EURO Journal on Computational Optimization 10 (2022) 100051

Contents lists available at ScienceDirect

EURO Journal on Computational Optimization

www.elsevier.com/locate/ejco

Feasible rounding based diving strategies in branch-and-bound methods for mixed-integer optimization

Christoph Neumann [a], Stefan Schwarze [a,*], Oliver Stein [a], Benjamin Müller [b]

[a] *Institute of Operations Research, Karlsruhe Institute of Technology (KIT), Germany*
[b] *Zuse Institute Berlin (ZIB), Germany*

A R T I C L E   I N F O

Keywords:
Variable fixing
Diving
Granularity
MILP
MINLP

A B S T R A C T

In this paper, we study the behavior of feasible rounding approaches for mixed-integer optimization problems when integrated into branch-and-bound methods. Our research addresses two important aspects. First, we develop insights into how an (enlarged) inner parallel set, which is the main component for feasible rounding approaches, behaves when we move down a search tree. Our theoretical results show that the number of feasible points obtainable from the inner parallel set is nondecreasing with increasing depth of the search tree. Thus, they hint at the potential benefit of integrating feasible rounding approaches into branch-and-bound methods. Second, based on those insights, we develop a novel primal heuristic for MILPs that fixes variables in a way that promotes large inner parallel sets of child nodes.

Our computational study shows that combining feasible rounding approaches with the presented diving ideas yields a significant improvement over their application in the root node. Moreover, the proposed method is able to deliver best solutions for the MIP solver SCIP for a significant share

* Corresponding author.
E-mail address: stefan.schwarze@kit.edu (S. Schwarze).

https://doi.org/10.1016/j.ejco.2022.100051

of problems which hints at its potential to support solving MILPs.

1. Introduction

In this paper, we study the task of computing good feasible points for mixed-integer optimization problems. While our theoretical study covers general mixed-integer nonlinear optimization problems (MINLPs), the main focus of this paper will be on mixed-integer linear optimization problems (MILPs). Our work is based upon feasible rounding ideas from [16,17,19], which aim at quickly computing such points by relaxing the difficulties imposed by integrality constraints. To do so, they make use of a so-called *inner parallel set* of the continuously relaxed feasible set (cf. Section 2) for which any rounding of any of its elements is feasible for the original problem. This inner parallel set can be explicitly computed in the linear case and approximated in different ways in the nonlinear case. If it is nonempty, then the problem is called *granular*, and one can minimize the original objective function over it and round any of its optimal points to a point which is feasible for the original problem.

The task of computing a feasible point for mixed-integer optimization problems is known to be an NP-hard problem, even if all constraint functions are linear [20]. This has triggered the development of many primal heuristics, among them the feasibility pump [1,7,8], undercover [4], relaxation enforced neighborhood search [2], diving strategies [5], and many others (see [3] for a comprehensive overview).

What distinguishes feasible rounding approaches from the above methods is the underlying geometric notion of granularity, which allows us to better understand the circumstances needed for their successful applicability. This concept not only allows us to state conditions when the approaches can be used for the computation of feasible points, but it also enables the derivation of a priori error bounds for the objective value which indicate when they work well [16,18].

Feasible rounding approaches were successfully tested as standalone concepts for mixed-integer linear optimization problems [17], mixed-integer nonlinear convex optimization problems [16], and mixed-integer convex and nonconvex quadratically constrained quadratic optimization problems [19]. Granularity is often observed and easily exploited in problems without equality constraints on integer variables, hence the above mentioned as well as the present paper focus on such problems. In [15] it is demonstrated that granularity is also possible, but needs more effort and is less likely to occur, under the presence of equality constraints on integer variables.

So far it is untested how these approaches work when integrated in branch-and-bound methods. In particular, it has not been studied how exploring the search tree affects the inner parallel set. In this paper, we intend to close this gap. Additionally, based on these results, we develop a novel method that is specifically tailored to inner parallel sets and feasible rounding approaches. The paper is structured as follows.

In Section 2 we briefly introduce the basic concepts of an inner parallel set and of granularity. We then provide a theoretical analysis of the behavior of inner parallel sets when variables are fixed in Section 3. Thus, we investigate the theoretical potential of integrating feasible rounding approaches into branch-and-bound methods. Moreover, the results from this section give rise to a new primal heuristic which can improve upon standalone feasible rounding approaches. This is the content of Section 4. To arrive at a specific algorithm, we formulate a method for MILPs. Finally, in Section 5, we conduct a computational study on the MIPLIB 2017 [10] that sheds a light on the effectiveness of these diving strategies and also on the potential benefit of integrating feasible rounding approaches into the solver SCIP [9]. Section 6 concludes the article and offers directions for further research.

2. Preliminaries

We study mixed-integer nonlinear optimization problems of the form

$$MINLP: \quad \min_{(x,y)\in\mathbb{R}^n\times\mathbb{Z}^m} c^\top x+d^\top y \quad \text{s.t.} \quad g_i(x,y)\leqslant 0,\ i\in I,\ Ax+By\leqslant b,\ y^\ell\leqslant y\leqslant y^u,$$

with real-valued functions g_i, $i \in I$, defined on $\mathbb{R}^n \times \mathbb{R}^m$, a finite index set $I = \{1, \dots, q\}$, $q \in \mathbb{N}$, a (p, n)-matrix A and a (p, m)-matrix B, $p \in \mathbb{N}$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}^m$, $b \in \mathbb{R}^p$ and box constraints with $y^\ell, y^u \in \mathbb{Z}^m$, $y^\ell \leqslant y^u$. Moreover, with $\widehat{M}$ we denote the feasible set of the continuous relaxation *NLP* of *MINLP*, that is,

$$\widehat{M} = \{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m |\ g_i(x,y)\leqslant 0,\ i\in I,\ Ax+By\leqslant b,\ y^\ell\leqslant y\leqslant y^u\},$$

so that we may write the feasible set of *MINLP* as $M = \widehat{M} \cap (\mathbb{R}^n \times \mathbb{Z}^m)$.

In this section, we introduce inner parallel sets in a general (geometrical) context, which will be the foundation for the rest of this article. We briefly discuss how this concept can be used computationally in the special case of MILPs (i.e., $I = \varnothing$) for which we will develop a novel diving heuristic in Section 4.

2.1. Geometrical idea

Crucial for all feasible rounding approaches is the construction of the inner parallel set

$$\widehat{M}^- \ := \ \{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m |\ \{x\}\times B_\infty\left(y,\tfrac{1}{2}\right)\subseteq \widehat{M}\} \tag{1}$$

of the relaxed feasible set $\widehat{M}$, with the box

$$B_\infty\left(y, \tfrac{1}{2}\right) := \{\eta \in \mathbb{R}^m |\ \|\eta - y\|_\infty \leqslant \tfrac{1}{2}\}.$$

The decisive characteristic of this set is that it ensures the feasibility for *MINLP* of roundings of its elements. To be more specific, we call $(\breve{x}, \breve{y})$ rounding of a point $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$, if

$$\breve{x} = x,\ \breve{y} \in \mathbb{Z}^m,\ |\breve{y}_j - y_j| \leqslant \tfrac{1}{2},\ j = 1, \dots, m \tag{2}$$

hold, i.e., each component of y is rounded to a closest point in the integer grid and x remains unchanged. Then for any $(x, y) \in \widehat{M}^-$ we have $(\breve{x}, \breve{y}) \in M$, see [17]. This gives rise to the following definition of granularity.

Definition 2.1 *([17]). The feasible set M is called granular if its inner parallel set $\widehat{M}^-$ is nonempty. A problem MINLP is called granular if its feasible set M is granular.*

Notice that granularity is sufficient but not necessary for the consistency of *MINLP* and that it depends on the problem formulation due to its dependency on $\widehat{M}$. In particular, less tight formulations are beneficial for the consistency of $\widehat{M}^-$ and thus for the applicability of the granularity concept.

Such enlargements may be viewed as a preprocessing step that transforms $\widehat{M}$ to $\widetilde{M}$, with

$$M = \widetilde{M} \cap (\mathbb{R}^n \times \mathbb{Z}^m) \text{ and } \widetilde{M} \supseteq \widehat{M}, \tag{3}$$

where enlargement ideas range from small perturbations to the construction of structurally different formulations [19]. To use the granularity concept, one can then work with the *enlarged inner parallel set* $\widetilde{M}^-$, where the transition from $\widetilde{M}$ to $\widetilde{M}^-$ is defined as in (1). In Section 2.2, we provide an example of this enlargement procedure for MILPs. For more details and motivating examples we refer to [17].

For a set $S \subseteq \mathbb{R}^n \times \mathbb{R}^m$ we define the set of roundings obtainable from S as

$$R(S) := \{(\breve{x}, \breve{y}) \in \mathbb{R}^n \times \mathbb{Z}^m |\ (x, y) \in S \text{ and } (2)\}$$

and abbreviate $R := R(\widehat{M}^-)$.

Fig. 1 illustrates the construction of the inner parallel set $\widehat{M}^-$ for a two dimensional purely integer example. The set M consists of four feasible points, but only the filled points are obtainable as roundings from $\widehat{M}^-$, i.e., $R = \{(1, 0)^\top, (2, 0)^\top\}$.

2.2. *Construction of inner parallel sets for MILPs*

Next, we elaborate the algorithmic construction of inner parallel sets for the special case $I = \varnothing$, where *MINLP* collapses to an *MILP*. In this case, the relaxed feasible set reads as

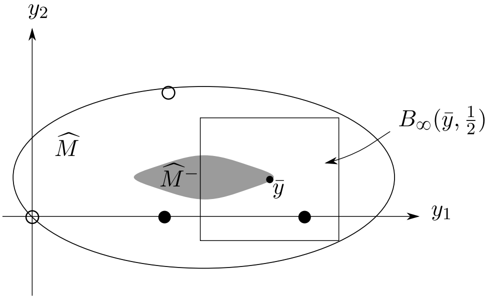

Fig. 1. Construction of the inner parallel set $\widehat{M}^-$. The filled points are obtainable as roundings from $\widehat{M}^-$ and thus form the set R.

$$\widehat{M} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m |\ Ax + By \leqslant b,\ y^\ell \leqslant y \leqslant y^u\},$$

and we can use the results from [17] to obtain a functional description of the enlarged inner parallel set as follows.

Let $\alpha_i^\top$ and $\beta_i^\top$ denote the rows of A and B, respectively. Moreover, let $\omega_i \in \mathbb{N}_0$ stand for the greatest common divisor of the entries of β_i, if $\beta_i \in \mathbb{Z}^m$ and $\alpha_i = 0$ hold, and be zero, otherwise. For a real number a, with $\lfloor a \rfloor_{\omega_i}$ we denote the floor function with respect to ω_i, that is

$$\lfloor a \rfloor_{\omega_i} = \begin{cases} \max\{z \in \omega_i\mathbb{Z} |\ z \leqslant a\}, & \omega_i \neq 0 \\ a, & \text{otherwise.} \end{cases}$$

Moreover, for $a \in \mathbb{R}^p$ and $\omega \in \mathbb{N}_0^p$, let

$$\lfloor a \rfloor_\omega := (\lfloor a_1 \rfloor_{\omega_1}, \dots, \lfloor a_p \rfloor_{\omega_p})^\top.$$

Then, with an enlargement parameter $\delta \in [0, 1)$, $\widetilde{b} := \lfloor b \rfloor_\omega + \delta\omega$ and the all-ones vector e of dimension m, an explicitly computable enlarged relaxed feasible set is

$$\widetilde{M} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m |\ Ax + By \leqslant \widetilde{b},\ y^\ell - \delta e \leqslant y \leqslant y^u + \delta e\}. \quad (4)$$

For ease of notation, we omit the δ-dependency of $\widetilde{M}$ and the reader may just think of δ as being some fixed value close to (but smaller than) one. The geometric idea behind this construction is to loosen the description of M, because the inner parallel set of a loosened description is more likely to be nonempty than the original description. For $\delta = 1$ the loosening would be so coarse that infeasible points became feasible. Instead, for $\delta < 1$ the presented enlargement guarantees $\widetilde{M} \cap (\mathbb{R}^n \times \mathbb{Z}^m) = M$ and, for sufficiently large values of δ, $\widetilde{M} \supseteq \widehat{M}$ [17].

Moreover, with $\|\beta\|_1 := (\|\beta_1\|_1, \dots, \|\beta_p\|_1)^\top$ the enlarged inner parallel set is

$$\widetilde{M}^- = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m |\ Ax + By \leqslant \widetilde{b} - \tfrac{1}{2}\|\beta\|_1,\ y^\ell + (\tfrac{1}{2} - \delta)e \leqslant y \leqslant y^u - (\tfrac{1}{2} - \delta)e\}, \quad (5)$$

where, again, we refer the reader to [17] for the derivation. We next illustrate the computation of the enlarged inner parallel set for a binary knapsack example which we shall also revisit in Section 3 to demonstrate the usefulness of fixing binary variables. Here, we use the abbreviation $\widetilde{R} := R(\widetilde{M}^-)$.

Example 2.2. Let us consider the (binary knapsack) feasible set

$$M = \{y \in \mathbb{B}^3 |\ 1 \leqslant \sum_{i=1}^{3} y_i \leqslant 2\}.$$

Using (5) with $\omega = (1,1)^\top$ and $\|\beta\|_1 = (3,3)^\top$ we can compute the enlarged inner parallel set

$$\widetilde{M}^- = \{y \in \mathbb{R}^3 |\ \tfrac{5}{2} - \delta \leqslant \textstyle\sum_{i=1}^3 y_i \leqslant \tfrac{1}{2} + \delta,\ (\tfrac{1}{2} - \delta)e \leqslant y \leqslant (\tfrac{1}{2} + \delta)e\},$$

which is empty for any $\delta \in [0,1)$. This also implies $\widetilde{R} = \varnothing$ for this example.

3. Fixing variables and inner parallel sets - a geometrical perspective

In this section, we present a geometrical perspective on the effects that occur when we move down a search tree. We investigate the implications of *fixing* integer variables to values $\ell \in \mathbb{Z}$. This covers the important case of branching on a binary variable and is often also feasible for an integer variable i when the difference of the bounds $y_i^u - y_i^\ell$ is small enough. Feasible rounding approaches work especially well for problems with a relatively small number of binary variables compared to general integer variables, which was noted in the computational study in [17] and further substantiated by the theoretical bounds derived in [18]. Therefore, the case of fixing binary variables is of particular interest for the present article.

In the following, we make no further distinction between different nodes of a branch-and-bound tree, but demonstrate the effects only for the root node of *MINLP*. We stress that this is for notational convenience only and that our results are applicable to fixing variables in any branch-and-bound node.

As we shall presently demonstrate, fixing integer variables increases the chances for finding good feasible points using feasible rounding approaches. To be more specific, fixing an integer variable i to a value $\ell \in \mathbb{Z} \cap [y_i^\ell, y_i^u]$ results in the i-ℓ-fixed relaxed feasible set

$$\widehat{M}_{(i)}(\ell) = \{(x, \widetilde{y}) \in \mathbb{R}^n \times \mathbb{R}^{m-1} |\ (x, (\widetilde{y}_1, \dots, \widetilde{y}_{i-1}, \ell, \widetilde{y}_i, \dots, \widetilde{y}_{m-1})) \in \widehat{M}\}. \tag{6}$$

Moreover, with

$$\widehat{M}_{(i)}(\ell)^- = \{(x, \widetilde{y}) \in \mathbb{R}^n \times \mathbb{R}^{m-1} |\ \{x\} \times B_\infty(\widetilde{y}, \tfrac{1}{2}) \subseteq \widehat{M}_{(i)}(\ell)\} \tag{7}$$

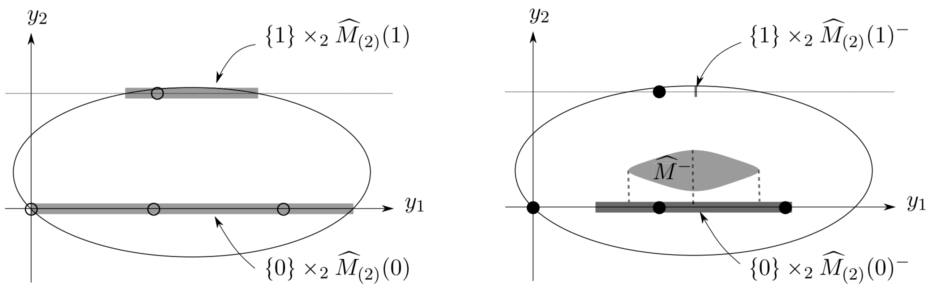

Fig. 2. Construction of the i-ℓ-relaxed feasible set (left) and the i-ℓ-fixed inner parallel sets (right) with $i = 2$ and $\ell \in \{0, 1\}$.

we denote the i-ℓ-fixed inner parallel set. We abbreviate the set of roundings obtainable from this set as $R_{(i)}(\ell) := R(\widehat{M}_{(i)}(\ell)^-)$.

Remark 3.1. The analysis in this section makes a connection between inner parallel sets and i-ℓ-fixed inner parallel sets that is independent of an enlargement step. Hence, while we make this connection only explicit for the sets $\widehat{M}^-$ and $\widehat{M}_{(i)}(\ell)^-$, all results will be equally valid for the connection of enlarged inner parallel sets $\widetilde{M}^-$ and their enlarged i-ℓ-fixed inner parallel sets $\widetilde{M}_{(i)}(\ell)^-$.

The following notation facilitates a comparison of inner parallel sets with i-ℓ-fixed inner parallel sets and thus the investigation of the effects of fixing integer variables. For $y \in \mathbb{R}^m$ and $i \in \{1, \ldots, m\}$ let

$$y^{-i} := (y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_m)^\top \in \mathbb{R}^{m-1},$$

and, correspondingly, for $y \in \mathbb{R}^{m-1}$ and some $\ell \in \mathbb{R}$, let

$$y^{+i}(\ell) := (y_1, \ldots, y_{i-1}, \ell, y_i, \ldots, y_{m-1})^\top \in \mathbb{R}^m$$

denote the vectors where we remove or insert an element at position i, respectively. Moreover, for $S^1 \subseteq \mathbb{R}$ and $S^2 \subseteq \mathbb{R}^{m-1}$, let

$$S^1 \times_i S^2 := \{s^{+i}(s^1) \in \mathbb{R}^m |\ s^1 \in S^1, s \in S^2\}.$$

The construction of i-ℓ-fixed (inner parallel) sets is illustrated in Fig. 2 for $i = 2$ and $\ell \in \{0, 1\}$. Remarkably, fixing y_2 results in

$$\{0\} \times_2 R_{(2)}(0) = \{(0,0)^\top, (1,0)^\top, (2,0)^\top\},\ \{1\} \times_2 R_{(2)}(1) = \{(1,1)^\top\}$$

and thus allows us to obtain all points in M as roundings from i-ℓ-fixed inner parallel sets. Recall from Fig. 1 that we were only able to obtain the two points $(1,0)^\top, (2,0)^\top$ as roundings from the inner parallel set $\widehat{M}^-$. Hence, this example shows that the number of roundings obtainable with feasible rounding approaches can increase when we move down a search tree. We will presently show that there is a crucial theoretical link between roundings from inner parallel sets and roundings from i-ℓ-fixed inner parallel sets which offers an explanation for this observation.

In fact, this link is already depicted on the right-hand side of Fig. 2: for any point $y \in \widehat{M}^-$, we have a "corresponding point" $y^{-2} \in (\widehat{M}_{(2)}(0))^-$, which is illustrated by the dashed lines from $\widehat{M}^-$ to $(\widehat{M}_{(2)}(0))^-$. The next lemma proves that this is not a coincidence, but that for any point from the inner parallel set, we always have a corresponding point in the i-ℓ-fixed inner parallel set if we choose ℓ to be the rounding of component i of y.

Lemma 3.2. *For any $(x,y) \in \widehat{M}^-$ and any $i \in \{1,\dots,m\}$, we have $(x, y^{-i}) \in \widehat{M}_{(i)}(\breve{y}_i)^-$.*

Proof. Let $(x,y) \in \widehat{M}^-$. Then by definition of $\widehat{M}^-$ we have

$$\{x\} \times B_\infty(y, \tfrac{1}{2}) = \{x\} \times [y_i - \tfrac{1}{2}, y_i + \tfrac{1}{2}] \times_i B_\infty(y^{-i}, \tfrac{1}{2}) \subseteq \widehat{M}.$$

With $\breve{y}_i \in [y_i - \frac{1}{2}, y_i + \frac{1}{2}]$, this implies

$$\{x\} \times \{\breve{y}_i\} \times_i B_\infty(y^{-i}, \tfrac{1}{2}) \subseteq \widehat{M} \cap (\mathbb{R}^n \times \{y \in \mathbb{R}^m |\ y_i = \breve{y}_i\}) = \{\breve{y}_i\} \times_{n+i} \widehat{M}_{(i)}(\breve{y}_i),$$

and dropping $\{\breve{y}_i\}$ in the cross product yields

$$\{x\} \times B_\infty(y^{-i}, \tfrac{1}{2}) \subseteq \widehat{M}_{(i)}(\breve{y}_i),$$

which shows the assertion. □

From a geometric point of view the statement of Lemma 3.2 may be interpreted as follows. While, for any $i \in \{1,\dots,m\}$ and any $\ell \in \mathbb{Z} \cap [y_i^\ell, y_i^u]$, any point $(x,y) \in \widehat{M}^-$ possesses the orthogonal projection $(x, (y^{-i})^{+i}(\ell))$ to the set $\mathbb{R}^n \times \{\ell\} \times_i \mathbb{R}^{m-1}$, one cannot expect this projection to be related to $\widehat{M}^-$ in the sense that (x, y^{-i}) lies in $\widehat{M}_{(i)}(\ell)^-$. In contrast to the general case, the lemma guarantees this relation for the special choice $\ell := \breve{y}_i$.

The next theorem uses this connection to show that the number of roundings obtainable from the inner parallel set is nondecreasing with increasing depth of the search tree.

Theorem 3.3. *For any $i = 1,\dots,m$, we have $R \subseteq \bigcup_{\ell \in \mathbb{Z} \cap [y_i^\ell, y_i^u]} \left(\{\ell\} \times_{n+i} R_{(i)}(\ell)\right)$.*

Proof. Let $(\breve{x}, \breve{y}) \in R$. For a corresponding point $(x,y) \in \widehat{M}^-$, Lemma 3.2 implies $(x, y^{-i}) \in \widehat{M}_{(i)}(\breve{y}_i)^-$. Note that, although the rounding $(\breve{x}, \overbrace{(y^{-i})})$ of (x, y^{-i}) is in general not unique, it can be chosen such that $\overbrace{(y^{-i})} = (\breve{y})^{-i}$ holds.

This shows $(\breve{x}, (\breve{y})^{-i}) \in R_{(i)}(\breve{y}_i)$ and, with $\ell := \breve{y}_i \in (\mathbb{Z} \cap [y_i^\ell, y_i^u])$, implies

$$(\breve{x}, \breve{y}) \in \{\ell\} \times_{n+i} R_{(i)}(\ell),$$

which proves the assertion. □

In summary, Theorem 3.3 together with our considerations from Figs. 1 and 2 immediately yields the following corollary.

Corollary 3.4. *The set of feasible points obtainable by feasible rounding approaches is nondecreasing and potentially increases with increasing depth of the search tree.*

Let us next revisit Example 2.2 to illustrate the explicit construction of i-ℓ-fixed enlarged inner parallel sets for MILPs.

Example 3.5. Let us consider the feasible set M from Example 2.2 and fix y_3. Again, with $\omega = (1,1)^\top$, this results in the two 3-ℓ-fixed enlarged sets

$$\widetilde{M}_{(3)}(0) = \{\widetilde{y} \in \mathbb{R}^2 |\ 1 - \delta \leqslant \widetilde{y}_1 + \widetilde{y}_2 \leqslant 2 + \delta,\ -\delta e \leqslant \widetilde{y} \leqslant (1+\delta)e\},$$
$$\widetilde{M}_{(3)}(1) = \{\widetilde{y} \in \mathbb{R}^2 |\ -\delta \leqslant \widetilde{y}_1 + \widetilde{y}_2 \leqslant 1 + \delta,\ -\delta e \leqslant \widetilde{y} \leqslant (1+\delta)e\},$$

and, with $\|\beta\|_1 = (2,2)^\top$, yields the enlarged inner parallel sets

$$\widetilde{M}_{(3)}(0)^- = \{\widetilde{y} \in \mathbb{R}^2 |\ 2 - \delta \leqslant \widetilde{y}_1 + \widetilde{y}_2 \leqslant 1 + \delta,\ (\tfrac{1}{2} - \delta)e \leqslant \widetilde{y} \leqslant (\tfrac{1}{2}+\delta)e\},$$
$$\widetilde{M}_{(3)}(1)^- = \{\widetilde{y} \in \mathbb{R}^2 |\ 1 - \delta \leqslant \widetilde{y}_1 + \widetilde{y}_2 \leqslant \delta,\ (\tfrac{1}{2} - \delta)e \leqslant \widetilde{y} \leqslant (\tfrac{1}{2}+\delta)e\}.$$

The crucial difference compared to the (unfixed) enlarged inner parallel set is that we no longer have to account for possible rounding errors of y_3 which results in the fact that each value of $\|\beta\|_1$ can be reduced from 3 to 2. Thus, while the enlarged inner parallel set of the original feasible set is empty for any $\delta \in [0,1)$, both i-3-fixed enlarged inner parallel sets are nonempty for $\delta \in [\frac{1}{2}, 1)$.

With $\widetilde{R}_{(i)}(\ell) := R(\widetilde{M}_{(i)}(\ell)^-)$, we even have $(\{0\} \times_3 \widetilde{R}_{(3)}(0)) \cup (\{1\} \times_3 \widetilde{R}_{(3)}(1)) = M$ for $\delta \in [\frac{1}{2}, 1)$, that is, all feasible points may be obtained as roundings from these 3-ℓ-fixed inner parallel sets.

Hence, Example 3.5 not only offers a computational perspective on the construction of i-ℓ-fixed inner parallel sets, but also further substantiates the potential of fixing integer variables for feasible rounding approaches.

Let us conclude this section with some considerations on the enlargement step. In Remark 3.1 we highlighted that the transition from $\widetilde{M}^-$ to $\widetilde{M}^-_{(i)}(\ell)$ is analogous to that from $\widehat{M}^-$ to $\widehat{M}^-_{(i)}(\ell)$ and that all results derived in this section are hence equally valid for this transition. Yet, there is an additional potential that can be harvested: there can be the possibility to enlarge the set $\widetilde{M}_{(i)}(\ell)$ even further, once variable i is fixed to ℓ. As an example, consider a constraint $\beta_i^\top y \leqslant b_i$ with $\beta_i = (1,3,3)^\top$ and $b_i = 3$. Then, when fixing y_1 and using the enlargement techniques introduced in Section 2.2, the entry ω_i can be increased from 1 to 3 in the transition from the set $\widetilde{M}$ to $\widetilde{M}_{(i)}(\ell)$. We will exploit this fact in our development of a diving method for MILPs in the following section.

4. A diving heuristic for MILPs

In this section, we elaborate some algorithmic ideas on how the results from the previous section can be used for the development of a diving heuristic. We formulate an explicit method for mixed-integer linear optimization problems, i.e., $I = \varnothing$, and use the same notation as in Section 2. In particular, we employ the construction of the enlarged inner parallel set from Section 2.2.

Some important considerations of Section 4.2 (the so-called *degree of freedom*) explicitly need linearity of the constraint functions, which is one of the main reasons we formulate the method for MILPs. We stress, however, that many results of this section generalize directly to mixed-integer nonlinear optimization problems (Section 4.1 as well as Proposition 4.3) and thus may serve as a foundation for the development of a diving method for MINLPs as well. Moreover, many MINLP solvers use LP relaxations in the nodes to which our results are also applicable.

We initially elaborate diving approaches for the cases of a nonempty and an empty inner parallel set separately, and subsequently bring them together into a general framework. In the first case, we show how to ensure that inner parallel sets of resulting child nodes remain nonempty. Our aim is to find a feasible point with improved objective value. For empty inner parallel sets we show how certain auxiliary optimization problems and ways of fixing variables are likely to generate nonempty inner parallel sets of child nodes.

4.1. A diving step for a nonempty enlarged inner parallel set

Let us initially elaborate a method for $\widetilde{M}^- \neq \varnothing$. Minimizing the objective function of *MILP* over the enlarged inner parallel set yields the *objective-based problem*

$$P^{ob}: \quad \min_{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m} c^\top x + d^\top y \quad \text{s.t.} \quad (x,y) \in \widetilde{M}^-.$$

Due to our assumption $\widetilde{M}^- \neq \varnothing$, the problem P^{ob} is either solvable or unbounded, where unboundedness of P^{ob} would imply unboundedness of *MILP*. As we develop a method that generates good feasible points, the latter case is not interesting in our context. In this section, we therefore assume that *MILP* is bounded. This, together with $\widetilde{M}^- \neq \varnothing$, guarantees the existence of an optimal point (x^{ob}, y^{ob}) of P^{ob}. We denote any rounding of (x^{ob}, y^{ob}) by $(\breve{x}^{ob}, \breve{y}^{ob})$ and the objective value of the rounded optimal point by $\breve{v}^{ob} = c^\top \breve{x}^{ob} + d^\top \breve{y}^{ob}$.

One crucial observation from Lemma 3.2 is that if the enlarged inner parallel set of some branch-and-bound node is nonempty and (x, y) is any of its feasible points, we immediately obtain m nonempty i-ℓ-fixed (child node) enlarged inner parallel sets, where $i \in \{1, \dots, m\}$ and $\ell = \breve{y}_i$.

Then, as a diving step, we may solve a corresponding i-ℓ-fixed objective-based problem

$$P^{ob}_{(i)}(\ell): \quad \min_{(x,\widetilde{y}) \in \mathbb{R}^n \times \mathbb{R}^{m-1}} c^\top x + (d^{-i})^\top \widetilde{y} + d_i \ell \quad \text{s.t.} \quad (x, \widetilde{y}) \in \widetilde{M}_{(i)}(\ell)^-,$$

denote any of its optimal points by $(x^{ob}, \widetilde{y}^{ob})$ and its optimal value by $v^{ob}_{(i)}(\ell)$. Due to the previous considerations on the possibility of an additional enlargement step of the i-ℓ-fixed inner parallel set, we suggest to fix variable i to ℓ *before* determining the vector ω in the computation of $\widetilde{M}_{(i)}(\ell)^-$ in accordance with (5).

Moreover, we abbreviate

$$(x^{ob}, \widetilde{y}^{ob})_{(i)}(\ell) := (x^{ob}, (\widetilde{y}^{ob})^{+i}(\ell)) \tag{8}$$

so that we can analogously denote the (rounded) *MILP*-feasible point obtained by solving the i-ℓ-fixed objective-based problem, rounding all y components and "re-inserting" value ℓ at position i with $(\breve{x}^{ob}, \breve{y}^{ob})_{(i)}(\ell)$. The objective value of $(\breve{x}^{ob}, \breve{y}^{ob})_{(i)}(\ell)$ is denoted by $\breve{v}^{ob}_{(i)}(\ell)$.

While this applies to roundings of *any* feasible point from $\widetilde{M}^-$, one fruitful idea is to (iteratively) use roundings of optimal points of (i-ℓ-fixed) objective-based problems, that is, to set $\ell = \breve{y}^{ob}_i$. The next example elaborates this idea in more detail and shows that, even though the fixing value for variable i is given by $\breve{y}^{ob}_i$, different orders of selecting variables can yield different feasible points.

Example 4.1. Consider the optimization problem

$$IP: \quad \min_{y \in \mathbb{Z}^3} -y_1 - 3y_3 \quad \text{s.t.} \quad y_1 + y_2 + 2y_3 \leqslant 3, \ -2y_1 - 2y_2 + y_3 \leqslant -1, \ 0 \leqslant y \leqslant 2e.$$

By using Equation (5) with $\delta = 0.9$, we can formulate the objective-based problem

$$P^{ob}: \quad \min_{y \in \mathbb{R}^3} -y_1 - 3y_3 \quad \text{s.t.} \quad y_1 + y_2 + 2y_3 \leqslant 1.9, \ -2y_1 - 2y_2 + y_3 \leqslant -2.6,$$
$$-0.4e \leqslant y \leqslant 2.4e$$

and compute its optimal point $y^{ob} = (1.82, -0.4, 0.24)^\top$. Rounding y^{ob} yields the IP-feasible point $\breve{y}^{ob} = (2, 0, 0)^\top$ with objective value $\breve{v}^{ob} = -2$.

Fixing $y_2 = 0$ and setting $\widetilde{y} := (y_1, y_3)$ yields the 2-0-fixed objective-based problem

$$P^{ob}_{(2)}(0): \quad \min_{\widetilde{y} \in \mathbb{R}^2} -y_1 - 3y_3 \quad \text{s.t.} \quad y_1 + 2y_3 \leqslant 2.4, \ -2y_1 + y_3 \leqslant -1.6, \ -0.4e \leqslant \widetilde{y} \leqslant 2.4e$$

with optimal point $(y_1, y_3)^{ob} = (1.12, 0.64)$ and thus the IP-feasible point $\breve{y}^{ob}_{(2)}(0) = (1,0,1)^\top$ with improved objective value $\breve{v}^{ob}_{(2)}(0) = -4$.

After solving the problem P^{ob}, we also had the options to fix $y_1 = 2$ or $y_3 = 0$. Both fixings, however, rule out the possibility to obtain the feasible point $(1,0,1)^\top$ on a path in the search tree and this point is hence only obtainable if we initially fix $y_2 = 0$.

Example 4.1 shows that fixing components of rounded optimal points from P^{ob} has the potential to yield improved points and that the choice of variables actually matters. When fixing one component, new options for other components become available - and thus new feasible points. In a diving heuristic, this allows the flexibility to select a component and thus to choose the order of fixing while ensuring nonempty inner parallel sets of child nodes. We will make some remarks on possible strategies for fixing variables in Section 4.3.

Remark 4.2. The main reason for our choice of fixing variable i to $\breve{y}^{ob}_i$ was that it guaranteed granularity of child nodes and that this particular choice is promising with regard to the objective value. Yet, to have more flexibility may be fertile for developing further diving ideas and may offer possibilities to obtain better feasible roundings. In this regard, note that if we have two points $(x^1, y^1), (x^2, y^2) \in \widetilde{M}^-$, again by Lemma 3.2 we can fix any i of these k variables to values $\ell \in \{\breve{y}^1_i, \breve{y}^2_i\}$. In our linear setting, the inner parallel set is convex and hence even all values from the interval $[\min\{\breve{y}^1_i, \breve{y}^2_i\}, \max\{\breve{y}^1_i, \breve{y}^2_i\}]$ are possible.

As a next step, we consider diving possibilities, similar to those developed so far, for an empty enlarged inner parallel set.

4.2. A diving step for an empty enlarged inner parallel set

In this section, we develop a diving method for non-granular nodes. To gather information about the "degree of non-granularity" and about the impact of fixing variables, it will turn out to be beneficial to investigate the (solvable) *feasibility problem*

$$P^f: \quad \min_{(x,y,z)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}} z \quad \text{s.t.} \quad (x,y,z) \in \widetilde{M}^-_L,$$

where the feasible set of P^f is the *lifted* enlarged inner parallel set of $\widehat{M}$,

$$\widetilde{M}^-_L = \{(x,y,z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} | \\ Ax + By - ze \leqslant \widetilde{b} - \tfrac{1}{2}\|\beta\|_1, \; y^\ell + (\tfrac{1}{2} - \delta)e \leqslant y \leqslant y^u - (\tfrac{1}{2} - \delta)e, \; z \geqslant -1\}.$$

Note that the introduced enlargement techniques work only for constraints where continuous variables are absent. Therefore, it is crucial to lift the problem *after* the application of an enlargement step, that is, after the computation of ω.

We denote an optimal point of P^f by (x^f, y^f, z^f) and its optimal value by v^f. As already mentioned in [17], granularity is equivalent to $v^f \leqslant 0$ which implies $(x^f, y^f) \in \widetilde{M}^-$ and thus $(\breve{x}^f, \breve{y}^f) \in M$. Moreover, we may obtain an *MILP*-feasible point even in the case of a non-granular problem where $v^f > 0$ holds. Hence, the "reverse implication" $(v^f > 0) \Rightarrow (\breve{x}^f, \breve{y}^f) \notin M$ is not true. Of course, this possibility to generate *non-granular feasible roundings* can be used algorithmically to find feasible points for more problems from practice.

We next establish a crucial property of diving methods which fix y-components to roundings $\breve{y}^f$ of y^f: this way of fixing entails that the optimal value v^f of the auxiliary problem P^f cannot deteriorate. To state this formally, analogously to the i-ℓ-fixed objective-based problem, with

$$P^f_{(i)}(\ell): \quad \min_{(x,\widetilde{y},z)\in\mathbb{R}^n\times\mathbb{R}^{m-1}\times\mathbb{R}} z \quad \text{s.t.} \quad (x,\widetilde{y},z) \in (\widetilde{M}_L)_{(i)}(\ell)^-$$

we denote the i-ℓ-fixed feasibility problem, with $(x^f, \widetilde{y}^f, z^f)$ any of its optimal points and with $v^f_{(i)}(\ell)$ its optimal value.

Proposition 4.3. *Let (x^f, y^f, z^f) be an optimal point of P^f. Then for any $i \in \{1, \ldots, m\}$ the following assertions are true:*

(a) $(x^f, (y^f)^{-i}, z^f)$ is feasible for $P^f_{(i)}(\breve{y}_i)$.
(b) the inequality $v^f_{(i)}(\breve{y}^f_i) \leqslant v^f$ is valid.

Proof. Part (a) is an immediate consequence of Lemma 3.2. It implies $v^f_{(i)}(\breve{y}_i) \leqslant z^f = v^f$ and thus part (b) of the assertion. □

Proposition 4.3b establishes a firm basis for a diving step in the sense that it offers possibilities to fix variables which guarantee that the degree of non-granularity cannot deteriorate. Of course, we are interested in actually improving upon the value $v^f > 0$, which is not ruled out, but also not immediately implied by Proposition 4.3. Therefore, we next derive conditions under which actual progress towards feasibility in the i-ℓ-fixed feasibility problem (i.e. $v_{(i)}(\breve{y}^f_i) < v^f$) is *guaranteed*. This will also help us to determine components of y whose fixings might be fruitful.

In this regard, let us examine a constraint j from $\widetilde{M}^-_L$ evaluated at (x^f, y^f, z^f),

$$\alpha_j^\top x^f + \beta_j^\top y^f - z^f \leqslant \widetilde{b}_j - \tfrac{1}{2}\|\beta_j\|_1 .$$

With B_{ji} denoting the entry located at row j and column i of B, using the relations

$$\|\beta_j\|_1 = \left\|\beta_j^{-i}\right\|_1 + |B_{ji}|, \text{ and } \beta_j^\top y^f = (\beta_j^{-i})^\top (y^f)^{-i} + B_{ji} y^f_i,$$

this constraint can be written as

$$\alpha_j^\top x^f + (\beta_j^{-i})^\top (y^f)^{-i} - z^f \leqslant \widetilde{b}_j - \tfrac{1}{2}(\|\beta_j^{-i}\|_1 + |B_{ji}|) - B_{ji} y_i^f. \tag{9}$$

Proposition 4.3a implies that evaluating the corresponding constraint j of $(\widetilde{M}_L)_{(i)}(\breve{y}_i^f)^-$ at $(x^f, (y^f)^{-i}, z^f)$ yields the valid inequality

$$\alpha_j^\top x^f + (\beta_j^{-i})^\top (y^f)^{-i} - z^f \leqslant \widetilde{b}_j - \tfrac{1}{2}\|\beta_j^{-i}\|_1 - B_{ji} \breve{y}_i^f. \tag{10}$$

Moreover, because the left-hand sides of inequalities (9) and (10) coincide, we can now compare their right-hand sides to see if constraint j is relaxed in the transition from P^f to $P_{(i)}^f(\ell)$. Subtracting the right-hand side of (9) from the right-hand side of (10) yields the *degree of freedom*

$$f_{ji} = \tfrac{1}{2}|B_{ji}| + B_{ji}(y_i^f - \breve{y}_i^f) \tag{11}$$

that becomes available in constraint j of the problem $P_{(i)}^f(\breve{y}_i^f)$ due to fixing variable i. Note that $f_{ji} \in [0, |B_{ji}|]$ not only confirms Proposition 4.3a, but also shows that often some leverage is possible in constraint j. In fact, we only have $f_{ji} = 0$, if either $B_{ji} = 0$, or $|y_i^f - \breve{y}_i^f| = \frac{1}{2}$ holds, where in the latter case, additionally $y_i^f - \breve{y}_i^f$ needs to have the opposite sign as B_{ji}. Phrased differently, if a variable appears in a constraint with $|y_i^f - \breve{y}_i^f| \neq \frac{1}{2}$, also a strictly positive degree of freedom is possible. Moreover, even if $|y_i^f - \breve{y}_i^f| = \frac{1}{2}$ holds, due to the implied ambiguity of the rounding $\breve{y}_i^f$, one might be able to choose $\breve{y}_i^f$ such that $f_{ji} = |B_{ji}|$ holds. Thus, if a variable appears in a constraint, apart from degenerate cases, we can also expect the possibility of a strictly positive degree of freedom.

In the following, let $J_A \subseteq \{1, \dots, p\}$ denote the index set of rows of $Ax + By - ze \leqslant \widetilde{b} - \frac{1}{2}\|\beta\|_1$ that are active at (x^f, y^f, z^f). Moreover, let $f_i \in \mathbb{R}^{|J_A|}$ denote the vector with entries $f_{ji}, j \in J_A$, where $|J_A| \leqslant p$ denotes the cardinality of J_A.

The next lemma shows that progress towards feasibility due to fixing variables can be guaranteed for each variable i which has a strictly positive degree of freedom in all active constraints, that is, $f_i > 0$.

Lemma 4.4. *With an optimal point (x^f, y^f, z^f) of P^f and $v^f > 0$, for some $i \in \{1, \dots, m\}$ let $f_i > 0$. Then we have $v_{(i)}(\breve{y}_i^f) < v^f$.*

Proof. By Proposition 4.3a, the point $(x^f, (y^f)^{-i}, z^f)$ is feasible for $P_{(i)}^f(\breve{y}_i)$. As its objective value coincides with v^f, it suffices to show that it is not optimal for $P_{(i)}^f(\breve{y}_i)$.

Indeed, optimality of $(x^f, (y^f)^{-i}, z^f)$ requires the activity of at least one constraint of $(\widetilde{M}_L)_{(i)}(\breve{y}_i^f)^-$ where z^f occurs, that is, due to $z^f = v^f > 0$, inequality (10) holds with equality for some $j \in \{1, \dots, p\}$.

For $j \in J_A$ this is ruled out by our assumption $f_{ji} > 0$. Moreover, for $j \in \{1, \dots, p\} \setminus J_A$ inequality (9) is strictly satisfied. This, with $f_{ji} \geqslant 0$, implies that also inequality (10) is strictly satisfied. Hence, $(x^f, (y^f)^{-i}, z^f)$ cannot be optimal and the assertion is shown. □

The next example illustrates how the degree of freedom f_i may indeed guide us towards a successful diving step.

Example 4.5. Let us consider the feasible set

$$M = \{y \in \mathbb{B}^3 |\ y_1 + y_2 + 2y_3 \leqslant 2,\ -y_1 - y_2 - 2y_3 \leqslant -1,\ 2y_1 - y_2 - y_3 \leqslant 1\}.$$

Adding the first two constraints of the corresponding feasibility problem

$$\begin{aligned} P^f: \quad \min_{(y,z)\in\mathbb{R}^4} \ z \quad \text{s.t.} \quad y_1 + y_2 + 2y_3 - z &\leqslant \delta, \\ -y_1 - y_2 - 2y_3 - z &\leqslant -3 + \delta, \\ 2y_1 - y_2 - y_3 - z &\leqslant -1 + \delta, \\ z &\geqslant -1 \end{aligned}$$

yields the lower bound on the optimal value $z \geqslant \frac{3}{2} - \delta > 0$ which proves that M is not granular. This also shows that the P^f-feasible point $(y^f, z^f) = (\frac{1}{2} - \delta, \frac{1}{2} + \delta, 0.25, \frac{3}{2} - \delta)^\top$ which realizes this lower bound is optimal for P^f.

In the following, let us assume $\delta > 0$ so that the rounding of y^f is uniquely defined by $\breve{y}^f = (0, 1, 0)^\top$. Notice that $\breve{y}^f$ is a non-granular feasible rounding which is already useful if one is interested in computing *some* feasible point of M. Yet, to be able to compute feasible points with improved objective value, e.g. by using objective diving steps, a granular node is crucial so that a feasibility diving step still makes sense.

For the selection of a fixing variable, only the first two constraints are active in (y^f, z^f) independently of the choice of $\delta > 0$, that is, $J_A = \{1, 2\}$. Computing the degree of freedom thus yields the three positive vectors

$$f_1 = \begin{pmatrix} \frac{1}{2} + 1(\frac{1}{2} - \delta - 0) \\ \frac{1}{2} - 1(\frac{1}{2} - \delta - 0) \end{pmatrix},\ f_2 = \begin{pmatrix} \frac{1}{2} + 1(\frac{1}{2} + \delta - 1) \\ \frac{1}{2} - 1(\frac{1}{2} + \delta - 1) \end{pmatrix},\ \text{and } f_3 = \begin{pmatrix} 1 + 2(\frac{1}{4} - 0) \\ 1 - 2(\frac{1}{4} - 0) \end{pmatrix}.$$

To promote granularity, one usually chooses δ close to one (e.g. $1 - 10^{-4}$) and hence only fixing y_3 offers a notable degree of freedom for *both* constraints.

This positive degree of freedom is sufficient to yield a granular 3-0-child node. Indeed, when fixing $y_3 = 0$ we obtain the enlarged inner parallel set

$$\widetilde{M}_{(3)}(0)^- = \{y \in \mathbb{R}^2 |\ y_1 + y_2 \leqslant 1 + \delta,\ -y_1 - y_2 \leqslant -2 + \delta,\ 2y_1 - y_2 \leqslant -\tfrac{1}{2} + \delta\},$$

which is nonempty for any $\delta \geqslant \frac{1}{2}$ because it contains the feasible point $(\frac{1}{2}, \frac{1}{2} + \delta)^\top$.

On the other hand, the 1-0-fixed enlarged inner parallel set contains the two inequalities

$$y_2 + 2y_3 \leqslant \delta + \tfrac{1}{2},$$
$$-y_2 - 2y_3 \leqslant \delta - \tfrac{5}{2}.$$

Adding these constraints together with $\delta < 1$ again shows that they are unattainable and that we thus have $\widetilde{M}_{(1)}(0)^- = \varnothing$. Using the same arguments, one easily sees that $\widetilde{M}_{(2)}(1)^- = \varnothing$ holds as well so that deciding by the degree of freedom indeed seems to be a fruitful possibility for fixing variables.

For practical applications of larger dimensions, the requirement of Lemma 4.4 might often be too strict; a necessary condition which will often be violated is that *one* integer variable occurs in *every* active constraint. The next result shows how this requirement can be weakened, if we allow the flexibility to fix multiple variables in one diving step. Indeed, we will presently show that then it is sufficient if each active constraint contains at least one variable from a group of variables with a positive degree of freedom.

To state this formally, with $k \leqslant m$, an index set $\bar{I} = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, m\}$ and a set of corresponding integer values $\bar{L} = \{\ell_{i_1}, \ldots, \ell_{i_k}\}$, in the following let the $\bar{I}$-$\bar{L}$-fixed enlarged inner parallel set $\widetilde{M}_{(\bar{I})}(\bar{L})^-$ be defined analogously to Equations (6) and (7) where, instead of fixing one variable y_i to ℓ_i, we now fix each y_i with $i \in \bar{I}$ to the corresponding value $\ell_i \in \bar{L}$. Moreover, we extend this notation to the $\bar{I}$-$\bar{L}$-objective-based problem and the $\bar{I}$-$\bar{L}$-feasibility problem, as well as to their feasible sets, (rounded) optimal points and (optimal) objective values. For this purpose, $y_{\bar{I}} \in \mathbb{R}^{|\bar{I}|}$ denotes the vector with entries $y_i, i \in \bar{I}$.

We are again interested in values $\bar{L}$ that correspond to roundings of components of an P^f-optimal point, that is $\bar{L} = \{\breve{y}^f_{i_1}, \ldots, \breve{y}^f_{i_k}\}$. Then, a repeated application of Lemma 3.2 shows that $(x^f, (y^f)^{-\bar{I}}, z^f)$ is feasible for $P^f_{(\bar{I})}(\bar{L})$. Moreover, using the arguments from Equations (9) - (11), it is straightforward to see that the degree of freedom $f_{j\bar{I}}$ for constraint j which is available by fixing variables $i \in \bar{I}$ coincides with the sum of degrees of freedom of these variables, that is,

$$f_{j\bar{I}} = \sum_{i \in \bar{I}} f_{ji} = \sum_{i \in \bar{I}} \left(\tfrac{1}{2}|B_{ji}| + B_{ji}(y^f_i - \breve{y}^f_i) \right). \tag{12}$$

Again with $f_{\bar{I}} \in \mathbb{R}^{|J_A|}$ defined as the vector with entries $f_{j\bar{I}}$, $j \in J_A$, we can extend Lemma 4.4 to the following proposition.

Proposition 4.6. *With an optimal point (x^f, y^f, z^f) of P^f, for some $\bar{I} \subseteq \{1, \ldots, m\}$ let $f_{\bar{I}} > 0$. Then we have $v^f_{(\bar{I})}(\breve{y}^f_{\bar{I}}) < v^f$.*

For a variable i, let $J_i := \{j \in J_A |\ f_{ji} > 0\}$ denote the index set of constraints for which variable i has a strictly positive degree of freedom. Then with the set of active constraints where a positive degree of freedom is possible $J_U = \bigcup_{i \in \{1,\ldots,m\}} J_i$, there exists some index set $\bar{I}$ with $f_{\bar{I}} > 0$, if and only if $J_U = J_A$ holds. Therefore, by Proposition 4.6, $J_U = J_A$ is sufficient to ensure $v^f_{(\bar{I})}(\breve{y}^f_{\bar{I}}) < v^f$.

If this is the case, a natural task for a diving step is to find the minimum number of variables to fix such that progress towards feasibility is guaranteed. This question coincides with the set covering problem (cf., e.g., [14]), where J_U is the universe and $\{J_i|\ i \in \{1, \ldots, m\}\}$ is the collection of sets. This set covering problem is also of interest for $J_U \subsetneq J_A$. In this case, it minimizes the number of fixings which guarantees a positive degree of freedom for those active constraints for which a positive degree of freedom is possible.

As the set covering problem is NP-hard, solving this problem to optimality just for deciding which variables to fix seems to be out of order. Hence we suggest to use a greedy method instead, where theoretical results for worst case objective bounds on the greedy algorithm for set covering problems [6,11] make it a suitable choice for our purpose.

Applied to our context, the greedy algorithm starts with $k = 0$, $\bar{I}^0 = \varnothing$ and iteratively chooses a variable i_k so that J_{i_k} contains the largest number of uncovered elements of J_U, i.e.

$$i_k = \operatorname*{arg\,max}_{i \in \{1,\ldots,m\}} |\{j \in J_i|\ j \in J_U \setminus (\bigcup_{\bar{i} \in \bar{I}^k} J_{\bar{i}})\}|. \tag{13}$$

It then updates $\bar{I}^{k+1} = \bar{I}^k \cup i_k$ and $k = k + 1$.

For obtaining a feasible solution to the set covering problem, this is repeated until $J_U = \bigcup_{i \in \bar{I}^k} J_i$ holds. This leads to the fact that in each diving step the number of variables to be fixed may differ. If one is interested in specifying the number of variables to be fixed in each diving step, the greedy method can run some predefined number of iterations, fixing only the corresponding variables. We will specify this idea more precisely in our computational study. Let us next use the preceding considerations for the development of concrete algorithms.

4.3. An algorithmic framework for inner parallel set diving

In this section, we tie together considerations from the previous sections and illustrate how diving ideas can be used to extend and improve feasible rounding approaches. Like in the previous sections, we describe these methods as starting from the root node of a search tree but stress that this is for notational convenience only and that they can be applied in any node of a search tree. To ensure quick convergence, the suggested methods won't use any backtracking strategies but will dive straight to the leaves of the search tree.

We may either solve the problem P^{ob} or the problem P^f to determine if the enlarged inner parallel set of the root node is nonempty. If it is empty, we can apply feasibility diving steps as introduced in Section 4.2, until we possibly obtain a nonempty enlarged

Algorithm 1: Feasibility-IPS-diving.

Data: a mixed-integer optimization problem $MILP$
Result: a non-granularity measure v^{fd} with fixed variable-value pairs $\bar{I}$, $\bar{L}$, and, if successful, an $MILP$-feasible point $(\breve{x}^{fd}, \breve{y}^{fd})$

1 set $k \leftarrow 0$, $\bar{I}^k \leftarrow \varnothing$, $\bar{L}^k \leftarrow \varnothing$, $v^{fd} \leftarrow \infty$
2 **while** $v^{fd} > 0$ *and* $\bar{I}^k \subsetneq \{1, \ldots, m\}$ **do**
3 compute a minimal point (x^k, y^k, z^k) of

$$P^f_{(\bar{I}^k)}(\bar{L}^k): \quad \min_{(x,\widetilde{y},z)\in\mathbb{R}^n\times\mathbb{R}^{m-|\bar{I}^k|}\times\mathbb{R}} z \quad \text{s.t.} \quad (x, \widetilde{y}, z) \in (\widetilde{M}_L)_{(\bar{I}^k)}(\bar{L}^k)^-,$$

with merged rounding $(\breve{x}^f, \breve{y}^f)_{(\bar{I}^k)}(\bar{L}^k)$ and non-granularity measure $v^f_{(\bar{I}^k)}(\bar{L}^k)$
4 set $v^{fd} \leftarrow v^f_{(\bar{I}^k)}(\bar{L}^k)$
5 **if** $(\breve{x}^f, \breve{y}^f)_{(\bar{I}^k)}(\bar{L}^k) \in M$ **then**
6 $(\breve{x}^{fd}, \breve{y}^{fd}) \leftarrow (\breve{x}^f, \breve{y}^f)_{(\bar{I}^k)}(\bar{L}^k)$
7 **end**
8 choose a set of indices $I^k \subseteq \{1, \ldots, m\} \setminus \bar{I}^k$
9 set $\bar{I}^{k+1} \leftarrow \bar{I}^k \cup I^k$, $\bar{L}^{k+1} \leftarrow \bar{L}^k \cup \{\widetilde{y}^k_{i_k} \mid i_k \in I^k\}$, $k \leftarrow k+1$
10 **end**
11 set $\bar{I} \leftarrow \bar{I}^{k-1}$, $\bar{L} \leftarrow \bar{L}^{k-1}$

inner parallel set of some child node. The detailed procedure, feasibility-**I**nner**P**arallel**S**et-diving, is outlined in Algorithm 1 and can be summarized as follows.

In each iteration k, we fix variables to roundings of optimal points of the $\bar{I}^k$-$\bar{L}^k$-fixed feasibility problem. Recall that we obtain a nonempty $\bar{I}^k$-$\bar{L}^k$-fixed enlarged inner parallel set, if and only if the optimal value $v^f_{(\bar{I}^k)}(\bar{L}^k)$ of $P^f_{(\bar{I}^k)}(\bar{L}^k)$ is less or equal than zero, and that obtaining an $MILP$-feasible point is possible even if $v^f_{(\bar{I}^k)}(\bar{L}^k) > 0$ holds (cf. Example 4.5). Hence we check if $(\breve{x}^f, \breve{y}^f)_{(\bar{I}^k)}(\bar{L}^k)$ is feasible for $MILP$ in every iteration and, if this is the case, store it (cf. Line 6) so that a feasible point can be returned after termination of the method even in the non-granular case.

The method terminates when the optimal value of the $\bar{I}^k$-$\bar{L}^k$-fixed feasibility problem is nonpositive, or when all variables are fixed. For choosing a set of indices to be fixed in Line 8, one possibility is to use the greedy algorithm aiming at impacting as many active constraints as possible.

If feasibility-IPS-diving terminates with an index set $\bar{I}$ and a corresponding value set $\bar{L}$ such that $v^{fd} \leqslant 0$ holds, the $\bar{I}$-$\bar{L}$-fixed objective-based problem is consistent and we can apply objective-based diving steps. Note that the case $\bar{I} = \bar{L} = \varnothing$ corresponds to a granular root node.

This is the starting point for Algorithm 2, which outlines a method that takes as input a nonempty $\bar{I}$-$\bar{L}$-fixed enlarged inner parallel set and aims at obtaining a feasible point $(\breve{x}^{obd}, \breve{y}^{obd})$ with improved objective value $\breve{v}^{obd}$ for the bounded problem $MILP$. We remark that the boundedness assumption is only for the sake of readability and our focus on computing good feasible points. In fact, Algorithm 2 could be modified to encompass unbounded MILPs as well by additionally checking if $P^{ob}_{(\bar{I})}(\bar{L})$ is unbounded and, if this is the case, returning a certificate for unboundedness of $MILP$.

Algorithm 2: Objective-IPS-diving.

Data: a bounded mixed-integer optimization problem *MILP*, an index set $\bar{I}^0$ and corresponding values $\bar{L}^0$ such that $\widetilde{M}_{(\bar{I}^0)}(\bar{L}^0)^- \neq \varnothing$

Result: a good *MILP*-feasible point $(\breve{x}^{obd}, \breve{y}^{obd})$ with objective value $\breve{v}^{obd}$

1 set $k \leftarrow 0,\ \breve{v}^{obd} \leftarrow +\infty$

2 **while** *some quality criterion is not met and* $\bar{I}^k \subsetneq \{1, \ldots, m\}$ **do**

3 compute an optimal point (x^k, y^k) of the problem

$$P^{ob}_{(\bar{I}^k)}(\bar{L}^k): \min_{(x,\tilde{y}) \in \mathbb{R}^n \times \mathbb{R}^{m-|\bar{I}^k|}} c^\top x + (d^{-\bar{I}^k})^\top \tilde{y} + \sum_{i \in \bar{I}^k} d_i \ell_i \quad \text{s.t.} \quad (x, \tilde{y}) \in \widetilde{M}_{(\bar{I}^k)}(\bar{L}^k)^-,$$

with merged rounding $(\breve{x}^{ob}, \breve{y}^{ob})_{(\bar{I}^k)}(\bar{L}^k)$ and its objective value $\breve{v}^{ob}_{(\bar{I}^k)}(\bar{L}^k)$

4 **if** $\breve{v}^{ob}_{(\bar{I}^k)}(\bar{L}^k) < \breve{v}^{obd}$ **then**

5 $(\breve{x}^{obd}, \breve{y}^{obd}) \leftarrow (\breve{x}^{ob}, \breve{y}^{ob})_{(\bar{I}^k)}(\bar{L}^k)$

6 $\breve{v}^{obd} \leftarrow \breve{v}^{ob}_{(\bar{I}^k)}(\bar{L}^k)$

7 **end**

8 choose a set of indices $I^k \subseteq \{1, \ldots, m\} \setminus \bar{I}^k$

9 set $\bar{I}^{k+1} \leftarrow \bar{I}^k \cup I^k,\ \bar{L}^{k+1} \leftarrow \bar{L}^k \cup \{\breve{y}^k_{i_k} \mid i_k \in I^k\},\ k \leftarrow k+1$

10 **end**

Boundedness of *MILP* implies that every problem $P^{ob}_{(\bar{I}^k)}(\bar{L}^k)$ is also bounded. Moreover, consistency of $P^{ob}_{(\bar{I}^k)}(\bar{L}^k)$ follows from Lemma 3.2 together with the consistency of the initial $\bar{I}$-$\bar{L}$-fixed enlarged inner parallel set. Hence we can iteratively compute rounded optimal points of $\bar{I}^k$-$\bar{L}^k$-fixed objective-based problems. If the objective value $\breve{v}^{ob}_{(\bar{I}^k)}(\bar{L}^k) = c^\top \breve{x}^k + d^\top \breve{y}^k + \sum_{i \in \bar{I}^k} d_i \ell_i$ of the rounded (and merged) optimal point $(\breve{x}^{ob}, \breve{y}^{ob})_{(\bar{I}^k)}(\bar{L}^k)$ improves upon that of previously found points, the latter is stored in Line 5.

Let us conclude this section with a few remarks on the choice of indices in Line 8. We only derived sufficient conditions for progress in the objective value of the feasibility problem in Proposition 4.6, but similar ideas apply to the objective-based problem as well. In particular, the sets $\widetilde{M}^-$ and $\widetilde{M}^-_L$ as well as their i-ℓ-fixed counterparts only differ in the appearance of the variable z. Therefore, by using equations (9) and (10) without the occurrence of z, we see that the degree of freedom gained in the transition from $\widetilde{M}^-$ to $\widetilde{M}^-_{(i)}(\ell)$ exactly coincides with (11). Yet, notice that while the optimal value $v^f_{(\bar{I})}(\bar{L})$ is meaningful in the sense that it contains information about the degree of non-granularity, this is not the case for the value $v^{ob}_{(\bar{I})}(\bar{L})$. Indeed, within the framework of objective-IPS-diving, we would rather be interested in certifying progress of the objective value of the rounded optimal point $\breve{v}^{ob}_{(\bar{I})}(\bar{L})$. Yet, due to the appearance of the term $\sum_{i \in \bar{I}} d_i \ell_i$ in the objective function as well as due to rounding effects, this is more intricate and not easy to predict. Still, choosing indices in accordance with equation (13) offers new flexibility in the constraints and is thus likely to enable the possibility of obtaining different roundings, which might be beneficial for obtaining new (and hopefully improved) values $v^f_{(\bar{I})}(\bar{L})$.

5. Computational study

The main intention of our computational study is to show that feasible rounding approaches can benefit from applying diving steps as outlined in Algorithms 1 and 2. In particular, our results show that the granularity concept can be extended to encompass more problems by using feasibility-IPS-diving, and that the objective values of the feasible points generated in the root node can be improved by applying objective-IPS-diving steps.

Corollary 3.4 states that the number of roundings is nondecreasing and *potentially increasing* with increasing depth of the search tree. Our analysis shows that we can actually expect to obtain an *increasing* number of feasible roundings via inner parallel sets in branch-and-bound trees for practical problems.

Additionally, we examine the influence of different choices of fixing variables and demonstrate that the introduced greedy strategy for feasibility-IPS-diving usually finds granular nodes faster than a random strategy. Finally, we address the important question whether the generated points can add value to the arsenal of primal heuristics, using the example of the solver framework SCIP [9].

The test bed of our computational study stems from the collection set of the MIPLIB 2017 [10]. We have collected instances without equality constraints on integer variables, as the latter need a special treatment when feasible rounding approaches are applied (cf. [15]). We further discarded problems with special constraint types (indicator constraints) which would also need a special treatment.

We have implemented the feasible rounding approaches with diving strategies outlined in Algorithms 1 and 2 in Matlab R2020a and in SCIP 7.0 [9] with SoPlex 5.0.0 using the PySCIPOpt interface [13]. The tests of Section 5.2 were run on an Intel i7 processor with 4 cores running at 4 GHz Turbo Boost and 16 GB of RAM and those of Section 5.3 were run on an Intel i7 processor with 8 cores with 3.60 GHz and 32 GB of RAM.

Before we report the results of our computational study, we initially clarify the selection of variables in the diving steps. Subsequently, in the first part of our computational study, we evaluate the improvements gained by feasibility- and objective-IPS-diving compared to the root node using our Matlab implementation. We conclude our study with evaluating the possible benefit of integrating feasible rounding approaches and diving ideas into the solver framework SCIP. In this last part of our study, we focus on objective-IPS-diving for problems which are root-node granular.

When closely related LPs are solved in sequence, the use of warm-start ideas is common in the literature and usually very beneficial. In our context, optimal points of objective-based problems are *feasible* for I-L-fixed objective-based problems but usually lie in the interior of their feasible sets. Finding a good warm-start basis for the simplex algorithm is therefore nontrivial. As our aim is to demonstrate the potential of the method, in our computational study we solved each LP from scratch. We postpone a thorough study of warm-start capabilities, which might be able to significantly speed up the diving procedure, to future research.

5.1. Selection of variables

Recall that the flexibility of our diving method introduced in Section 4.3 lies in the choice of the variables to fix. We propose and evaluate two methods for this. The first is to select fixing variables at random. The second method is to run the greedy algorithm in each iteration, choosing the variables to be fixed according to Equation (13). To avoid collecting variable constraint pairs (i, j) with trivial degrees of freedom, we set $J_i := \{j \in J_A | \ f_{ji} > 10^{-4}|B_{ji}|\}$ (cf. Equation (11)).

To ensure comparability, we fix $k = \lceil m/30 \rceil$ variables in each step for the greedy as well as for the random method. This guarantees at most 30 rounds of fixing even if all child nodes are non-granular.

For the greedy algorithm, this has two effects. First, the number of fixings might not be enough to cover all active constraints. Second, we might have covered all active constraints with less than k variables so we need a secondary selection criterion. Concerning the latter, we decided to use the overall impact on all active constraints, that is, once all constraints were covered, we selected remaining variables i according to their overall impact

$$f_{J_A i} = \sum_{j \in J_A} f_{ji} = \sum_{j \in J_A} \tfrac{1}{2}|B_{ji}| + B_{ji}(y_i^f - \breve{y}_i^f).$$

Note that, while $f_{j\bar{I}} \in \mathbb{R}$ denotes the impact on constraint j of fixing all variables from $\bar{I}$, the value $f_{J_A i} \in \mathbb{R}$ stands for the added up impact on all active constraints of fixing variable i.

5.2. Improvement due to IPS-diving steps

In this section, we investigate the effectiveness of IPS-diving ideas. First, we evaluate feasibility-IPS-diving steps by comparing the number of (root node) granular problems to that of problems where our diving strategies found *some* granular node. Second, we show that applying objective-IPS-diving steps yields feasible points with improved objective value compared to the root node and assess the significance of this improvement.

Discarding problems with indicator constraints as well as problems with equality constraints on integer variables yielded 288 instances. We ran into memory problems with the instance *supportcase38* and observed numerical instabilities with the instance *npmv07* (points obtained from Gurobi did not satisfy several equality constraints with the expected feasibility tolerance of 1e-06). Removing these two problems yields a test bed of 286 instances for which we found 136 to be granular in the root node.

Using feasibility-IPS-diving with both diving strategies, we were able to find granular nodes for 167 problems so that the share of problems for which we may compute granularity based feasible points increases from 47.6% (root node only) to 58.4% (using feasibility-IPS-diving). Viewed from another perspective, from the 150 non-granular

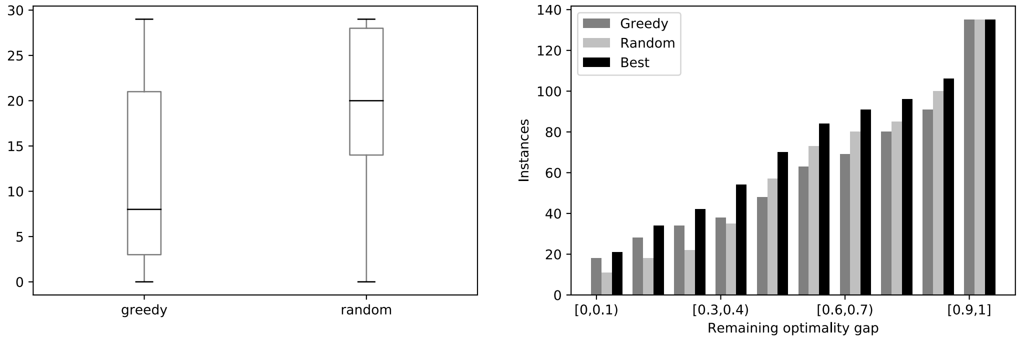

Fig. 3. Two plots comparing the success of both diving strategies. The left plot visualizes statistics for the number of explored nodes until a granular node is found for 25 non-granular instances. The right plot shows a cumulative histogram of the remaining optimality gap compared to the optimality gap in the root node (14) for 120 granular instances.

instances, feasibility-IPS-diving was successful in 31 cases, which amounts to a success rate of 20%. Thus our first finding is that using diving steps increases the applicability of the granularity concept. We report detailed results for these 167 problems in Table 2 in the appendix.

As a comparison of the two methods for selecting fixing variables (random and greedy), we can state that for 25 non-granular instances both diving methods are able to find granular nodes. The random strategy finds a granular node in four additional cases, and in two cases only the greedy strategy yields a granular node. This shows that different orders of fixing indeed yield different outcomes and points to the fact that different strategies can be complementary. Concerning the chances of finding *some* granular node, the greedy strategy does not seem to offer an advantage over randomly fixing variables.

Yet, as the boxplot of the number of iterations of both methods shown on the left-hand side in Fig. 3 reveals, the greedy method usually finds granular nodes much earlier in the search tree. Indeed, for the 25 instances where both methods yield a granular node, the median number of iterations is eight for the greedy and 20 for the random method and also the 25th and 75th percentiles differ significantly (three vs. 14 and 21 vs. 28 iterations). Additionally, a direct comparison of the number of iterations for each problem individually shows that the greedy method needs (often significantly) less iterations than the random method in 16 cases and more iterations in only two cases. Hence, if we are interested in quickly finding granular nodes, the greedy strategy seems to be the better choice.

For the 136 (root-node) granular problems, we can compute and evaluate the improvement yielded by objective-IPS-diving. In this regard, with $v^\star$ denoting the optimal (or best known) value obtained from the MIPLIB 2017 website,[1] for each problem with $\breve{v}^{ob} \neq v^\star$ we compute the value

[1] https://miplib2017.zib.de/tag_collection.html.

$$gap_{closed} = (\breve{v}^{ob} - \breve{v}^{obd})/(\breve{v}^{ob} - v^{\star}), \tag{14}$$

which measures the optimality gap closed by IPS-diving steps (recall that $\breve{v}^{ob}$ and $\breve{v}^{obd}$ stand for the objective values of the points obtained by solving the objective-based problem in the root node and by applying objective-IPS-diving, respectively). This ratio is one, if and only if objective-IPS-diving finds an optimal point, and zero, if there is no improvement in the objective value.

For the instance *p500x2988d* the rounding of the optimal point of the objective-based problem $(\breve{x}^{ob}, \breve{y}^{ob})$ was already optimal for *MILP* and we therefore subsequently analyze only the remaining 135 problems. For all these problems, the value of $\breve{v}^{ob} - v^{\star}$ was above 0.25 and the values for gap_{closed} were hence well-defined.

The right-hand side of Fig. 3 summarizes our results by plotting a cumulative histogram of the number of instances over the remaining optimality gap, that is, over $1 - gap_{closed}$. It includes the gap closed by both strategies individually, as well as a third option *best*, which is the gap closed *collectively* by both strategies. This can be seen as a scenario where we run both diving strategies in the root node and use the best feasible point.

We find that for 70 of the 135 problems, more than half of the optimality gap is closed by applying both diving strategies. For the random and the greedy method individually, this is the case for 57 and 48 problems, respectively. Moreover, in our test bed the greedy strategy performed better than the random strategy in closing more than 80% of the optimality gap (shown in the first and second set of bars), but the random strategy outperformed the greedy method with respect to closing more moderate optimality gaps, e.g., the above mentioned 50%. This again demonstrates that it is beneficial to apply different fixing strategies from the root node.

Overall we conclude that combining feasible rounding approaches with diving strategies yields a significant improvement over their application in the root node only. The greedy fixing method is particularly promising for finding granular nodes early, yet when performing objective-IPS-diving steps at granular nodes it does not offer an advantage over randomly fixing variables.

5.3. Integration of feasible rounding approaches and diving ideas into a solver framework

In a second experiment, we study the potential benefit of integrating feasible rounding approaches with and without diving steps into a solver framework. We use SCIP for this purpose and initially evaluate the quality of the generated feasible points compared to the best solution SCIP obtains with its various heuristics [3] after solving the root node. To this end, we test objective-IPS-diving using the PySCIPOpt interface by including it as a primal heuristic. The method is executed once, after the processing of the root node is finished. After fixing variables in our diving approaches, the variable bound tightening techniques implemented in SCIP are applied to further reduce the domain of other

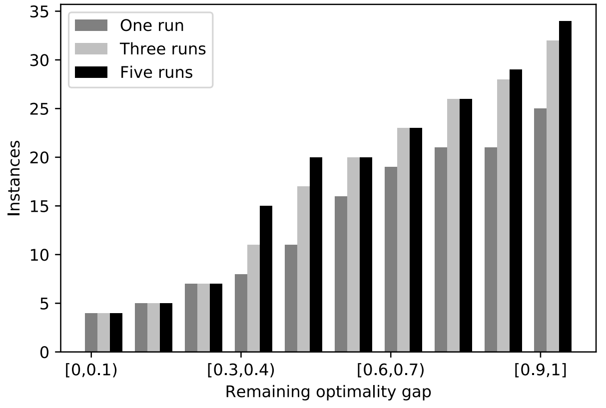

Fig. 4. Cumulative histogram of the remaining optimality gap (14) compared to the optimality gap of SCIP's best solution in the root node for 34 instances where five dives yield best solutions.

variables. Thus many variables can be fixed additionally, which indeed led to a significant reduction of the number of diving nodes needed to be explored. The python code for these experiments is publicly available at https://github.com/schwarze-st/FRA_BB.

We focus our analysis on instances where the root node is granular and apply up to five dives using the random strategy with different seeds. As described in Section 5.1, in each run we need to solve at most 30 linear optimization problems (of decreasing size).

The test bed again contains granular problems from the MIPLIB 2017. We collected all instances without equality constraints on integer variables where SCIP executed feasible rounding approaches within 30 minutes. Discarding 8 instances due to memory limitations yielded 320 instances out of which 128 were granular. Thus, we obtained 128 problems that are granular after SCIP completely processed the root node of the branch-and-bound search. Apart from the instances *buildingenergy, ramos3, scpj4scip, scpk4, and scpl4*, we were able to apply five dives within the time limit of 30 minutes. For those five problems, we use the best solution found within 30 minutes for our analysis.

We report that in 16 cases the feasible point obtained by solving the root node is able to improve upon those previously found by SCIP. By applying one, three and five random dives, this number is increased to 25, 32 and 34, respectively. Detailed results for the 34 instances where five dives yield better solutions than the ones found by SCIP can be found in Table 3 in the appendix. The significant increase in best solutions again highlights the potential of applying diving steps when using feasible roundings approaches. Moreover, the number of best solutions increases significantly when three dives are applied (compared to one). With five random dives, we only obtain two additional best solutions (compared to three dives) which suggests that three dives might be a good compromise between effort and benefit of the method.

This is further substantiated in Fig. 4, which gives an impression of the relative improvement compared to SCIP's best solutions in a cumulative histogram. Once more we display the remaining optimality gap $1 - gap_{closed}$, where the $\breve{v}^{ob}$ in Equation (14) is replaced with the best solution of SCIP. For 17 problems, more than half of the optimality gap is closed by the points obtained with three rounds of diving. For one diving round, this is the case only for nine problems which again shows that running objective-IPS-diving more than one time can be beneficial.

To give a broader impression of the potential of integrating these methods into SCIP, we ran a second experiment. Here, we compared the time SCIP needs to compute a feasible point of similar quality without integrating feasible rounding approaches with diving steps for the 32 problems where three rounds of diving yield best solutions. To this end, we executed SCIP with plain settings, and report the run time when SCIP finds a feasible point of similar quality for the first time.

We report that SCIP needs additional time to compute such a feasible point in 27 cases compared to the pure run time of our diving heuristic. To give an impression of the potential benefit of the method, we list the 12 instances in Table 1 where SCIP needs more than 30 seconds additional time to compute a feasible point of similar quality. Here we list the objective value obtained by feasible rounding approaches in the root node and after three diving rounds, as well as the optimal (or best known) objective value. As a comparison, we list the time for these three diving rounds and that of SCIP for computing a feasible point of similar quality. Due to the so-called performance variability of global solvers [12], we checked our results for robustness by running SCIP additionally with five different LP random seeds and reporting the shortest, median and longest run time among these five. For the runs with default LP random seed, we also report the time SCIP spends in primal heuristics only, as a lower bound for the time SCIP needs for the computation of feasible points of similar quality. Additionally, we report which heuristic delivered a point of similar or better quality and list its total run time.

In most instances from Table 1, the additional time SCIP needs to compute a point of similar quality is quite significant, even when we investigate the best of five runs and the time only spent in heuristics. In five cases, the run time of our diving heuristic is even shorter than the run time of the best performing SCIP heuristic, which found a point of similar or better quality. This is particularly true for the problem *gsvm2rl12*, where SCIP fails to find such a point within 30 minutes. Interestingly, for this problem the objective value is already available without applying diving steps and even applying three rounds of diving takes no longer than 30 seconds while SCIP spends 949.4 seconds in primal heuristics and does not produce a feasible point of similar quality. Even in the best of five case, SCIP needs 729.7 seconds to find such a point so that feasible rounding approaches offer a remarkable improvement for that problem.

We can further conclude that there is not one dominant SCIP heuristic, as different methods are listed in the *best* column. We stress that comparing our run time to this column is disadvantageous for our method, as it only starts in the root node and does not exploit further information. However, for *b2c1s1*, *neos-983171*, and both *opm2* instances, the best heuristic is called exactly once and in the root node which makes their run times more comparable. For the instances *opm2* we see a remarkable run time improvement and, in contrast to the problem *gsvm2rl12*, here the diving steps are crucial. For both instances of the problem *sorell* diving steps are also crucial and SCIP needs much more

Table 1
Instances selected from root node granular problems, where SCIP needs significantly more time to compute a feasible point with similar quality.

name	objective			time	time SCIP full				time SCIP heuristics		
	root	diving	optimal	diving	one run	best	median	worst	all	best	name best
b1c1s1	72555.0	69071.5	24544.2	0.8	34.87	34.9	34.9	35.0	7.9	1.9	feaspump
b2c1s1	73676.5	68701.5	25687.9	1.5	44.91	44.9	45.0	45.1	9.9	2.1	feaspump
dg012142	25623489.0	14373382.6	2300867.0	6.8	40.71	40.7	40.7	40.8	8.4	0.2	oneopt
gsvm2rl11	42635.3	39792.8	18121.6	15.9	64.28	63.7	64.2	64.3	19.7	1.3	oneopt
gsvm2rl12	34.4	34.4	22.1	29.4	1800.00	729.7	1800.0	1800.0	949.4	–	–
gsvm2rl9	16382.8	13611.9	7438.2	3.9	365.03	51.3	103.8	365.2	94.5	4.5	pscostdiv.
mushroom-best	3613.9	2072.9	0.1	11.4	79.58	5.5	62.3	135.4	17.7	2.7	distr.-div.
neos-983171	50987.0	8747.0	2360.0	84.7	227.77	207.4	231.1	283.0	129.0	68.8	rens
opm2-z10-s4	−1489.0	−22681.0	−33269.0	18.2	295.14	294.6	297.4	299.1	200.6	199.2	farkasdiv.
opm2-z8-s0	−2220.0	−11328.0	−15775.0	6.7	50.00	49.8	50.0	50.0	23.6	23.2	farkasdiv.
sorrell7	−45.0	−160.0	−196.0	622.1	1753.36	1290.5	1800.0	1800.0	39.5	10.9	pscostdiv.
sorrell8	−168.0	−324.0	−350.0	11.8	547.90	538.0	564.1	589.5	7.9	0.6	pscostdiv.

total run time. For these problems, the time spent in heuristics and especially in the successful diving heuristic (which is called in each case once in a node of depth six respectively seven) is quite low so the benefit of using feasible rounding approaches with diving steps within SCIP is less clear. Besides *pscostdiving*, seven other diving procedures are called on average 32 times before one is successful for *gsvm2rl9*, which shows a clear benefit for this instance.

Overall, our results show that in some cases feasible rounding approaches combined with the introduced diving ideas can be beneficial and help state-of-the-art software to compute good feasible points more quickly. While this improvement is possible in the root node (e.g. in the case of *gsvm2rl12*), the application of diving steps makes it significantly more likely.

6. Conclusion and outlook

In this article we developed new theoretical insights into how inner parallel sets change within branch-and-bound methods. This allowed us to show that the number of roundings obtainable within a search tree is nondecreasing with increasing depth of the search tree. Moreover, we provided examples that demonstrate that this number can actually increase.

Based on these results we developed a novel diving method for MILPs with two remarkable features. First, applying an objective-based diving step to a granular node retains granularity. Second, the measure of non-granularity of a feasibility diving step in a non-granular node cannot deteriorate after the application of this step. In the latter case we additionally derived sufficient conditions for an actual improvement in the measure of non-granularity.

Our computational study on problems from the MIPLIB 2017 shows two main benefits of the diving methods. First, a considerable number of 31 out of 150 instances is not granular in the root node but becomes granular in some child node explored by our diving strategies. Indeed, the share of instances for which we were able to exploit the granularity concept increased from 47.6% to 58.4% when diving steps were applied. Second, for root-granular instances, our evaluation of the closed optimality gap shows that objective diving steps are able to significantly improve the quality of feasible roundings compared to the root node. This second effect is further substantiated by a comparison with SCIP, where the number of best incumbent solutions provided by feasible rounding approaches is significantly increased when objective-IPS-diving steps are applied.

Both benefits not only confirm the effectiveness of the diving method, but also show that the number of roundings obtainable with feasible rounding approaches can be expected to be increasing with the exploration of a branch-and-bound tree. This fact could be further exploited by integrating suitable backtracking strategies into the diving procedure, which we postpone to future research.

We further wish to point out that within the scope of our diving approaches, the appearing linear optimization problems solved sequentially are closely related. Therefore, it might be interesting to investigate warm-start possibilities, which we also leave for future research.

Moreover, while our theoretical results on the number of roundings hold in the general MINLP setting, the diving heuristic is made explicit and studied computationally for MILPs only. The transfer of most ideas from the linear to the nonlinear setting is straightforward. The only exception is the computation of the degree of freedom where the polyhedrality of the feasible set is used explicitly. The derivation of a similar measure in the nonlinear setting which potentially includes information of the possibility of using smaller Lipschitz constants after variables are fixed is also left to future research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors are grateful to Alexander Martin for initially suggesting to investigate the effects of fixing variables on inner parallel sets, and to two anonymous reviewers for their precise and constructive remarks which helped to significantly improve this paper.

Appendix A. Detailed computational results

See Tables 2 and 3 below.

References

[1] Tobias Achterberg, Timo Berthold, Improving the feasibility pump, Discrete Optim. 4 (1) (2007) 77–86.
[2] Timo Berthold, RENS, Math. Program. Comput. 6 (1) (2014) 33–54.
[3] Timo Berthold, Heuristic algorithms in global MINLP solvers: Zugl., Techn. Univ., Berlin, 2014, Diss.; 1. aufl. edition, Verl. Dr. Hut, München, 2015.
[4] Timo Berthold, Ambros M. Gleixner, Undercover: a primal MINLP heuristic exploring a largest sub-MIP, Math. Program. 144 (1–2) (2014) 315–346.
[5] Pierre Bonami, João P.M. Gonçalves, Heuristics for convex mixed integer nonlinear programs, Comput. Optim. Appl. 51 (2) (2012) 729–747.
[6] Vasek Chvatal, A greedy heuristic for the set-covering problem, Math. Oper. Res. 4 (3) (1979) 233–235.
[7] Matteo Fischetti, Fred Glover, Andrea Lodi, The feasibility pump, Math. Program. 104 (1) (2005) 91–104.
[8] Matteo Fischetti, Domenico Salvagnin, Feasibility pump 2.0, Math. Program. Comput. 1 (2–3) (2009) 201–222.
[9] Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, Gregor Hendel, Christopher Hojny, Thorsten Koch, Pierre Le Bodic, Stephen J. Maher, Frederic Matter, Matthias Miltenberger, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Christine Tawfik, Stefan Vigerske, Fabian Wegscheider, Dieter Weninger, Jakob Witzig, The SCIP Optimization Suite 7.0: Technical Report.

Table 2
Instances with some granular node, corresponding objective values and number of feasibility diving iterations.

	granular			objective				iterations	
	root	greedy	random	root	greedy	random	best known	greedy	random
30_70_45_05_100	True	True	True	8985.0	4730.0	4498.0	9.0	0	0
30_70_45_095_100	True	True	True	8870.0	3956.0	3918.0	3.0	0	0
30_70_45_095_98	True	True	True	9158.0	4417.0	4169.0	12.0	0	0
50v-10	True	True	True	199236.7	30232.1	97076.9	3311.2	0	0
CMS750_4	False	True	False	inf	779.0	inf	252.0	4	30
a1c1s1	True	True	True	21033.2	21029.4	21029.4	11503.4	0	0
a2c1s1	True	True	True	20866.1	20865.3	20865.3	10889.1	0	0
ab51-40-100	True	True	True	−1023114612.0	−3757026942.0	−2661503127.0	−10420305975.0	0	0
ab67-40-100	True	True	True	−1121661829.0	−5902982915.0	−5315400098.0	−11186253442.0	0	0
ab69-40-100	True	True	True	−1078937214.0	−5799631585.0	−5342401322.0	−11186281442.0	0	0
ab71-20-100	True	True	True	−1823703279.0	−6233051084.0	−3766441410.0	−10420305975.0	0	0
ab72-40-100	True	True	True	−1144435108.0	−3849272486.0	−4851918405.0	−11186620442.0	0	0
app3	False	False	True	inf	inf	6449642.4	5751714.3	25	25
australia-abs-cta	False	True	True	inf	4562.2	9854.3	106.9	30	30
b1c1s1	True	True	True	69466.5	69376.9	69336.1	24544.2	0	0
b2c1s1	True	True	True	67451.0	67250.0	66085.5	25687.9	0	0
beasleyC1	True	True	True	102.0	102.0	102.0	85.0	0	0
beasleyC2	True	True	True	232.0	232.0	217.0	144.0	0	0
beasleyC3	False	True	True	6844.0	964.0	4710.0	754.0	1	20
berlin	True	True	True	1921.0	1921.0	1321.0	1044.0	0	0
berlin_5_8_0	False	True	True	inf	95.0	92.0	62.0	4	15
bg512142	True	True	True	3968845.9	278117.6	267769.7	184202.8	0	0
bmocbd	True	True	True	−32041010095048.5	−32041010095048.5	−32041010095048.5	−32041010095050.0	0	0
bmocbd2	True	True	True	−31953009810558.2	−31953009810558.2	−31953009810558.2	−31953010000000.0	0	0
bmocbd3	True	True	True	24865940299999.4	49057829841.7	61288920456.7	−372986719.7	0	0
bmoipr2	False	True	True	383315500000.0	108118487164.1	164298702888.9	−46416168.3	20	30
brasil	True	True	True	32720.0	32720.0	19702.0	13655.0	0	0
buildingenergy	True	True	True	34305.7	34196.5	34197.2	33283.9	0	0
cdc7-4-3-2	True	True	True	−1.0	−127.0	−128.0	−289.0	0	0
cod105	True	True	True	0.0	−3.0	−4.0	−12.0	0	0
core2536-691	False	False	True	inf	inf	7101.0	689.0	30	9
cost266-UUE	True	True	True	42188320.7	42008592.2	42014519.8	25148940.6	0	0
cvs08r139-94	True	True	True	−85.0	−85.0	−85.0	−116.0	0	0
cvs16r106-72	True	True	True	−25.0	−49.0	−36.0	−81.0	0	0
cvs16r128-89	True	True	True	−78.0	−78.0	−78.0	−97.0	0	0
cvs16r70-62	True	True	True	−15.0	−17.0	−15.0	−42.0	0	0
cvs16r89-60	True	True	True	−19.0	−31.0	−27.0	−65.0	0	0

(*continued on next page*)

Table 2 (*continued*)

	granular			objective				iterations	
	root	greedy	random	root	greedy	random	best known	greedy	random
dale-cta	False	True	True	inf	584.2	584.2	0.0	30	30
dg012142	True	True	True	77158148.1	12143833.8	12345726.7	2300867.0	0	0
ex1010-pi	True	True	True	8596.0	2832.0	4309.0	235.0	0	0
fast0507	True	True	True	55637.0	2586.0	28261.0	174.0	0	0
g200x740	True	True	True	45614.0	45558.0	45614.0	44316.0	0	0
gen-ip002	True	True	True	−3543.6	−4573.4	−4550.5	−4783.7	0	0
gen-ip016	True	True	True	−7700.6	−8783.1	−9052.6	−9476.2	0	0
gen-ip021	True	True	True	3014.7	2738.2	2478.3	2361.5	0	0
gen-ip036	True	True	True	−3827.2	−4552.4	−4500.0	−4606.7	0	0
gen-ip054	True	True	True	11138.2	7593.9	7923.5	6841.0	0	0
ger50-17-ptp-pop-3t	True	True	True	14477.0	13687.2	9191.8	5231.1	0	0
ger50-17-ptp-pop-6t	True	True	True	17651.4	16456.0	11865.1	8942.6	0	0
ger50-17-trans-dfn-3t	True	True	True	553623.6	488587.9	257477.4	3969.4	0	0
ger50-17-trans-pop-3t	True	True	True	553803.5	493766.7	259576.4	4038.4	0	0
ger50__17__trans	True	True	True	555975.2	499809.5	260518.2	7393.3	0	0
germany50-UUM	True	True	True	751380.0	689860.0	666830.0	628490.0	0	0
glass-sc	True	True	True	74.0	46.0	51.0	23.0	0	0
gr4x6	True	True	True	284.1	252.2	222.2	202.3	0	0
gsvm2rl11	True	True	True	43040.0	42693.9	41964.4	18121.6	0	0
gsvm2rl12	True	True	True	50.0	42.2	36.7	22.1	0	0
gsvm2rl3	True	True	True	0.6	0.6	0.6	0.3	0	0
gsvm2rl5	True	True	True	10.0	10.0	10.0	5.4	0	0
gsvm2rl9	True	True	True	15597.5	15597.5	14896.9	7438.2	0	0
iis-glass-cov	True	True	True	73.0	45.0	50.0	21.0	0	0
iis-hc-cov	True	True	True	78.0	52.0	50.0	17.0	0	0
in	False	True	True	inf	450.0	1489.0	58.0	9	30
istanbul-no-cutoff	False	True	True	330.3	330.3	330.3	204.1	25	30
k16x240b	True	True	True	12874.0	12874.0	12779.0	11393.0	0	0
khb05250	True	True	True	126786075.5	120511827.0	122443058.0	106940226.0	0	0
manna81	True	True	True	−12867.0	−13162.0	−13162.0	−13164.0	0	0
mas74	True	True	True	736774.2	50264.5	28886.9	11801.2	0	0
mas76	True	True	True	782652.6	69745.4	64247.9	40005.1	0	0
mc11	True	True	True	13548.0	13548.0	13167.0	11689.0	0	0
mc7	True	True	True	5884.0	5875.0	4740.0	3417.0	0	0
mc8	True	True	True	1994.0	1977.0	1865.0	1566.0	0	0
mik-250-20-75-1	True	True	True	0.0	0.0	0.0	−49716.0	0	0
mik-250-20-75-2	True	True	True	0.0	0.0	0.0	−50768.0	0	0
mik-250-20-75-3	True	True	True	0.0	0.0	0.0	−52242.0	0	0

Table 2 (*continued*)

	granular			objective				iterations	
	root	greedy	random	root	greedy	random	best known	greedy	random
mik-250-20-75-4	True	True	True	0.0	0.0	0.0	−52301.0	0	0
mik-250-20-75-5	True	True	True	0.0	0.0	0.0	−51532.0	0	0
n13-3	True	True	True	20570.0	17275.0	17325.0	13385.0	0	0
n3700	True	True	True	1831715.1	1654139.0	1426174.0	1227629.0	0	0
n3705	True	True	True	1847346.1	1597719.0	1442446.0	1225465.0	0	0
n3707	True	True	True	1788849.3	1625368.0	1354690.0	1186691.0	0	0
n3709	True	True	True	1811682.7	1676838.0	1405693.0	1207965.0	0	0
n370b	True	True	True	1911867.1	1669199.0	1442808.0	1236963.0	0	0
n5-3	True	True	True	16325.0	12725.0	14285.0	8105.0	0	0
n6-3	True	True	True	25100.0	19400.0	21550.0	15175.0	0	0
n7-3	True	True	True	22010.0	17890.0	19325.0	15426.0	0	0
n9-3	True	True	True	28825.0	21995.0	26030.0	14409.0	0	0
neos-1112782	True	True	True	22500000000000.0	22500000000000.0	1065578894123.9	571844066711.0	0	0
neos-1112787	True	True	True	20000000000000.0	20000000000000.0	593525796221.3	564772773667.0	0	0
neos-1171737	False	True	True	inf	−63.0	−34.0	−195.0	25	30
neos-1367061	True	True	True	31856051.5	31780764.1	31781461.2	31320456.3	0	0
neos-1430701	False	True	True	0.0	−42.0	−18.0	−77.0	12	25
neos-1442119	False	True	True	0.0	−98.0	−54.0	−181.0	13	27
neos-1603965	True	True	True	865504980.4	627172725.2	864287980.4	619244367.7	0	0
neos-2987310-joes	False	True	False	inf	−222862.3	inf	−607702988.3	30	30
neos-3072252-nete	False	True	True	24183360.0	23371673.0	13723328.0	11807698.0	27	27
neos-4290317-perth	False	True	True	inf	3278581.7	3912559.0	3017324.0	1	18
neos-4954672-berkel	False	True	True	18517796.0	9962581.0	8110833.0	2612710.0	2	28
neos-5076235-embley	False	False	True	inf	inf	3232.0	2362.0	25	23
neos-5079731-flyers	False	True	True	inf	3102.0	3476.0	2440.0	22	25
neos-5192052-neckar	False	True	True	−1100000.0	−9030000.0	−11280000.0	−11670000.0	2	6
neos-565672	False	True	True	3467700892969.3	649310915748.4	619139408126.0	90693.5	1	1
neos-787933	True	True	True	1764.0	1764.0	1748.0	30.0	0	0
neos-848198	True	True	True	170974.0	72169.0	66620.0	51837.0	0	0
neos-872648	True	True	True	52.6	52.6	48.8	48.6	0	0
neos-873061	True	True	True	152.4	152.4	146.0	113.7	0	0
neos-885086	False	False	True	inf	inf	−34.0	−243.0	30	30
neos-933638	True	True	True	5074445.1	136384.0	144466.7	276.0	0	0
neos-933966	True	True	True	5074472.1	132378.0	131450.8	318.0	0	0
neos17	True	True	True	0.5	0.4	0.4	0.2	0	0
neos22	False	True	True	inf	939684.4	1188643.8	779715.0	29	29
neos5	True	True	True	26.5	17.0	22.0	15.0	0	0

(*continued on next page*)

Table 2 (*continued*)

	granular			objective				iterations	
	root	greedy	random	root	greedy	random	best known	greedy	random
nexp-150-20-1-5	True	True	True	102.0	89.0	82.0	66.0	0	0
nexp-150-20-8-5	True	True	True	17880.0	4288.0	15864.0	231.0	0	0
nexp-50-20-1-1	False	True	True	inf	68.0	58.0	29.0	8	6
nexp-50-20-4-2	False	True	True	inf	340.0	739.0	71.0	5	9
ns4-pr6	True	True	True	29550.0	29452.0	29452.0	29314.0	0	0
opm2-z10-s4	True	True	True	−1489.0	−3122.0	−2983.0	−33269.0	0	0
opm2-z12-s8	True	True	True	−1678.0	−7231.0	−3536.0	−58540.0	0	0
opm2-z6-s1	True	True	True	−1076.0	−2174.0	−1510.0	−6202.0	0	0
opm2-z7-s8	True	True	True	−1654.0	−4881.0	−2503.0	−11242.0	0	0
opm2-z8-s0	True	True	True	−2220.0	−5443.0	−3066.0	−15775.0	0	0
osorio-cta	False	True	True	inf	0.0	0.0	0.0	7	7
p200x1188c	True	True	True	20962.0	20962.0	17373.0	15078.0	0	0
p500x2988	True	True	True	72594.0	72561.0	72542.0	71836.0	0	0
p500x2988c	True	True	True	17538.4	17538.0	17538.0	15215.0	0	0
p500x2988d	True	True	True	6.0	6.0	6.0	6.0	0	0
qiu	True	True	True	1805.2	347.5	511.7	−132.9	0	0
queens-30	True	True	True	0.0	−3.0	−15.0	−40.0	0	0
r50x360	True	True	True	2016.0	2016.0	2016.0	1653.0	0	0
railway_8_1_0	False	True	True	inf	482.0	486.0	400.0	5	20
ramos3	True	True	True	1087.0	558.0	630.0	192.0	0	0
ran12x21	True	True	True	5550.0	4751.0	4537.0	3664.0	0	0
ran13x13	True	True	True	4439.0	4122.0	3940.0	3252.0	0	0
ran14x18-disj-8	True	True	True	9675.7	5423.0	4758.0	3712.0	0	0
scpj4scip	True	True	True	1181054.0	57929.0	485293.0	128.0	0	0
scpk4	True	True	True	1193303.0	103775.0	496832.0	321.0	0	0
scpl4	True	True	True	2364401.0	117232.0	967349.0	262.0	0	0
scpm1	True	True	True	5990141.0	251162.0	2447966.0	557.0	0	0
scpn2	True	True	True	11936844.0	280763.0	4835563.0	516.0	0	0
set3-09	True	True	True	1985526.9	1759784.8	1759789.9	176497.1	0	0
set3-10	True	True	True	1992089.6	1756477.4	1756482.4	185179.0	0	0
set3-15	True	True	True	1925867.6	1701104.1	1701109.2	124886.0	0	0
set3-16	True	True	True	1919621.7	1701416.1	1701421.3	134040.4	0	0
set3-20	True	True	True	1916274.1	1712935.2	1712940.2	159462.6	0	0
seymour	True	True	True	744.0	489.0	555.0	423.0	0	0
seymour1	True	True	True	476.6	421.3	435.5	410.8	0	0
snip10x10-35r1budget17	False	True	True	270.0	223.9	164.8	72.3	15	21
sorrell3	True	True	True	0.0	−9.0	−1.0	−16.0	0	0
sorrell4	True	True	True	0.0	−2.0	0.0	−24.0	0	0

Table 2 (*continued*)

	granular			objective				iterations	
	root	greedy	random	root	greedy	random	best known	greedy	random
sorrell7	True	True	True	0.0	−140.0	−65.0	−196.0	0	0
sorrell8	True	True	True	0.0	−298.0	−171.0	−350.0	0	0
sp150x300d	True	True	True	70.0	69.0	70.0	69.0	0	0
stockholm	True	True	True	157.0	154.0	151.0	125.0	0	0
supportcase12	False	True	True	inf	−5235.7	−7239.6	−7559.5	19	19
supportcase39	True	True	True	−1078779.3	−1081370.0	−1082077.7	−1085069.6	0	0
supportcase42	True	True	True	10.4	9.0	9.0	7.8	0	0
ta1-UUM	True	True	True	510951307.9	289145959.4	293656954.8	7518328.2	0	0
tanglegram4	True	True	True	55202.0	24714.0	27548.0	10696.0	0	0
tanglegram6	True	True	True	8856.0	3425.0	4865.0	1224.0	0	0
thor50dday	True	True	True	204179.0	204179.0	60985.0	40417.0	0	0
toll-like	True	True	True	2204.0	729.0	1345.0	610.0	0	0
tr12-30	False	True	True	inf	196614.0	185722.9	130596.0	7	6
usAbbrv-8-25_70	False	True	True	inf	200.0	193.0	120.0	4	21
v150d30-2hopcds	True	True	True	117.0	55.0	68.0	41.0	0	0
var-smallemery-m6j6	True	True	True	5153.2	362.1	183.5	−149.4	0	0
z26	True	True	True	−123.0	−980.0	−756.0	−1187.0	0	0

Table 3
A comparison of objective values for instances where feasible rounding approaches yield best solutions after five dives.

	root node	one dive	three dives	five dives	SCIP	best known
b1c1s1	72555.00	71900.43	69071.53	61920.13	69333.52	24544.25
b2c1s1	73676.52	69221.52	68701.52	68701.52	71120.52	25687.90
bg512142	302107.00	275544.50	273390.00	273390.00	6301928.50	184202.75
buildingenergy	34196.59	34196.59	34196.59	34196.59	42652.34	33283.85
dg012142	25623489.00	14981424.74	14373382.60	14373382.60	33433439.00	2300867.00
gen-ip016	−7700.61	−9132.49	−9312.16	−9312.16	−9241.61	−9476.16
gen-ip054	10928.19	7875.28	7620.17	7207.21	7235.30	6840.97
gr4x6	332.35	236.50	218.60	218.60	219.35	202.35
gsvm2rl11	42635.34	41080.59	39792.84	39792.84	83555.95	18121.64
gsvm2rl12	34.35	34.35	34.35	34.35	50.00	22.12
gsvm2rl5	10.00	10.00	9.07	8.81	10.00	5.42
gsvm2rl9	16382.76	13611.89	13611.89	13611.89	31802.40	7438.18
haprp	4604106.31	3813762.73	3792385.43	3792385.43	4557402.61	3673280.68
mushroom-best	3613.90	2285.86	2072.90	2063.90	4208.00	0.06
neos-1112782	22001768507718.88	2067197222917.62	2062009991394.43	607470121819.71	24774435200000.00	571844066711.00
neos-1112787	20002990000000.00	1095537656401.14	1070535971510.15	590877191280.26	21786753400000.00	564772773667.00
neos-1367061	31780762.62	31780762.62	31780762.62	31780762.62	33300456.26	31320456.26
neos-1445743	−3187.00	−12951.00	−15684.00	−15684.00	−11109.00	−17905.00
neos-1445765	−2468.00	−13730.00	−13823.00	−14019.00	−11513.00	−17783.00
neos-983171	50987.00	9431.00	8747.00	8747.00	9272.00	2360.00
opm2-z10-s4	−1489.00	−18032.00	−22681.00	−22871.00	−20344.00	−33269.00
opm2-z12-s8	−1678.00	−30145.00	−37232.00	−38613.00	−38015.00	−58540.00
opm2-z6-s1	−1040.00	−3986.00	−4494.00	−4494.00	−3808.00	−6202.00
opm2-z7-s8	−1654.00	−7243.00	−7638.00	−7681.00	−5599.00	−11242.00
opm2-z8-s0	−2220.00	−10005.00	−11328.00	−11328.00	−9833.00	−15775.00
qiu	3173.59	1919.21	603.68	603.68	1805.18	−132.87
set3-09	1122029.59	1042062.59	1030350.59	938802.21	1759784.82	176497.15
set3-10	1679684.07	991124.54	961363.41	750274.15	1756477.36	185179.04
set3-15	1362877.05	1161545.90	1024364.57	869928.01	1701104.14	124886.00
set3-16	1209249.73	913569.70	877326.71	661927.58	1701416.05	134040.41
set3-20	1663579.45	1141125.67	863520.06	771482.81	1712935.20	159462.57
seymour1	437.31	425.43	421.72	421.45	438.08	410.76
sorrell7	−45.00	−153.00	−160.00	−160.00	−152.00	−196.00
sorrell8	−168.00	−324.00	−324.00	−329.00	−301.00	−350.00

[10] Ambros Gleixner, Gregor Hendel, Gerald Gamrath, Tobias Achterberg, Michael Bastubbe, Timo Berthold, Philipp Christophel, Kati Jarck, Thorsten Koch, Jeff Linderoth, Marco Lübbecke, Hans D. Mittelmann, Derya Ozyurt, Ted K. Ralphs, Domenico Salvagnin, Yuji Shinano, MIPLIB 2017: data-driven compilation of the 6th mixed-integer programming library, Math. Program. Comput. 34 (4) (2021) 361.
[11] David S. Johnson, Approximation algorithms for combinatorial problems, J. Comput. Syst. Sci. 9 (3) (1974) 256–278.
[12] Andrea Lodi, Andrea Tramontani, Performance variability in mixed-integer programming, in: Theory Driven by Influential Applications, INFORMS, 2013, pp. 1–12.
[13] Stephen Maher, Matthias Miltenberger, João Pedro Pedroso, Daniel Rehfeldt, Robert Schwarz, Felipe Serrano, PySCIPOpt: mathematical programming in Python with the SCIP optimization suite, in: Gert-Martin Greuel, Thorsten Koch, Peter Paule, Andrew Sommese (Eds.), Mathematical Software - ICMS 2016, in: Lecture Notes in Computer Science, Springer, Cham, Heidelberg, 2016, pp. 301–307.
[14] George L. Nemhauser, Laurence A. Wolsey, Integer and Combinatorial Optimization, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, New York, 2010.
[15] Christoph Neumann, Oliver Stein, Feasible rounding approaches for equality constrained mixed-integer optimization problems, Optimization (2021) 1–26.
[16] Christoph Neumann, Oliver Stein, Generating feasible points for mixed-integer convex optimization problems by inner parallel cuts, SIAM J. Optim. 31 (3) (2021) 2396–2428.
[17] Christoph Neumann, Oliver Stein, Nathan Sudermann-Merx, A feasible rounding approach for mixed-integer optimization problems, Comput. Optim. Appl. 72 (2) (2019) 309–337.
[18] Christoph Neumann, Oliver Stein, Nathan Sudermann-Merx, Bounds on the objective value of feasible roundings, Vietnam J. Math. 48 (2) (2020) 299–313.
[19] Christoph Neumann, Oliver Stein, Nathan Sudermann-Merx, Granularity in nonlinear mixed-integer optimization, J. Optim. Theory Appl. 184 (2) (2020) 433–465.
[20] Christos H. Papadimitriou, Kenneth Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Dover Books on Computer Science, Dover Publications, Newburyport, 2013.