

# Three ideas for a Feasibility Pump for nonconvex MINLP

Pietro Belotti · Timo Berthold

Received: date / Accepted: date

**Abstract** We describe an implementation of the Feasibility Pump heuristic for nonconvex MINLPs. Our implementation takes advantage of three novel techniques, which we discuss here: a hierarchy of procedures for obtaining an integer solution, a generalized definition of the distance function that takes into account the nonlinear character of the problem, and the insertion of linearization cuts for nonconvex constraints at every iteration. We implemented this new variant of the Feasibility Pump as part of the global optimization solver COUENNE. We present experimental results that compare the impact of the three discussed features on the ability of the Feasibility Pump to find feasible solutions and on the solution quality.

**Keywords** Feasibility Pump · MINLP · Hessian

## 1 Introduction

In this paper, we describe a new variant of the Feasibility Pump algorithm to find feasible solutions for mixed integer nonlinear programming (MINLP) problems. We consider an implementation of the Feasibility Pump within a global solver, in our case COUENNE.

---

P. Belotti  
Fair Isaac Europe Ltd, International Square, Starley Way, Birmingham B37 7GN, United Kingdom  
E-mail: pietrobelotti@fico.com

T. Berthold  
Fair Isaac Germany GmbH, Takustr.7, 14195 Berlin, Germany  
E-mail: timoberthold@fico.com

An MINLP is an optimization problem of the form

$$\begin{aligned}
 \min \quad & c^\top x \\
 \text{s.t.} \quad & g_j(x) \leq 0 \quad \text{for } j = 1, \dots, m, \\
 & L_i \leq x_i \leq U_i \text{ for } i = 1, \dots, n, \\
 & x_i \in \mathbb{Z} \quad \text{for } i \in \mathcal{I},
 \end{aligned} \tag{1}$$

where  $\mathcal{I} \subseteq \{1, \dots, n\}$  is the index set of the integer variables,  $d \in \mathbb{R}^n$ ,  $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$  for  $j = 1, \dots, m$ , and  $L \in (\mathbb{R} \cup \{-\infty\})^n$ ,  $U \in (\mathbb{R} \cup \{+\infty\})^n$  are lower and upper bounds on the variables, respectively. Since fixed variables can always be eliminated, we assume w.l.o.g. that  $L_i < U_i$  for  $i = 1, \dots, n$ , i.e., that the interior of  $[L, U]$  is nonempty. Note that a nonlinear objective function can always be reformulated by introducing one additional constraint and variable. We chose the convention of assuming a linear objective for the ease of presentation; the presented algorithm and the conducted experiments address general MINLPs.

There are many subclasses of MINLP; the following four will play a role in this article:

- If all constraint functions  $g_j$  are convex, problem (1) is called a *convex MINLP*.
- If all constraint functions  $g_j$  are affine, problem (1) is called a *mixed integer linear program (MIP)*.
- If  $\mathcal{I} = \emptyset$ , problem (1) is called a *nonlinear program (NLP)*.
- If  $\mathcal{I} = \emptyset$  and all  $g_j$  are affine, problem (1) is called a *linear program (LP)*.

Primal heuristics are algorithms that try to find feasible solutions of good quality for a given optimization problem within a reasonably short amount of time. They are incomplete methods, hence there is usually no guarantee that they will find any solution, let alone an optimal one.

For mixed-integer (linear) programming it is well-known that general-purpose primal heuristics are able to find high-quality solutions for a wide range of problems [8, 27]. For mixed-integer nonlinear programming, research in the last five years has shown an increasing interest in general-purpose primal heuristics [9–12, 15, 16, 19, 20, 33, 36, 37].

Discovering good feasible solutions at an early stage of the MINLP solving process has several advantages:

- The bounding step of the branch-and-bound [32] algorithm depends on the quality of the incumbent solution; a better primal bound leads to more nodes being pruned and hence to smaller search trees.
- The same holds for presolving and domain propagation strategies such as reduced cost fixing. Better solutions can lead to tighter domain reductions. Besides cut generation, bound tightening is one of the most important features for an efficient implementation of a branch-and-bound algorithm for nonconvex MINLPs, see, e.g., [6, 29].

- In practice, it is often sufficient to compute a heuristic solution whose objective value is within a certain quality threshold. For hard MINLPs that cannot be solved to optimality within a reasonable amount of time, it might still be possible to generate good primal solutions quickly.
- Improvement heuristics such as RINS [21, 14, 12] or Local Branching [26, 37, 12] need a feasible solution as starting point.

Feasibility Pump (FP) algorithms follow the idea of decomposing the set of constraints of a mathematical programming problem into two parts: integer feasibility and constraint feasibility. At least for MIP and convex MINLP, both are “easy” to achieve: the former by rounding, the latter by solving an LP or a convex NLP, respectively. Each can be done in polynomial time. Two sequences of points  $\{\tilde{x}^k\}_{k=1}^K$  and  $\{\bar{x}^k\}_{k=1}^K$ , for  $K \in \mathbb{Z}_{\geq 0}$ , are generated such that  $\tilde{x}$  contains integral points that may violate constraints  $g_j(x) \leq 0$  for one or more  $j = 1, 2, \dots, m$  and  $\bar{x}$  contains points that are feasible for a continuous relaxation to the original problem but might not be integral. These two sequences are related to each other in that the points of  $\{\tilde{x}^k\}$  are obtained through rounding of the integral coordinates of points in  $\{\bar{x}^k\}$  and these, in turn, are obtained via projections of points in  $\{\tilde{x}^k\}$ .

One focus of this chapter is the application of a Feasibility Pump inside a global solver such as BARON [41], COUENNE [6], LINDOGLOBAL [34], BONMIN [14] or SCIP [2], whereas previous publications considered the Feasibility Pump as a standalone procedure. This has an impact on the design choices carried out in developing the heuristic, in particular balancing efficiency versus completeness. If primal heuristics are applied as supplementary procedures inside a global solver, the overall performance of the solver is the main objective. To this end, it is often worth sacrificing success on a small number of instances for a significant saving in average running time. The enumeration phase of the Feasibility Pump presented in [7] is a typical example of a component that is crucial for its impressive success rate as a standalone algorithm, but it will not be applied when the Feasibility Pump is used inside a global solver, see [8].

This paper features three novel contributions aimed at a more flexible use of a Feasibility Pump within a nonconvex MINLP solver:

- (i) a hierarchy of rounding procedures, ranked by efficiency (typically converse to the quality of the provided points) for finding integral points; which routine to use is decided and automatically adapted at runtime;
- (ii) an improved and parameterized distance function for the rounding step, taking into account second-order information of the continuous NLP relaxation; and
- (iii) the separation of linearization cuts that approximate the *convex envelope* of the nonconvex feasible set of the NLP relaxation of (1), as opposed to employing Outer Approximation cuts [24] only for the convex part of the problem. We know of no other FP that generates linearization cuts.

## 2 Feasibility pumps for MINLP

The *Feasibility Pump* (FP) algorithm was originally introduced by Fischetti, Glover, and Lodi in 2005 [25] for mixed binary programs, i.e., for the special case of MIPs in which  $l_j = 0$  and  $u_j = 1$  for all  $j \in \mathcal{I}$ . The principal idea is as follows. The LP relaxation of a MIP is solved. The LP optimum  $\bar{x}$  is then rounded to the closest integral point. This part of the FP algorithm is called the *rounding step*. If  $\tilde{x}$  is not feasible for the linear constraints, the objective function of the LP is changed to an  $\ell_1$  distance function:

$$\Delta(x, \tilde{x}) := \sum_{j \in \mathcal{I}} |x_j - \tilde{x}_j| = \sum_{j \in \mathcal{I}: \tilde{x}_j=0} x_j + \sum_{j \in \mathcal{I}: \tilde{x}_j=1} (1 - x_j) \quad (2)$$

and a new  $\bar{x}$  is obtained by minimizing  $\Delta(x, \tilde{x})$  over the LP relaxation of the MIP. The process is iterated until  $\tilde{x} = \bar{x}$  which implies feasibility (w.r.t. the MIP). The operation of obtaining a new  $\bar{x}$  from  $\tilde{x}$  is known as the *projection step*, as it consists of projecting  $\tilde{x}$  to the feasible set of a continuous relaxation of the MIP along the direction  $\Delta(x, \tilde{x})$ .

The original work by Fischetti, Glover, and Lodi [25] led to many follow-up publications [7, 3, 28, 30, 4, 13, 23, 22] on Feasibility Pumps for MIP that improve the performance mostly by extending one of the two basic steps of rounding and projection. Most important for the present paper is the work by Achterberg and Berthold [3] who suggest to replace function (2) by a convex combination of (2) and the original objective  $c^\top x$  in order to produce higher-quality solutions.

Versions of the Feasibility Pump for convex MINLP have been presented by Bonami et al. [15] and Bonami and Gonçalves [16]. The only publication for nonconvex MINLP that we are aware of is D'Ambrosio et al. [20]. Bonami and Gonçalves [16] keep the rounding step as in [25] and replace solving an LP in the projection phase by solving a convex NLP, using again the distance function (2) as an objective. Recently, Sharma et al. [38, 39] presented an integration of the Objective Feasibility Pump idea by Achterberg and Berthold and the algorithm of Bonami and Gonçalves: a scaled sum of the distance function and the original objective is used as an objective for the convex NLP. In [15], the authors suggest using an  $\ell_2$  norm for the projection step. Further, their implementation of the rounding step differs significantly from all previous FP variants. Instead of performing an instant rounding to the nearest integer, they solve an MIP relaxation which is based on an outer approximation [24] of the underlying MINLP. The particular difficulty addressed in [20] is that of handling the nonconvex NLP relaxation if adapting the algorithm of [15] to the nonconvex case. The authors suggest using a stochastic multistart approach, feeding the NLP solver with different randomly generated starting points, and solving the NLP to local optimality as if it was a convex problem. In the following three sections, we present our suggestions to extend the Feasibility Pump for nonconvex MINLP. The first one mainly aims at saving computation time, while the second and the third incorporate the nonlinear problem structure directly into the FP algorithm.

### 3 A hierarchy of rounding procedures

In the preceding publications on Feasibility Pump heuristics, several ideas have been proposed to generate good integer points during the rounding step. The original work [25] and one of the nonlinear FPs [16] suggest plain rounding in order to get a point which is closest w.r.t. the  $\ell_2$  norm. The Feasibility Pump 2.0 [28] uses an iterated “round-and-propagate” procedure in order to get a point which is close w.r.t. the  $\ell_2$  norm but “more feasible” for the relaxation. The nonlinear Feasibility Pumps suggested by [15, 20] solve an MIP to get the closest integer point w.r.t. the  $\ell_1$  norm that fulfills *all* constraints of the linear relaxation.

The observed shift from “stay close to the previous point” to “stay close, but also fulfill the relaxation” leads us to the idea of trying several rounding procedures which address these goals in various ways. Similar to other FPs for MINLP, the rounding step is performed by solving a MIP that is a relaxation of the original MINLP. In convex MINLP, such a MIP can be obtained by applying Outer Approximation [24] and maintaining integrality constraints, while in the nonconvex case the MIP is obtained as a LP relaxation of the MINLP with the integrality constraints.

In order to obtain a point  $\tilde{x}$  that satisfies the integrality constraints, we select a procedure from the following list:

- (i) Solve the MIP relaxation with a node limit and an emphasis on good solutions.
- (ii) Solve the MIP relaxation with a node limit (smaller than in (i)), disabling time-consuming cutting plane separation, branching and presolving strategies.
- (iii) Solve the root node of the MIP relaxation of method (i), then apply the RENS heuristic, see [10].
- (iv) Solve the root node of the MIP relaxation, then apply the Objective Feasibility Pump 2.0 [28] for MIPs.
- (v) Apply the round-and-propagate algorithm of [28]: selectively round a fractional variable to the nearest integer and then apply a domain propagation to restrict the feasible set, until either a feasible solution is found or the remaining MIP is proven infeasible. In the latter case, round all remaining unfixed variables to the nearest integer.
- (vi) Choose an integral point from a solution pool (e.g. from suboptimal solutions of applying procedure (i)), see below.
- (vii) Apply a random perturbation to  $\tilde{x}^{k-1}$  and round to the nearest integer  $\tilde{x}^k$ .

Note that the integral points generated by procedures (v) to (vii) might be infeasible for the MIP problem of the current iteration. While procedures (i) to (iv) might terminate without an integral point, procedure (v) will always produce an integral point. The last two procedures are implemented solely to avoid cycling, i.e. for the case that a point constructed by procedure (v) is identical to one visited in a previous iteration. Further, procedures (v) to (vii)

are ordered by their expected running time and the expected likelihood to produce a good integral point. The higher in the list a procedure is, the more expensive it is and the more likely it is to produce high-quality (w.r.t. the distance function) solutions.

Having this variety of options to produce integral points, the question remains *which* to apply *when*. At the first iteration of the FP, we employ procedure (ii). If the current procedure successfully provides us with a “new” (not yet visited) integer point for three iterations in a row, we proceed with the next (cheaper and less aggressive) method of the above list. If, at any iteration, the current procedure does not terminate within the given limits or produces a point that was already visited, we proceed with the previous (more expensive and more powerful) method of the above list. Note that the list of procedures from (i) to (vii) has decreasing complexity and, in general, declines in solution quality. At subsequent iterations of the FP we use the procedure that was successful previously, but switch down to a cheaper routine when there were three successful iterations in a row.

In principle, the list could be prepended at the head to include methods that capture constraint feasibility even better (and are most likely more time consuming), for instance a convex (nonlinear) relaxation of the nonconvex MINLP. However, computational results presented in [20] indicate that this results in a significant computational overhead with little impact.

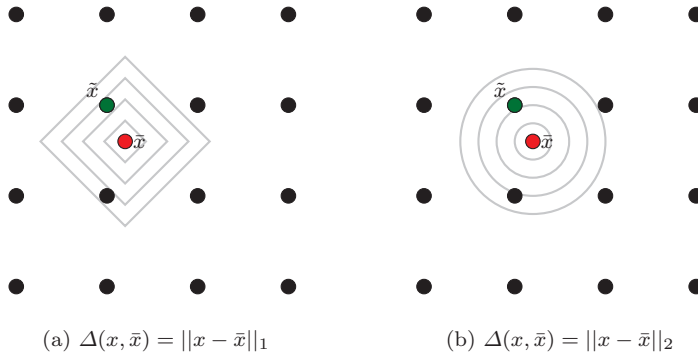
If procedures (i), (ii), or (iii) are used for the rounding step, these may produce more than one MIP-feasible solution. The suboptimal points might be used for later iterations, and are therefore stored in a solution pool. This approach is motivated by two observations: first, in (i) we solve similar MIPs over and over again, mainly using a different objective (plus some new cuts). Each known feasible point from a previous call may be used as a starting solution in subsequent calls. Procedures (iii) and (iv) will also benefit from a given upper bound as they consider a restricted search space. Second, considering a point which was initially a candidate, but was then discarded carries more information about the problem than one generated through random perturbation. Thus, the previously collected points are used in (vi). We rank the points in the pool by the value of the distance function at the current iteration.

#### 4 Follow the Hessian: an improved distance function

Feasibility Pumps for MIP use the  $\ell_1$  norm for the projection and the rounding step; the nonlinear FPs by Bonami et al. [15] and D’Ambrosio et al. [20] use  $\ell_2$  for the projection and  $\ell_1$  for rounding. De Santis et al. [22] present variants of the distance function that help draw a connection with the Frank-Wolfe algorithm and that show some computational advantage w.r.t. the more traditional distance functions. Either way, exclusively using these norms as objective functions for auxiliary optimization problems ignores the fact that a “close” solution is not necessarily a good one: the original objective function is

completely neglected and no information is used on how much the constraints are violated.

To overcome this issue, we use the norm of a vector obtained from a linear transformation applied to  $x - \bar{x}^k$ . As a motivating example, consider first an unconstrained integer nonlinear optimization problem  $\min\{f(x) : x \in \mathbb{Z}^n\}$ , where  $f \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R})$ . Note that this example diverges from our assumed MINLP formulation in that it uses a nonlinear objective function. We find this digression necessary to explain the idea expressed in this section, which ultimately refers to the MINLP standard we specified in (1). Assume that  $\bar{x}$  is a local optimum of the continuous relaxation  $\min\{f(x) : x \in \mathbb{R}^n\}$ . Level curves of the  $\ell_1$  and the  $\ell_2$  distance functions w.r.t.  $\bar{x}$  are given in Figure 1. In either norm,  $\tilde{x}$  is the closest integer point (the result is norm-invariant on  $\mathbb{Z}^n$ ).



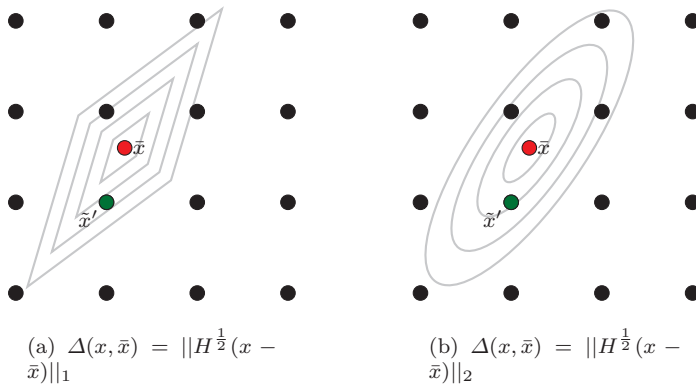
**Fig. 1** Level curves (gray) of different norm functions  $\Delta(x, \bar{x})$ . The closest integer point to  $\bar{x}$  is  $\tilde{x}$ . In 1(a) and 1(b), the norm  $\|\cdot\|_p$  is used for  $p = 1$  and  $p = 2$  respectively.

Now consider the second-degree Taylor series approximation of  $f$  at  $\bar{x}$ :

$$f(x) \approx f(\bar{x}) + \nabla f|_{\bar{x}}^\top (x - \bar{x}) + \frac{1}{2}(x - \bar{x})^\top H(x - \bar{x}) \quad (3)$$

which is convex and quadratic. We want to use (3) for constructing an improved distance function which uses information about  $f$ . Since the problem is unconstrained, the gradient of  $f$  is null, i.e.,  $\nabla f|_{\bar{x}} = 0$ , and its Hessian is positive semidefinite, i.e.,  $H = \nabla^2 f|_{\bar{x}} \succeq 0$ . Thus, minimizing (3) is equivalent to minimizing  $(x - \bar{x})^\top H(x - \bar{x})$  given that the first two terms can be ignored.

As shown in Figure 2(b), the level curves of this new function are ellipsoids whose axes and axis lengths are defined by the eigenvectors and eigenvalues of  $H$ . Note that this relation is inverse proportional, i.e., the larger the eigenvalue, the steeper the ascent, the *shorter* the corresponding axis. A convex, piecewise linear approximation of this objective function is  $\|H^{\frac{1}{2}}(x - \bar{x})\|_1$ . Its level curves are represented in Figure 2(a). This is a distance function that incorporates information about the original objective function. Both functions that are



**Fig. 2** Level curves (grey) of different norm functions using second order information associated with the objective function of the original problem. The distance function in 2(b) and its piecewise linear approximation in 2(a) both lead to  $\tilde{x}'$  as best integral point in the vicinity of  $\bar{x}$ .

displayed in Figure 2 find  $\tilde{x}'$  as best (w.r.t. the Hessian) integral point near  $\bar{x}$ . Note that, in general, neither  $\|H^{\frac{1}{2}}(x - \bar{x})\|_1$  nor  $\|H^{\frac{1}{2}}(x - \bar{x})\|_2$  yield a minimum in the hypercube  $[\lfloor \bar{x} \rfloor, \lceil \bar{x} \rceil]$  containing  $\bar{x}$ . Also, the closest point changes with the norm as one could construct an example where  $\operatorname{argmin}_{x \in \mathbb{Z}^n} \|H^{\frac{1}{2}}(x - \bar{x})\|_p$  depends on  $p$ .

The advantage of using the Hessian  $H$ , which incorporates second-order information about the current optimal solution of the nonlinear problem, is that a minimizer of  $\|H^{\frac{1}{2}}(x - \bar{x})\|_1$ , while possibly far from  $\bar{x}$  in terms of the  $\ell_1$  norm, corresponds to an integer point whose objective function is close to that of  $\bar{x}$ , hence providing a “good” solution from the objective function standpoint.

Let us now generalize this to the constrained version. If we consider an MINLP with a nonlinear objective function, the Hessian of the objective function is, in general, indefinite at the optimum  $\bar{x}$  of the relaxation as there might be active constraints. In case of a linear objective (as in our definition (1)), the Hessian is constant zero. As an alternative, we use the Hessian of the Lagrangian function of the NLP relaxation of the original MINLP:

$$\tilde{H} = \left( \frac{\partial^2 \mathcal{L}}{\partial x_i \partial x_j} \right)_{i,j=1,2,\dots,n}.$$

While this is a straightforward generalization of the unconstrained case, it must be pointed out that there might be one or more active constraints at a local optimum. Sufficient conditions for optimality in the constrained case dictate that  $\tilde{H}$  is only positive semidefinite on the null space of the gradient [5], i.e., in  $\{x \in \mathbb{R}^n : c^\top x = 0\}$ . We hence correct  $\tilde{H}$  as follows: if an eigenvalue decomposition of  $\tilde{H}$  has  $q$  negative eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_q$  whose eigenvectors



have non-zero component in the null space of the gradient, we replace each of those eigenvalues with a value that is higher than the maximum eigenvalue of  $\tilde{H}$ . This allows us to focus the search on solutions that are closer to the active constraints rather than along the direction expressed by  $c$ .

**Definition 1 (Hesse-distance)** Let  $\tilde{H} \in \mathbb{R}^{n \times n}$  be the Hessian of the Lagrangian (possibly modified as described above and a reference point  $\bar{x} \in [l, u]$  be given. We call

$$\tilde{\Delta}_{\text{int}}(x, \bar{x}) = \|\tilde{H}^{\frac{1}{2}}(x - \bar{x})\|_1$$

the  $\ell_1$  Hesse-distance of  $x$  to  $\bar{x}$ .

We suggest to incorporate the Hesse-distance into the objective functions of the auxiliary MIPs that are solved in the rounding step of the nonlinear Feasibility Pump. While there have been several publications for improving the distance function for the projection step [3,15,20,22], this is, up to our knowledge, the first attempt of drawing information from the constraint functions into the distance function definition of a Feasibility Pump algorithm. In the spirit of the Objective Feasibility Pump, we came up with the following combination:

$$\Delta_{\text{int}}(x, \bar{x}) = \alpha_{\text{dist}}\|x - \bar{x}^k\|_1 + \alpha_{\text{H}}\tilde{\Delta}_{\text{int}}(x, \bar{x}) + \alpha_{\text{orig}}c^{\text{T}}x.$$

Typically, one would increase  $\alpha_{\text{dist}}$  in every iteration, making it converge to one and fade out the other two, thereby shifting the focus from solution quality towards pure feasibility. Note that for any value of these parameters the objective function is piecewise linear and convex. Note also that one can easily extend the definition of the Hesse-distance to the Euclidean case and incorporate it into the objective function for the projection phase. Preliminary experiments revealed, however, that this is not beneficial.

## 5 Separation of linearization cuts and postprocessing

Techniques to generate a linear relaxation of a MINLP can be used in an incremental fashion as a *separation* procedure: given a solution  $\tilde{x}$  to a MIP relaxation that is not feasible for the MINLP itself, find a linear cut  $a^{\text{T}}x \leq d$  that is fulfilled by all solutions of the MINLP, but  $a^{\text{T}}\tilde{x} > d$  (or show that no such inequality exists). LP-based branch-and-bound solvers for MINLP, such as COUENNE or SCIP, typically solve such separation problems to improve local dual bounds at each node. For details on branch-and-cut for MINLP, see, e.g., [40,42]. In marked difference to previous nonlinear Feasibility Pumps, our implementation also separates linear over- and underestimators for nonconvex functions and not exclusively Outer Approximation cuts for convex parts of the problem.

A typical problem occurring in iterative heuristics such as the FP is cycling. Some versions prevent cycling by adding no-good cuts. Our FP variant

attempts to avoid cycling in two ways. First, linear inequalities for nonconvex MINLPs are added to eliminate infeasible integer points. For convex constraints, Outer Approximation cuts are added. For nonconvex constraints, the situation is more involved. As for most global optimization solvers COUENNE uses the standard approach of reformulating nonconvex constraints via an expression tree whose nodes are variables and elementary nonlinear functions. For these nonlinear functions, underestimators are used to produce valid linear relaxations. For instance, nonconvex bilinear terms can be addressed via McCormick underestimators [35]. This might still allow for permanently separating the MIP solution from the feasible region.

However, when the MIP terminates with an optimal solution that is infeasible for the (nonconvex) MINLP, but inside the convex hull of its feasible set, no linear cut can be added to separate the solution from its feasible set. This leads to the second way of avoiding cycling: we forbid particular assignments to the integer variables by the use of a tabu list.

A final component of our Feasibility Pump implementation is the postprocessing of feasible solutions. If a MINLP feasible solution  $\tilde{x}$  is found, the values for the continuous variables are only optimal for the distance objective used at the last iteration. To check whether there are better solutions, we run a simple local search improvement heuristic: we obtain a restriction of the original MINLP by fixing all integer variables to the values of the Feasibility Pump solution  $\tilde{x}$  and solve it to local optimality with a convex NLP solver such as IPOPT.

## 6 Computational experiments

We implemented the Feasibility Pump within COUENNE 0.4.7 [6], which is a MINLP solver that uses CBC 2.8.9 [18] as a branch-and-bound manager. Within our implementation, the auxiliary MIP problems are solved by SCIP 3.0.2 [2], linked against SOPLEX 1.7.2 [44]. The auxiliary NLPs are solved by IPOPT 3.11.7 [43, 31]. The results were obtained on a cluster of 64bit Intel Xeon X5672 CPUs at 3.20 GHz with 12 MB cache and 48 GB main memory, running an OPENSUSE 12.3 with a GCC 4.7.2 compiler. Turboboost was disabled. In all experiments, we ran only one job per node to reduce fluctuations in the measured running times that might be caused by interference between jobs that share resources, in particular the memory bus.

As a test set, we used the MINLPLIB [17]. We excluded all instances that were solved by COUENNE before the Feasibility Pump was called or that got linearized by presolving, and four instances for which COUENNE triggered an error. This resulted in a test set of 218 instances. We compared the following six different settings of the nonlinear Feasibility Pump:

- **default** uses a Manhattan distance function, without contributions of the Hessian of the Lagrangian or the original objective, i.e.,  $\alpha_H = \alpha_{\text{orig}} = 0$ ; this setting does not add convexification cuts for non-convex parts of the problem; the auxiliary MIP is always solved by running SCIP with a stall

node limit of 1000 and aggressive heuristic settings; this resembles the Feasibility Pump of [20].

- **cuts** uses cuts for nonconvex parts of the problem (in addition to standard MIP cuts and Outer Approximation cuts); otherwise the same as default;
- **hierarchy** uses different algorithms to solve the MIP in different iterations of the Feasibility Pump, see Section 3; otherwise the same as default;
- **hessian** constructs the objective of the auxiliary MIP as a combination of the Manhattan distance and the Hessian of the Lagrangian, see Section 4; we chose  $\alpha_{\text{dist}} = 1 - 0.95^k$  and  $\alpha_{\text{H}} = 0.95^k$  at the  $k$ -th iteration; otherwise the same as default;
- **objective** constructs the objective of the auxiliary MIP as a combination of the Manhattan distance, the Hessian of the Lagrangian and a linear approximation of the original objective; we chose  $\alpha_{\text{dist}} = 1 - 0.95^k$ ,  $\alpha_{\text{H}} = 0.95^k$ , and  $\alpha_{\text{orig}} = 0.9^k$  at the  $k$ -th iteration, otherwise the same as default;
- **simple** applies rounding to the nearest integer instead of solving an auxiliary MIP in the rounding phase, compare [16].

Table 1 shows aggregated results of our experiments. For each of the six settings, we give three performance indicators: **feas**, the number of instances (out of 218) for which this setting found a feasible solution; **bet : wor**, the number of instances for which this setting found a better/worse solution (in terms of the objective function value) as compared to the default setting; and **time**, the running time in shifted geometric mean (see, e.g., [1, 9]), including COUENNE’s presolving and reformulation algorithms being applied, i.e., the time needed for processing the root node with one single call to the Feasibility Pump algorithm. We used a shift of 10 seconds for the running time.

**Table 1** Performance of different Feasibility Pump versions for a single call at the root node (aggregated results)

setting	feas	bet : wor	time
default	150	–	14.9
cuts	155	24 : 48	13.6
hierarchy	157	23 : 20	14.0
hessian	154	25 : 16	22.8
objective	138	45 : 30	23.9
simple	97	17 : 78	12.1

First of all, we observe that each of the five non-default settings outperforms the default in at least one of the three measures of performance. The hierarchy setting is the only one to outperform the default in all three measures, albeit only slightly in the quality of the solution found.

We further see that the cuts and the hessian setting both lead to slightly more solutions being found. This could be expected since both attempt to better incorporate the structure of the nonlinear feasibility region into the auxiliary MIP formulation. While using the Hessian leads to better solutions

being found, applying cutting plane separation for nonconvex constraints deteriorates the quality of the solution found more often than it improves it. Computing the Hessian of the Lagrangian itself might take considerable time, both settings that make use of this feature show an increase in running time by more than 50 %.

Similar to results for linear Feasibility Pumps, using the original objective often leads to better solutions being produced by the Feasibility Pump, but at the same time reduces the number of solutions being found. Note that the `bet : wor` statistic includes those cases for which only one of the settings found a solution. Finally, the “simple” setting, which does not use an auxiliary MIP at all, is only slightly faster than the default setting, but much worse in terms of found solutions and solution quality. It can be seen as an implementation of [16] for nonconvex MINLP.

In our implementation, we observe another behavior typical of Feasibility Pumps: although they are very successful in finding feasible solutions (about 75 % of the instances for the hierarchy setting), these solutions are often of a mediocre quality. In geometric mean, the primal gap (as defined in, e.g., [9]) of the solutions found was 34 %. This is almost in line with the Objective Feasibility Pump for MIP that was shown in [3] to reduce the average gap from 55 % for the original Feasibility Pump to 29.5 %.

From our results we conclude that within a global solver, one should use the hierarchy setting and, at least in the initial runs of the FP (i.e., in the first BB nodes), a dynamic setting of  $\alpha_H$  that makes use of the Hesse-distance. For a standalone FP or when there is an emphasis on feasibility, a concurrent run between the hierarchy (most solutions) and the objective setting (highest solution quality) seems most promising.

## 7 Conclusion

We presented and evaluated three novel ideas for solving nonconvex MINLPs with a Feasibility Pump: the generation of valid cutting planes for nonconvex nonlinearities, using a hierarchy of MIP solving procedures, and applying an objective function for the auxiliary MIPs that incorporates second-order information. For the latter, we introduced the so-called Hesse-distance.

In our computational experiments, the dynamic use of various MIP solving strategies showed the favorable behavior to produce more solutions and better solutions in a shorter average running time. A convex combination of the Hesse-distance function and the Manhattan distance likewise improved the number of solutions found and their quality.

**Acknowledgements** The major part of this research was conducted while Timo Berthold was at Zuse Institute Berlin and Pietro Belotti was at Clemson University. We would like to thank our former affiliations for all the support they gave us, including, but not limited to, this project.

## References

1. Achterberg, T.: Constraint integer programming. Ph.D. thesis, Technische Universität Berlin (2007)
2. Achterberg, T.: SCIP: Solving Constraint Integer Programs. *Mathematical Programming Computation* **1**(1), 1–41 (2009)
3. Achterberg, T., Berthold, T.: Improving the Feasibility Pump. *Discrete Optimization* **4**(1), 77–86 (2007)
4. Baena, D., Castro, J.: Using the analytic center in the feasibility pump. *Operations Research Letters* **39**(5), 310–317 (2011). DOI 10.1016/j.orl.2011.07.005
5. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: *Nonlinear programming: theory and algorithms*. John Wiley & Sons (2013)
6. Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods & Software* **24**, 597–634 (2009)
7. Bertacco, L., Fischetti, M., Lodi, A.: A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization* **4**(1), 63–76 (2007)
8. Berthold, T.: Primal heuristics for mixed integer programs. Diploma thesis, Technische Universität Berlin (2006)
9. Berthold, T.: Heuristic algorithms in global MINLP solvers. Ph.D. thesis, Technische Universität Berlin (2014)
10. Berthold, T.: RENS – the optimal rounding. *Mathematical Programming Computation* **6**(1), 33–54 (2014)
11. Berthold, T., Gleixner, A.M.: Undercover: a primal MINLP heuristic exploring a largest sub-MIP. *Mathematical Programming* **144**(1–2), 315–346 (2014)
12. Berthold, T., Heinz, S., Pfetsch, M.E., Vigerske, S.: Large neighborhood search beyond MIP. In: L.D. Gaspero, A. Schaerf, T. Stützle (eds.) *Proceedings of the 9th Metaheuristics International Conference (MIC 2011)*, pp. 51–60 (2011)
13. Boland, N.L., Eberhard, A.C., Engineer, F.G., Tsoukalas, A.: A new approach to the feasibility pump in mixed integer programming. *SIAM Journal on Optimization* **22**(3), 831–861 (2012)
14. Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wächter, A.: An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization* **5**, 186–204 (2008)
15. Bonami, P., Cornuéjols, G., Lodi, A., Margot, F.: A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming* **119**(2), 331–352 (2009)
16. Bonami, P., Gonçalves, J.: Heuristics for convex mixed integer nonlinear programs. *Computational Optimization and Applications* **51**, 729–747 (2012)
17. Bussieck, M.R., Drud, A.S., Meeraus, A.: MINLPlib – a collection of test models for mixed-integer nonlinear programming. *INFORMS Journal on Computing* **15**(1), 114–119 (2003)
18. CBC user guide – COIN-OR. <http://www.coin-or.org/Cbc>
19. D’Ambrosio, C., Frangioni, A., Liberti, L., Lodi, A.: Experiments with a feasibility pump approach for nonconvex MINLPs. In: P. Festa (ed.) *Experimental Algorithms, Lecture Notes in Computer Science*, vol. 6049, pp. 350–360. Springer Berlin Heidelberg (2010)
20. D’Ambrosio, C., Frangioni, A., Liberti, L., Lodi, A.: A storm of feasibility pumps for nonconvex MINLP. *Mathematical Programming* **136**, 375–402 (2012)
21. Danna, E., Rothberg, E., Pape, C.L.: Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming* **102**(1), 71–90 (2004)
22. De Santis, M., Lucidi, S., Rinaldi, F.: A new class of functions for measuring solution integrality in the feasibility pump approach. *SIAM Journal on Optimization* **23**(3), 1575–1606 (2013)
23. De Santis, M., Lucidi, S., Rinaldi, F.: Feasibility pump-like heuristics for mixed integer problems. *Discrete Applied Mathematics* **165**, 152–167 (2014)
24. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming* **36**(3), 307–339 (1986)

25. Fischetti, M., Glover, F., Lodi, A.: The feasibility pump. *Mathematical Programming* **104**(1), 91–104 (2005)
26. Fischetti, M., Lodi, A.: Local branching. *Mathematical Programming* **98**(1-3), 23–47 (2003)
27. Fischetti, M., Lodi, A.: Heuristics in mixed integer programming. In: J.J. Cochran, L.A. Cox, P. Keskinocak, J.P. Kharoufeh, J.C. Smith (eds.) *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc. (2010). Online publication
28. Fischetti, M., Salvagnin, D.: Feasibility pump 2.0. *Mathematical Programming Computation* **1**, 201–222 (2009)
29. Gleixner, A., Vigerske, S.: Analyzing the computational impact of individual MINLP solver components. Talk at MINLP 2014, Carnegie Mellon University, Pittsburgh, PA, USA (2014). <http://minlp.cheme.cmu.edu/2014/papers/gleixner.pdf>
30. Hanafi, S., Lazić, J., Mladenović, N.: Variable neighbourhood pump heuristic for 0-1 mixed integer programming feasibility. *Electronic Notes in Discrete Mathematics* **36**, 759–766 (2010). DOI 10.1016/j.endm.2010.05.096
31. Ipopt (Interior Point OPTimizer). <http://www.coin-or.org/Ipopt/>
32. Land, A.H., Doig, A.G.: An automatic method of solving discrete programming problems. *Econometrica* **28**(3), 497–520 (1960)
33. Liberti, L., Mladenović, N., Nannicini, G.: A recipe for finding good solutions to MINLPs. *Mathematical Programming Computation* **3**, 349–390 (2011)
34. Lindo Systems, Inc. <http://www.lindo.com>
35. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. *Mathematical Programming* **10**, 146–175 (1976)
36. Nannicini, G., Belotti, P.: Rounding-based heuristics for nonconvex MINLPs. *Mathematical Programming Computation* **4**(1), 1–31 (2012)
37. Nannicini, G., Belotti, P., Liberti, L.: A local branching heuristic for MINLPs. *ArXiv e-prints* (2008). URL <http://arxiv.org/abs/0812.2188>
38. Sharma, S.: Mixed-integer nonlinear programming heuristics applied to a shale gas production optimization problem. Master’s thesis, Norwegian University of Science and Technology (2013)
39. Sharma, S., Knudsen, B.R., Grimstad, B.: Towards an objective feasibility pump for convex MINLPs. *Computational Optimization and Applications* pp. 1–17 (2014)
40. Tawarmalani, M., Sahinidis, N.V.: Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications, *Nonconvex Optimization and Its Applications*, vol. 65. Kluwer Academic Publishers (2002)
41. Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming* **99**, 563–591 (2004)
42. Vigerske, S.: Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming. Ph.D. thesis, Humboldt-Universität zu Berlin (2013)
43. Wächter, A., Biegler, L.T.: On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming* **106**(1), 25–57 (2006)
44. Wunderling, R.: Paralleler und objektorientierter Simplex-Algorithmus. Ph.D. thesis, Technische Universität Berlin (1996)