
fo9_ar4_1	—	—	57027	57468
nvs05	5.47	0.00	3	4
ortez	-9532.04	1.00	13	20
fo7_ar25_1	—	—	67319	67538
fo9_ar3_1	49.40	45.00	446	592
du-opt5	112.02	1.00	5	8
meanvarx	14.37	2.00	23	26
space25a	657.73	6.00	80	106
tls7	48.80	1641.00	27458	28995
csched1	-30174.60	0.00	4	5
netmod_kar1	0.00	2.00	17	31
eg_int_s	8.32	6.00	6	13
m6	123.98	11.00	105	206
no7_ar25_1	175.87	157.00	2635	2871
waterz	315610.00	1544.00	13551	26704
fo8_ar25_1	—	—	62646	62978

[1]Björn Geissler, Antonio Morsi, Lars Schewe, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Discrete Optimization, Cauerstr. 11, 91058 Erlangen, Germany, [2]Martin Schmidt, (a) Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Discrete Optimization, Cauerstr. 11, 91058 Erlangen, Germany; (b) Energie Campus Nürnberg, Fürther Str. 250, 90429 Nürnberg, Germany

E-mail address: [1]`{bjoern.geissler,antonio.morsi,lars.schewe}@math.uni-erlangen.de`

E-mail address: [2]`mar.schmidt@fau.de`