Inner Parallel Sets in Mixed-Integer Optimization

Zur Erlangung des akademischen Grades
eines Doktors der Wirtschaftswissenschaften

Doktor rerum politicarum

von der KIT-Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte **Dissertation**
von

Christoph Neumann, M.Sc.

Tag der mündlichen Prüfung:	22. Juli 2021
Referent:	Prof. Dr. Oliver Stein
Korreferent:	Prof. Dr. Anja Fischer

Abstract

This thesis contains an extensive study of inner parallel sets in mixed-integer optimization. Inner parallel sets are a recent idea in this context and offer a possibility to relax the difficulties imposed by integrality constraints by guaranteeing feasibility of roundings of their (continuous) elements. To be able to use inner parallel sets algorithmically, various modifications, such as their enlargements and inner and outer approximations, are helpful and sometimes even necessary. Such ideas are introduced and investigated in this thesis, both theoretically as well as computationally.

From our theoretical study of inner parallel sets emerge a number of feasible rounding approaches which mainly focus on the computation of good feasible points for mixed-integer linear and nonlinear minimization problems. Good feasible points are useful in the context of solving these problems by providing tight upper bounds on the objective value. In especially difficult cases, feasible rounding approaches may also be considered as an alternative to solving a problem.

The contributions of this thesis include a thorough discussion of possibilities to enlarge inner parallel sets in the linear as well as in the nonlinear setting. Moreover, we introduce a novel cutting plane method based on inner parallel sets for mixed-integer convex minimization problems. This method, in addition to computing a good feasible point, also provides a lower bound on the objective value which is another important ingredient for solving such minimization problems. We study the possibility of dealing with equality constraints on integer variables which at first glance seem to prevent a nonempty inner parallel set. Under the occurrence of such constraints, we show that inner parallel sets can be nonempty in a reduced variable space, which allows the application of feasible rounding approaches. Finally, we investigate the behavior of inner parallel sets when integrated into search trees. Our study gives rise to a novel diving method which turns out to be a major improvement over standalone feasible rounding approaches.

We test the introduced methods on standard libraries for mixed-integer linear, convex and nonconvex minimization problems separately in several computational studies. The computational results illustrate the potential of our ideas.

Acknowledgements

The idea of using inner parallel sets for the computation of good feasible points for mixed-integer optimization problems arose within Oliver Stein's lecture "Gemischt-ganzzahlige Optimierung" in 2016. He presented inner parallel sets in the context of his recently published results on error bounds and I, at that time a student close to the end of my master's degree, wondered why we didn't use this concept for the computation of feasible points. Who would have guessed that today, 5 years later, we have coauthored six papers on that topic?

So a first word of gratitude goes to you, Oliver. Your inspiring lecture was the starting point for all ideas from this thesis. The detailed and nuanced feedback you gave me in our joint projects were an immense contribution to my learning. For me, the (by far) most joyous and exciting moments in the context of my PhD happened when I popped into your office (without prior notice) with half-baked new ideas. Thanks for bearing with me also at times when the ideas seemed clear in my head, but not so much in my expression of speech.

Oliver later admitted that he initially was not very convinced that using inner parallel sets for computing feasible points could be a fruitful idea. In fact, apparently it took some persuasion of Nathan Sudermann-Merx, who was teaching assistant at that time. So Nathan, big thanks to you for believing in this project, for putting in the efforts to convince Oliver and for our fruitful discussions!

To be fair, Oliver was right in the sense that we needed several modifications, in particular ideas around possibilities of enlarging inner parallel sets (which are introduced in this thesis), to make this concept applicable to problems from practice.

Aside from Nathan and Oliver, I acknowledge the valuable input from Benjamin Müller and Stefan Schwarze who helped to shape and to implement ideas presented in Chapter 7 of this thesis. Adrian Kruck offered valuable support in the context of implementing cutting plane ideas, and when installing open source optimization software for computational studies I relied on the help of Thorsten Rüger, Peter Kirst as well as the team around Coin-OR and SCIP. I was quite inexperienced in this endeavor which turned out to be much more difficult and frustrating than anticipated ("that's why you should use Linux!"). So thanks to everybody involved for your patience and your valuable help.

During the time of my PhD, I enjoyed many fruitful conversations with my close friends Jonas Kohler and Robert Mohr on related and unrelated research questions (ranging from inner parallel sets over optimization in machine learning to meditation and metaphysics). I'm sure that your wisdom and inspiration has found a way into my writing.

I further would like to thank my colleagues Christian Füllner, Claudio Kretz, Jana Rollmann, Marcel Sinske, Maren Beck, Martina Staiger, Michael Müller, Paula Peters, Tobias Dittrich and (again) Robert and Stefan. Our "Aktivpausen", playing basketball, spikeball and table tennis, enriched the time of my PhD enormously.

In this context, I would also like to express a word of gratitude to Clemens Puppe. You provided an excellent coffee machine that I was allowed use, even though I was not employed at your chair. More importantly though, your lectures peaked my interest in academic research already in the first semester of my bachelor's degree, so many thanks for your inspiring way of teaching.

A final word of appreciation is to you, Vivian. The possibility of doing this PhD relied on moving to Karlsruhe and on traveling to quite a few conferences and thus took up space in our relationship. You cheered me up when algorithms did not

work as well as I had hoped and when reviews were frustrating to read. I am deeply grateful for your support.

One perspective on this thesis is that it is concerned with improving optimization algorithms which, as we shall explain shortly, can be used in different contexts and may be helpful for solving various decision problems. Clearly the task of methodical research is not dictate the field of its application, yet I would like to humbly express a wish. May the knowledge accumulated in this thesis be in the service of solving pressing issues of our time in a way that benefits our planet's ecosystem and all living beings.

Contents

Abstract **iii**

Acknowledgements **v**

1 Introduction **1**

2 Basic Ideas of Inner Parallel Sets **5**
2.1 Motivation and Basic Notation 5
2.2 Geometrical Intuition of the Inner Parallel Set 6
2.3 A Functional Description for the Inner Parallel Set 8
2.3.1 An Inner Approximation 8
2.3.2 Computation of Lipschitz Constants 10

3 Enlarging Inner Parallel Sets **13**
3.1 A Preprocessing Step 13
3.2 Polyhedral Constraints 15
3.3 Nonlinear Constraints 19
3.4 The Interplay Between Enlargements and Lipschitz Constants 21
3.5 Pseudo-Granularity 22

4 Using Inner Parallel Sets for Computing Feasible Points **27**
4.1 Algorithmic Considerations 27
4.2 Computational Study in the Polyhedral Case 29
4.2.1 Granular Optimization Problems from the MIPLIB Libraries . . 30
4.2.2 Comparison with Gurobi for Granular Optimization Problems 32
4.2.3 Comparison with Gurobi for Nongranular Optimization Problems 36
4.2.4 Influence of Presolving Techniques 37
4.3 Computational Study in the Nonlinear Case 38
4.3.1 Computation of Lipschitz Constants 39
4.3.2 Pseudo-Granular Problems and Quality of the Generated Feasible Points 40
4.3.3 Effects of Different Enlargement Parameters for Box Constraints 45
4.4 Conclusions and Outlook 45

5 Inner Parallel Cuts **47**
5.1 The Special Role of Mixed-Integer Convex Optimization 47
5.2 Setting and Motivation 48
5.3 The Inner Parallel Cutting Plane Method 50
5.3.1 Statement of the Algorithm 50
5.3.2 Convergence of the Algorithm 53
5.3.3 Using Inner Parallel Cuts in Outer Approximation Based Methods 56

5.3.4 The Effect of a Nonlinear Objective Function 56
5.4 Bounds on the Objective Value 57
5.4.1 A-Posteriori Bounds 57
5.4.2 A-Priori Bounds 59
5.5 Computational Study 62
5.5.1 Instances with Feasible Points 63
5.5.2 Non-granular Consistent Instances 71
5.5.3 Inconsistent Instances 73
5.6 Conclusions 74

6 Equality Constraints and Inner Parallel Sets 75
6.1 Basic Ideas and Different Possibilities 75
6.2 Reduced Problems 78
6.3 A Reduction Technique Tailored to Feasible Rounding Approaches . . 82
6.4 Computational Results 86
6.4.1 Practical Complexity of the Elimination Procedure 86
6.4.2 Granularity in Equality Constrained Problems 88
6.5 Conclusions and Further Investigations 92

7 Inner Parallel Sets in Search Trees 93
7.1 Preliminaries 93
7.2 Fixing Variables and Inner Parallel Sets - a Geometrical Perspective . . 95
7.3 A Diving Heuristic for MILPs 98
7.3.1 A Diving Step for a Nonempty Enlarged Inner Parallel Set . . . 98
7.3.2 A Diving Step for an Empty Enlarged Inner Parallel Set 100
7.3.3 An Algorithmic Framework for Inner Parallel Set Diving 104
7.4 Computational Study 106
7.4.1 Selection of Indices 107
7.4.2 Improvement Due to IPS-Diving Steps 107
7.4.3 Possibilities of Integrating Feasible Rounding Approaches and Diving Ideas Into a Solver Framework 109
7.5 Conclusion and Outlook 111

8 Conclusion and Directions for Future Research 113

A Complementary Material for the Computational Studies 115
A.1 Computational Results for Plain Feasible Rounding Approaches 115
A.2 Computational Results for Inner Parallel Cuts 118
A.3 Computational Results for Equality Constrained Problems 120
A.4 Computational Results for Diving Methods 123

Bibliography 129

Index 135

List of Figures

2.1 Construction of the inner parallel set for a purely integer set on the left-hand side and a mixed-integer set on the right-hand side 7

3.1 Construction of the (enlarged) inner parallel set for an $MILP$ 17
3.2 Enlargement for a quadratic polynomial 20
3.3 $\widetilde{T}^{-}_{(\delta,\tau)}$ for different enlargement parameters δ and $\tau = 2.9$ 25

4.1 Number of instances for which feasible points are computed 35
4.2 Number of instances solved to the cutoff value given by FRA-SOR . . 44

5.1 Construction of the enlarged inner parallel set for an $MICP$ 49
5.2 Idea of the IPCP for the MICP from Example 5.3.1 52
5.3 Cumulative histogram of (estimates of) the optimality gap (5.18) of the feasible points 68
5.4 Biased performance profile: Time (log scale) needed for SCIP and the two Bonmin methods to compute a feasible point of similar quality as that of the IPCP 69
5.5 Maximum constraint violation $g_{\ell_k}(\breve{x}^k, \breve{y}^k)$ over iterations 71
5.6 Progress in the objective value before applying the post processing step for problems where the algorithm terminates early 71

6.1 Eliminating the equality constraint from the original feasible set (left) results in a nonempty inner parallel set in the reduced space (right) . . 77
6.2 Cumulative share of problems for which the reduction scheme can be applied successfully over time 87
6.3 Share of extracted equality constraints for instances with separable continuous variables 88

7.1 Construction of the inner parallel set $\widehat{M}^{-}$. The filled points are obtainable as roundings from $\widehat{M}^{-}$ and thus form the set R. 94
7.2 Construction of the i-ℓ-relaxed feasible set (left) and the i-ℓ-fixed inner parallel sets (right) with $i = 2$ and $\ell \in \{0, 1\}$. 96
7.3 A comparison of diving methods among each other and with the root node 108
7.4 Computational effort and quality of the generated feasible point compared to SCIPs incumbent solution in the root node 110

List of Tables

4.1 Granular presolved instances from the MIPLIB libraries 31
4.2 A comparison of the feasible rounding approaches and Gurobi with regard to time (seconds) and objective value on presolved models (I) . 33
4.3 A comparison of the feasible rounding approaches and Gurobi with regard to time (seconds) and objective value on presolved models (II) . 34
4.4 Nongranular instances for which FRA-SLOR yields a feasible point . . 36
4.5 Instances where the computation of Lipschitz constants is especially difficult 39
4.6 Comparison of FRA-SOR with three algorithms implemented in Bonmin for 39 instances from the MINLPLib 43
4.7 Instances where the computation of Lipschitz constants is possible for $\delta = 0.5$, but not for $\delta = 1 - 10^{-4}$ 45

5.1 Problems for which the IPCP computes a feasible point 67
5.2 Time (seconds) needed to solve an instance with and without RIPCs . 70
5.3 Computed lower bounds after 1800 seconds run time with and without RIPCs 70
5.4 Computational cost of the non-granularity certificate for 24 consistent instances from the MINLPLib, where $\star$ and $\star\star$ are representative for 20 and 30 variations of problems, respectively 72
5.5 Performance for difficult non-granular (inconsistent) problems 74

6.1 Instances where the the computation of $C_{R,:}$ was particularly important. With q and $q - t$ we denote the number of rows of the matrices C and $\widetilde{D}_2$, respectively 89
6.2 Comparison with Gurobi for granular problems 91

7.1 Instances where SCIP needs significantly more time to compute a feasible point with similar quality 110

A.1 A comparison of the feasible rounding approaches and Gurobi with regard to time (seconds) and objective value on unaltered models (I) . 116
A.2 A comparison of the feasible rounding approaches and Gurobi with regard to time (seconds) and objective value on unaltered models (II) . 117
A.3 Solver times (t) and lower bounds (lb) for original models (OM) and models with reversed inner parallel cuts (RICPs). NaN means that a solver error occurred 119
A.4 Run times of each reduction step for reducible problems 122
A.5 Instances with some granular node, corresponding objective values and number of feasibility diving iterations 127
A.6 A comparison of objective values for instances where feasible rounding approaches yield best incumbent solutions 128

Chapter 1

Introduction

Decision problems arise in different sectors of human endeavor including economics (e.g. portfolio optimization), science (e.g. computer science in the context of artificial intelligence), engineering (e.g. structural mechanics), logistics and many others (see [18] for a comprehensive introduction).

As it is the case for the above mentioned areas, also in general it is often possible to find a precise mathematical formulation of a decision problem. This enables resorting to and motivates the development of (efficient) algorithms for obtaining their solutions which, due to the immense complexity of such problems, would otherwise not be possible. By introducing new mathematical concepts and algorithmic ideas this thesis thus aims to contribute to the solution of such decision problems.

Independently of the area of application, whether we are able to (algorithmically) solve a decision problem is to a large extend determined by the nature of the involved mathematical functions and the structure of the feasible region. In fact, it is possible to make several notable distinctions in the theoretical and practical complexity of a decision problem.

Concerning the involved mathematical functions, important distinctions can be made between cases where all of them are known to be affine (linear) or sufficiently smooth and convex. In these cases problems are generally much easier to solve than under the occurrence of nonconvex functions. With respect to the feasible region we can state that a problem becomes more challenging if discrete (integer) variables are needed for modeling the decision problem. Such variables occur, for instance, when we want to optimize over indivisible goods or when logical connections between variables are needed.

In fact, decision problems including discrete variables *or* nonconvex functions are known to be *NP*-hard [68]. Loosely speaking, this means that finding an algorithm which deterministically solves either of these problems and whose number of iterations is bounded by a polynomial with respect to the input size of the problem would solve a Millennium Prize Problem [19]. It might even turn out that such an algorithm cannot be found at all. However, as we will elaborate a bit more fully, this should not be confused with the statement that we are not able to solve such problems of relevant sizes deterministically to global optimality.

The thesis at hand focuses on the case where the presence of discrete variables poses additional challenges for solving the optimization problem. This very general modeling paradigm allows for the presence of discrete and continuous variables and such problems are accordingly coined mixed-integer (linear/convex/nonconvex) optimization problems, dependently on the nature of the involved mathematical functions.

In the past decades, many intriguing ideas formed the basis for the impressive development of optimization software in mixed-integer optimization, dating back at least to 1958, where Gomory generalized the simplex method to a finite algorithm

for integer linear optimization [32]. In his seminal paper, problems including up to seven variables were solved on an E101 computer. Nowadays standard test libraries of mixed integer (linear) optimization problems [31] contain instances with more than 20 million variables which are quickly solved on standard commercial computers by state-of-the-art software. This immense improvement is partly driven by the rapid increase in hardware capabilities but, as demonstrated in [11], even more so by the development of new algorithmic ideas.

A notable example for the impact of algorithmic ideas from the mixed-integer community on our ability to solve real world problems is the traveling-salesman problem (TSP), which is also *NP*-hard. Here, a salesperson wants to find the shortest path that visits every city on a list exactly once and to then return to the origin city. This has proven to be an important problem which is widely applicable. A first major breakthrough for the TSP is documented in a famous article from Dantzig, Fulkerson and Johnson in 1954 [24] where they presented the first solution to a "large scale TSP" which included 49 cities. Recently, the TSP has been solved up to 49.603 cities using modern tools from integer linear optimization [23] which further highlights the difference between *NP*-hard and not practically solvable.

These examples demonstrate that the advances in mixed-integer linear optimization in recent years were enormous and have found their way into many applications. Interestingly, in contrast to their purely continuous counterpart, this is not equally true for the field of mixed-integer convex optimization. In fact, this field is only slowly starting to get more attraction and, apparently, "mixed-integer convex programming has not entered the mainstream of optimization techniques" [51]. Due to the additional difficulty introduced by nonconvexity, the research field of mixed-inter nonconvex optimization problems is rather in its infancy with regard to general purpose solvers, especially when compared to mixed-integer linear optimization.

The contribution of the thesis at hand is directed towards all these types of mixed-integer optimization problems. We thoroughly investigate the interesting and fruitful concept of *inner parallel sets*, which is a novelty in the context of computing feasible points for mixed-integer optimization problems that allows us to relax the difficulties posed by integer constraints. This, in turn, enables tackling (part of) the mixed-integer problem by resorting to methods from continuous optimization. As we shall discover, this concept is relatively widely applicable to mixed-integer linear and nonlinear optimization problems and various intriguing effects occur that open doors for further investigation.

Inner parallel sets were first used in the context of mixed-integer optimization in [71] and [72]. In these articles, this concept is applied to derive error bounds for *possibly infeasible* rounded optimal points of a continuous relaxation of the mixed-integer optimization problem. In contrast, the thesis at hand focuses on using inner parallel sets for *guaranteeing feasibility* of rounded points of certain continuous relaxations.

This work is organized as follows. Chapter 2 is introductory and familiarizes the reader with the basic notation. It introduces the concept of an inner parallel set and discusses its basic property for mixed-integer optimization problems - the feasibility for roundings of its (continuous) elements. In Chapter 3 we illustrate why enlarging this set is important and how this can be achieved practically. We further discuss issues that appear in the enlargement process for mixed-integer nonlinear optimization problems and offer remedies that partly resolve these issues.

Chapter 4 introduces feasible rounding approaches for the computation of good feasible points for mixed-integer optimization problems that naturally emerge from

our theoretical considerations. Computational results for these algorithms on standard test libraries demonstrate that inner parallel sets can indeed be used successfully for the computation of such points in the linear and the nonlinear context.

In Chapter 5 we turn towards mixed-integer convex optimization problems and develop a novel cutting-plane method that combines known ideas from Kelley's cutting plane method for convex optimization problems [44] with inner parallel sets. Our computational results indicate that an integration of this approach into existing outer approximation methods that aim at solving mixed-integer convex optimization problems could be very fruitful.

All ideas from the preceding chapters are directly applicable to mixed-integer optimization problems under the absence of equality constraints on integer variables. The reason being that, at first glance, the occurrence of such constraints seems to prohibit a nonempty inner parallel set. In Chapter 6, however, we show that there is the possibility of using inner parallel sets even under the occurrence of such equality constraints for the computation of feasible points by constructing an equivalent model in a reduced variable space. In this chapter, we also computationally evaluate the underlying reduction scheme.

Chapter 7 investigates the behavior of inner parallel sets in search trees. This covers the important case of examining theoretical properties when feasible rounding approaches are integrated into branch-and-bound methods. Our theoretical results additionally give rise to the development of a diving method for mixed-integer linear optimization problems which is based on inner parallel sets. This diving method not only makes feasible rounding approaches available to more instances from practice, but is also able to improve the quality of the generated feasible points significantly. In a computational study, we quantify this improvement and evaluate the potential of integrating feasible rounding approaches into state-of-the-art software.

Chapter 8 concludes this thesis. It highlights areas where the use of inner parallel sets is particularly promising and points to possible avenues for future research.

This thesis is based upon a number of articles. The work of Chapters 2-4 are based upon [61] and [63] published in Computational Optimization and Applications and Journal of Optimization Theory and Applications, respectively. Both are joint work with Oliver Stein and Nathan Sudermann-Merx. The articles which are the foundation for Chapter 5 and 6 are joint work with Oliver Stein and currently under review. Preprints are available at [60] and [59]. Chapter 7 is based on a working paper with Benjamin Müller, Stefan Schwarze and Oliver Stein. Further published work that discusses error bounds of the generated feasible points of the methods introduced in Chapter 4 can be found in [62] and is not part of this thesis.

Chapter 2

Basic Ideas of Inner Parallel Sets

In this chapter, we formally introduce mixed-integer optimization problems and the notion of an inner parallel set. Moreover, we study the main properties of inner parallel sets in the context of mixed-integer optimization problems. Initially, Section 2.1 introduces the basic notation of mixed-integer linear and nonlinear optimization problems which will be used throughout this thesis. In Section 2.2 we offer a geometrical perspective on inner parallel sets and Section 2.3 presents ideas which are fundamental for using inner parallel sets algorithmically.

2.1 Motivation and Basic Notation

We study mixed-integer (nonlinear) optimization problems of the form

$$MINLP: \qquad \min_{(x,y)\in\mathbb{R}^n\times\mathbb{Z}^m} f(x,y) \quad \text{s.t.} \quad g_i(x,y) \leqslant 0,\ i \in I,\ (x,y) \in D,$$

with real-valued functions f and g_i, $i \in I$, defined on $\mathbb{R}^n \times \mathbb{R}^m$, a finite index set $I = \{1, \ldots, q\}$, $q \in \mathbb{N}_0$, and a nonempty and polyhedral set

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

with a (p,n)-matrix A, a (p,m)-matrix B and $b \in \mathbb{R}^p$. In several cases, it will be illustrative to consider box constraints on the integral variables (as they usually appear in practice) separately, that is, to write

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

with $y^\ell \in (\mathbb{Z} \cup \{-\infty\})^m$, $y^u \in (\mathbb{Z} \cup \{\infty\})^m$ and $y^\ell \leqslant y^u$. At other times, writing these constraints separately would burden the notation. Whenever this is the case, we omit writing them explicitly and assume that they are modeled via the constraints $Ax + By \leqslant b$.

We denote the feasible set of the continuous (nonlinear) relaxation $\widehat{MINLP}$ of $MINLP$ by

$$\widehat{M} \ = \ \{(x,y) \in D |\ g(x,y) \leqslant 0\},$$

where g denotes the vector of functions g_i, $i \in I$. Thus, the feasible set M of $MINLP$ can be written as $M = \widehat{M} \cap (\mathbb{R}^n \times \mathbb{Z}^m)$.

This thesis distinguishes three crucial instances of $MINLP$ separately

- Mixed-integer linear optimization problems (MILPs), where $I = \varnothing$ holds.
- Mixed-integer convex optimization problems (MICPs), where all constraint functions g_i, $i \in I$, are convex on D.

- General mixed-integer nonlinear optimization problems which include nonconvex functions g_i, $i \in I$.

While methods for solving $MINLP$ have their place in this thesis (e.g. in Chapters 5 and 7), we will mostly be concerned with the related (sub)task of finding good feasible points. In fact, finding a (good) feasible point is crucial for guaranteeing (quick) convergence of methods that aim to solve $MINLP$. In any branch-and-bound algorithm, upper bounds obtained from feasible points can be used to prune a potentially large number of the nodes in the search tree. Moreover, by providing good incumbent solutions early in the search process, downside effects of a premature termination (e.g. due to time limits) are significantly reduced.

In the literature, methods that accomplish this task are coined *primal heuristics* [7]. This is the case, even though the methods themselves can be deterministic in the sense that given a particular input the method will always produce the same output and that these methods may even have a guaranteed convergence in finite time.

Even for purely integer linear optimization problems, the construction of feasible points is known to be an *NP*-hard problem [65]. Yet its importance in the quest of solving mixed-integer optimization problems has triggered the development of many search heuristics, among them the feasibility pump (cf e.g. [1, 12, 14, 26, 27]), Undercover [9], relaxation enforced neighborhood search [8], diving strategies [15] and many others (see [6, Section 6] for a survey).

Starting in this chapter but also as a general orientation in this thesis, instead of initially introducing certain methods for computing feasible points and then showing their properties (e.g. with respect to convergence), we rather focus on carefully analyzing an underlying structural property. These considerations will then allow us to understand the necessary and sufficient conditions for the applicability of the approaches emerging from this structural property and form an interesting platform for the development of further concepts and methods.

2.2 Geometrical Intuition of the Inner Parallel Set

In the following let us recall some constructions which were presented in the author's master thesis [58]. Apart from Example 2.3.7, concepts, results and proofs in this section are only slight modifications of those presented in [58], but need to be introduced to keep this thesis self-contained.

For any point $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$ we call $(\breve{x}, \breve{y})$ a rounding if

$$\breve{x} = x \quad \text{and} \quad \breve{y} \in \mathbb{Z}^m, \quad |\breve{y}_j - y_j| \leqslant \tfrac{1}{2},\ j = 1, \ldots, m,$$

hold, that is, y is rounded component-wise to a point in the integer grid $\mathbb{Z}^m$ and x remains unchanged. Note that a rounding does not have to be unique. With the sets

$$B_\infty\left(0, \tfrac{1}{2}\right) \ := \ \{y \in \mathbb{R}^m |\ \|y\|_\infty \leqslant \tfrac{1}{2}\} \quad \text{and} \quad K \ := \ \{0\} \times B_\infty\left(0, \tfrac{1}{2}\right)$$

any rounding of (x, y) satisfies

$$(\breve{x}, \breve{y}) \ \in \ ((x, y) + K) \cap (\mathbb{R}^n \times \mathbb{Z}^m)\,. \tag{2.1}$$

The central object of all approaches presented in this thesis is the *inner parallel set* of $\widehat{M}$ with respect to K,

$$\widehat{M}^- \ := \ \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m |\ (x, y) + K \subseteq \widehat{M}\}.$$

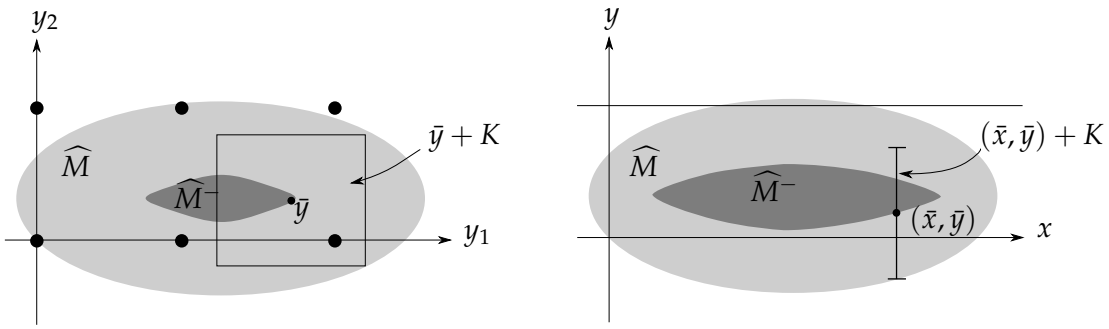


FIGURE 2.1: Construction of the inner parallel set for a purely integer set on the left-hand side and a mixed-integer set on the right-hand side

The geometrical construction of inner parallel sets is illustrated in Figure 2.1 for a purely integer feasible set on the left-hand side and a mixed-integer set on the right-hand side. Notice that, while the relaxed feasible sets $\widehat{M}$ coincide, the inner parallel set is actually larger in the mixed-integer case. This is precisely due to the fact that the box K in the mixed-integer case is a proper subset of the box K in the purely integer case.

In view of (2.1) for any $(x, y) \in \widehat{M}^-$ we have

$$(\breve{x}, \breve{y}) \in ((x, y) + K) \cap (\mathbb{R}^n \times \mathbb{Z}^m) \subseteq \widehat{M} \cap (\mathbb{R}^n \times \mathbb{Z}^m) = M.$$

and thus the following Lemma

Lemma 2.2.1. *Any rounding of any point $(x, y) \in \widehat{M}^-$ lies in M.*

Hence, as a first sufficient condition for consistency of M we obtain the following result.

Proposition 2.2.2. *If the inner parallel set $\widehat{M}^-$ is nonempty, then also M is nonempty.*

While the applicability of Proposition 2.2.2 of course hinges on a functional description of the inner parallel set $\widehat{M}^-$, a more serious drawback is that this condition may not be expected to show consistency of sets M involving binary variables, as they often appear in practice. In fact, if y_1 is a binary variable modeled as $y_1 \in \mathbb{Z} \cap [0, 1]$ then all $(x, y) \in \widehat{M}^-$ must satisfy $y_1 = 1/2$. This will often be ruled out by other constraints, the more so if many binary variables appear. Consequently, the set $\widehat{M}^-$ then is empty and the condition from Proposition 2.2.2 is useless. The resulting requirement to consider nonempty inner parallel sets gives rise to the following definition.

Definition 2.2.3. *We call the set M* granular *if the inner parallel set $\widehat{M}^-$ of $\widehat{M}$ is nonempty. Moreover, we call a problem $MINLP$ granular if its feasible set M is granular.*

In this terminology Proposition 2.2.2 states that any granular problem $MINLP$ is consistent.

For practical purposes one needs to be able to compute at least a subset T^- of $\widehat{M}^-$ explicitly. While in the linear case it turns out to be possible to find a closed-form expression of the set $\widehat{M}^-$, we shall see that this is not generally possible with nonlinear constraints. In the latter case, we initially show how to construct an inner

approximation of $\widehat{M}^-$ such that, like the set $\widehat{M}^-$, it is not restricted by integrality constraints, which is a crucial advantage compared to the set M. One can then perform a feasibility test for T^- which, if successful, also implies that M is nonempty.

It may depend on the geometry of the relaxed feasible set $\widehat{M}$ whether M is granular or not. Fortunately, it turns out that the set $\widehat{M}$ can often be replaced by a set $\widetilde{M}$ in such a way that the corresponding new inner parallel set $\widetilde{M}^-$ becomes larger than $\widehat{M}^-$, but without losing the property that any rounding of any of its elements lies in M. This will be the focus of Chapter 3 and allow us to show and exploit granularity also for many MINLPs with binary variables.

2.3 A Functional Description for the Inner Parallel Set

For an algorithmic employment of Proposition 2.2.2 we aim at obtaining a functional description of at least a subset of the inner parallel set $\widehat{M}^-$. Since with the abbreviation

$$G := \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m | \; g(x,y) \leqslant 0\}$$

we may write $\widehat{M} = D \cap G$, the inner parallel set satisfies $\widehat{M}^- = D^- \cap G^-$. From [71, Lemma 2.3] we know the closed-form expression for the inner parallel set of D,

$$D^- = \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m | \; Ax + By \leqslant b - \tfrac{1}{2}\|\beta\|_1\}, \tag{2.2}$$

where $\beta_i^\intercal$, $i = 1, \ldots, p$, denote the rows of the matrix B and, by a slight abuse of notation, $\|\beta\|_1$ stands for the vector $(\|\beta_1\|_1, \ldots, \|\beta_p\|_1)^\intercal$. This immediately yields the following Proposition, which is algorithmically applicable to MILPs.

Proposition 2.3.1. *Any rounding $(\check{x}, \check{y})$ of any point $(x,y) \in D^-$ lies in D.*

Let us now turn towards the slightly more difficult case of nonlinear constraint functions.

2.3.1 An Inner Approximation

Note that the definition of the set G^- yields

$$\begin{aligned} G^- &= \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m | \; (x,y) + K \subseteq G\} \\ &= \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m | \; g(x, y+\eta) \leqslant 0 \;\; \forall\, \eta \in B_\infty(0, \tfrac{1}{2})\}. \end{aligned}$$

While the semi-infinite constraints

$$g_i(x, y+\eta) \leqslant 0 \;\; \forall\, \eta \in B_\infty(0, \tfrac{1}{2}), \; i \in I,$$

in the above description of G^- may in general not be rewritten explicitly by finitely many smooth constraints, upper bounds for the terms $g_i(x, y+\eta)$ with $\eta \in B_\infty(0, \frac{1}{2})$ at least lead to a functional description of an inner approximation of $\widehat{M}^-$. Indeed, for any functions c_i, $i \in I$, with

$$\forall\, (x,y) \in \mathbb{R}^n \times \mathbb{R}^m, \; \eta \in B_\infty(0, \tfrac{1}{2}), \; i \in I: \quad g_i(x, y+\eta) \leqslant c_i(x,y)$$

the set $\{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m | \; c(x,y) \leqslant 0\}$ is obviously a subset of G^-.

We aim at the construction of a function c which inherits computationally attractive features of g. The following approach will accomplish this by defining c as the

sum of g with a nonnegative constant, which stems from global Lipschitz conditions with respect to y uniformly in x for the functions g_i, $i \in I$, on the set D. This distinction between the roles of x and y is caused by the definition of inner parallel sets, whose geometric construction only depends on the discrete variable y.

To be more specific, for any $x \in \mathbb{R}^n$ we define the set

$$D(x) \;:=\; \{y \in \mathbb{R}^m |\; (x,y) \in D\}$$

and denote by

$$\mathrm{pr}_x D \;:=\; \{x \in \mathbb{R}^n |\; D(x) \neq \varnothing\}$$

the parallel projection of D to the "x-space" $\mathbb{R}^n$. Then the functions g_i, $i \in I$, are assumed to satisfy Lipschitz conditions with respect to the ℓ_∞-norm on the fibers $\{x\} \times D(x)$, independently of the choice of $x \in \mathrm{pr}_x D$.

Assumption 2.3.2. *For all $i \in I$ there exists some $L^i_\infty \geqslant 0$ such that for all $x \in \mathrm{pr}_x D$ and all $y^1, y^2 \in D(x)$ we have*

$$|g_i(x,y^1) - g_i(x,y^2)| \;\leqslant\; L^i_\infty \|(x,y^1) - (x,y^2)\|_\infty \;=\; L^i_\infty \|y^1 - y^2\|_\infty.$$

We allow for vanishing Lipschitz constants to cover trivial cases. Some problem classes for which the Lipschitz constants from Assumption 2.3.2 can be calculated will be discussed in Section 2.3.2.

Under Assumption 2.3.2, and with L_∞ denoting the vector of Lipschitz constants L^i_∞, $i \in I$, we may define the set

$$T^- := \left\{(x,y) \in D^- |\; g(x,y) + \tfrac{1}{2} L_\infty \leqslant 0\right\}.$$

We next repeat the proof that T^- is indeed an inner approximation of $\widehat{M}^-$. A precursor of this result was also used to show [72, Lemma 2.4].

Lemma 2.3.3. *Under Assumption 2.3.2 we have $T^- \subseteq \widehat{M}^-$.*

Proof. In the case $T^- = \varnothing$ the assertion trivially holds. Otherwise, let $(x,y) \in T^-$. We have to show that

$$(x, y+\eta) \in \widehat{M} = \{(x,y) \in D |\; g_i(x,y) \leqslant 0,\; i \in I\}$$

holds for any $\eta \in B_\infty\left(0, \frac{1}{2}\right)$.

First, $(x,y) \in D^-$ and $(0,\eta) \in K$ imply $(x, y+\eta) = (x,y) + (0,\eta) \in D$. This also yields $x \in \mathrm{pr}_x D$ and $y + \eta \in D(x)$. As also y lies in $D(x)$, Assumption 2.3.2 implies for any $i \in I$

$$g_i(x, y+\eta) - g_i(x,y) \;\leqslant\; L^i_\infty \|\eta\|_\infty \;\leqslant\; \tfrac{1}{2} L^i_\infty.$$

From the definition of T^- we thus obtain

$$g_i(x, y+\eta) \;\leqslant\; g_i(x,y) + \tfrac{1}{2} L^i_\infty \;\leqslant\; 0,$$

so that altogether we have shown $(x,y) \in \widehat{M}^-$. □

Lemma 2.3.3 implies the following result.

Theorem 2.3.4. *Under Assumption 2.3.2 any rounding $(\check{x}, \check{y})$ of any point $(x,y) \in T^-$ lies in M. In particular, if T^- is nonempty, then MINLP is granular and, thus, consistent.*

2.3.2 Computation of Lipschitz Constants

Let us identify some problem classes which are suitable for the computation of the necessary Lipschitz constants in Assumption 2.3.2. To ensure solvability of auxiliary problems, we assume that the set D is bounded (and thus compact). From the mean value theorem, the Hölder inequality and the Weierstrass theorem it is well-known that for fixed $x \in \mathrm{pr}_x D$ with a nonempty and compact set $D(x)$ and a continuously differentiable (in y) function $g_i(x,\cdot)$, $i \in I$, the value

$$L_\infty^i(x) = \max_{y \in D(x)} \|\nabla_y g_i(x,y)\|_1 \tag{2.3}$$

is a Lipschitz constant for $g_i(x,\cdot)$ on $D(x)$ with respect to the ℓ_∞-norm.

We remark that the assertions of Theorem 2.3.4 also hold if the set T^- is replaced by $\{(x,y) \in D^- | \, g(x,y) + L_\infty(x)/2 \leqslant 0\}$. However, the functional dependence of L_∞ on x may in general not be expected to possess smoothness and convexity properties which are computationally useful. For this reason, Assumption 2.3.2 aims at Lipschitz conditions which hold uniformly in x.

To achieve this uniformity, let us define the "worst case" Lipschitz constants among those in (2.3) with respect to $x \in \mathrm{pr}_x D$,

$$L_\infty^i := \sup_{x \in \mathrm{pr}_x D} L_\infty^i(x) = \sup_{x \in \mathrm{pr}_x D} \max_{y \in D(x)} \|\nabla_y g_i(x,y)\|_1, \; i \in I.$$

As we assume D to be a polytope, this actually yields

$$L_\infty^i = \max_{(x,y) \in D} \|\nabla_y g_i(x,y)\|_1, \; i \in I. \tag{2.4}$$

Note that L_∞^i is preferable over the Lipschitz constant

$$\widetilde{L}_\infty^i := \max_{(x,y) \in D} \|\nabla g_i(x,y)\|_1 = \max_{(x,y) \in D} (\|\nabla_x g_i(x,y)\|_1 + \|\nabla_y g_i(x,y)\|_1)$$

for g_i on D, since it promotes a larger set T^-. This shows that Assumption 2.3.2 is weaker than a general Lipschitz condition and illustrates our main motivation to require Lipschitz conditions only on the the fibers $\{x\} \times D(x)$.

Example 2.3.5. *If for some $i \in I$ the entries of the gradient $\nabla_y g_i$ are factorable functions, then techniques from interval arithmetic may be employed to compute L_∞^i as a guaranteed upper bound for $\max_{(x,y) \in D} \|\nabla_y g_i(x,y)\|_1$ (cf., e.g., [35, 57]). Again, since smaller Lipschitz constants lead to larger sets T^-, good upper bounds are beneficial for the consistency of T^-.*

Example 2.3.6. *If for some $i \in I$ we have $g_i(x,y) = F_i(x) + \beta_i^\mathsf{T} y$, then* (2.4) *boils down to $L_\infty^i = \|\beta_i\|_1$. This actually explains the inclusion "$\supseteq$" in* (2.2).

The next example will be important for enlargement considerations in Chapter 3, as well as for the computational study in Chapter 4 and corrects the author's error in [58]. In the following, e will denote the all-ones vector of suitable dimension.

Example 2.3.7. *If for some $i \in I$ we have*

$$g_i(x,y) = F(x) + \tfrac{1}{2} y^\mathsf{T} Q_y y + y^\mathsf{T} Q_x x + \beta^\mathsf{T} y + q,$$

with an (m,m)-matrix Q_y, an (m,n)-matrix Q_x and $q \in \mathbb{R}$, then

$$L_\infty^i = \max_{(x,y) \in D} \|Q_y y + Q_x x + \beta\|_1$$

may be computed by the vertex theorem of convex maximization [67, Corollary 32.3.4] as

$$L_\infty^i \;=\; \max_{(x,y)\in \mathrm{vert}D} \|Q_y y + Q_x x + \beta\|_1,$$

where vert$D$ *denotes the vertex set of D. Alternatively we may obtain L_∞^i as the optimal value of the linear program with complementarity constraints [29]*

$$\begin{aligned} LPCC: \qquad \max_{x,y,u,v} e^\intercal(u+v) \quad s.t. \quad & Q_y y + Q_x x + \beta = u - v,\ \ u^\intercal v = 0, \\ & u, v \geqslant 0,\ \ (x,y) \in D. \end{aligned}$$

As a third possibility, L_∞^i can be computed as the optimal value of a mixed-integer linear optimization problem. In fact, modeling the complementarity constraint $u^\intercal v = 0$ in the above LPCC by a big-M reformulation results in the problem

$$\begin{aligned} MILP: \qquad \max_{x,y,u,v,z} e^\intercal(u+v) \quad s.t. \quad & Q_y y + Q_x x + \beta = u - v, \\ & u \leqslant \mathrm{diag}(M_u) z, \\ & v \leqslant \mathrm{diag}(M_v)(e - z), \\ & u, v \geqslant 0,\ (x,y) \in D,\ z \in \mathbb{B}^m, \end{aligned}$$

with $M_u, M_v \in \mathbb{R}^m$ large enough, where $\mathrm{diag}(M_u)$ *and* $\mathrm{diag}(M_v)$ *denote the diagonal (m,m)-matrices with entries M_u and M_v (and, as announced, e stands for the m-dimensional all-ones vector).*

Notice that, in order to obtain a tight LP relaxation, we propose not one single big-M constant, but different big-M's for each variable u_i, v_i, $i = 1, \ldots, m$. We stress that good values for the entries of M_u and M_v can be computed explicitly from the problem data. Indeed, with $(q_x)_k^\intercal$ and $(q_y)_k^\intercal$ denoting row k of the Matrix Q_x and Q_y, respectively, we may set the k'th entries of M_u and M_v to

$$(M_u)_k = \max_{(x,y)\in D} (q_x)_k^\intercal x + (q_y)_k^\intercal y + \beta_k \;\; \text{and} \;\; (M_v)_k = -\left(\min_{(x,y)\in D} (q_x)_k^\intercal x + (q_y)_k^\intercal y + \beta_k \right),$$

which comes at the cost of solving $2m$ LPs. The validity of theses bounds for u and v immediately follows from the equality constraints $Q_y y + Q_x x + \beta = u - v$ together with the (remodeled) complementarity constraints.

As our numerical study in Section 4.3 shall reveal, separable quadratic constraints g_i constitute a relevant special case of this setting. Since they satisfy $Q_x = 0$, and under the explicit knowledge of box constraints for y, that is $y \in [y^\ell, y^u]$ with $y^\ell, y^u \in \mathbb{R}^m$, we may compute valid (albeit possibly coarser) values for $(M_u)_k$ and $(M_v)_k$ as

$$\begin{aligned} (M_u)_k &= \max_{y\in[y^\ell,y^u]} \sum_{j=1}^m (q_y)_{kj} y_j + \beta_k \\ &= \sum_{j:(q_y)_{kj}>0} (q_y)_{kj} y_j^u + \sum_{j:(q_y)_{kj}<0} (q_y)_{kj} y_j^\ell + \beta_k\,, \qquad (2.5) \\ (M_v)_k &= -\left(\min_{y\in[y^\ell,y^u]} \sum_{j=1}^m (q_y)_{kj} y_j + \beta_k \right) \\ &= -\left(\sum_{j:(q_y)_{kj}>0} (q_y)_{kj} y_j^\ell + \sum_{j:(q_y)_{kj}<0} (q_y)_{kj} y_j^u + \beta_k \right), \qquad (2.6) \end{aligned}$$

where $(q_y)_{kj}$ denotes the entry at row k and column j of Q_y. Note that these bounds are valid due to our previous construction and $\mathrm{pr}_y D \subseteq [y^\ell, y^u]$. In our computational study, we shall indeed use (2.5) *and* (2.6) *to quickly compute M_u and M_v.*

We remark that neither of the three above auxiliary problems for the computation of L^i_∞ is efficiently solvable in the sense that we may obtain an optimal value in polynomial time. However, our computational study will reveal that the auxiliary $MILP$ is quickly solvable for many practical applications.

Chapter 3

Enlarging Inner Parallel Sets

Any relaxation of the constraints of the (inner approximation of the) inner parallel set under which the assertions of Proposition 2.3.1 (Theorem 2.3.4) still hold is beneficial, since it increases chances for proving granularity of M and using inner parallel sets for the computation of feasible points. In this chapter we discuss the possibility of such enlargements.

We begin our investigation with a geometrical perspective on enlargements in Section 3.1. Subsequently, we study concrete enlargement ideas for polyhedral constraints in Section 3.2, and for nonlinear constraints in Section 3.3. Section 3.4 draws these results together and discusses an unwanted side effect, which can partly be resolved by a modification of the geometrically intuitive granularity concept to the algorithmically more attractive notion of pseudo-granularity. This is the content of Section 3.5.

Note that for MILPs we have $I = \varnothing$ and thus

$$\widehat{M}^- = \hat{D}^- = T^-,$$

which we will not always make explicit in the following considerations but, for ease of notation and to cover the possibility of nonlinear constraints, often simply refer to as T^-.

3.1 A Preprocessing Step

The main idea for the construction of such enlargements of T^- is a preprocessing step for the functional description of $MINLP$. It first enlarges the relaxed feasible set $\widehat{M}$ of M to some set $\widetilde{M} \supseteq \widehat{M}$ for which the feasible set M of $MINLP$ can still be written as

$$M \;=\; \widetilde{M} \cap (\mathbb{R}^n \times \mathbb{Z}^m)\,. \tag{3.1}$$

Then we call the inner parallel set

$$\widetilde{M}^- \;=\; \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m |\; (x,y) + K \subseteq \widetilde{M}\}$$

of $\widetilde{M}$ an *enlarged inner parallel set* of $\widehat{M}$ since the relation $\widehat{M} \subseteq \widetilde{M}$ implies $\widehat{M}^- \subseteq \widetilde{M}^-$. Depending on the functional description of $\widetilde{M}$ and, in particular, the appearing Lipschitz constants, the inner approximation $\widetilde{T}^-$ of $\widetilde{M}^-$ may then be larger than T^- (cf. Section 3.4 for a discussion of the appearing issues).

In general, the approach for the construction of $\widetilde{M}$ involves the replacement of the set D by a different set $\widetilde{D}$ as well as the replacement of the function g by a different function $\widetilde{g}$. The possibilities for the appropriate constructions of $\widetilde{D}$ and $\widetilde{g}$ range from small perturbations of D and g to the choice of structurally different objects.

When applying branch and cut ideas, one is usually rather interested in a *tight* formulation of a problem $MINLP$, that is, one wants to find a *small* set $\widetilde{M}$ for which (3.1) holds. We, on the other hand, are interested in a *non-tight* formulation and hence a *large* set $\widetilde{M}$, in order to promote the consistency of its inner parallel set $\widetilde{M}^-$.

Remark 3.1.1. *The previous discussion shows that granularity of M and thus of $MINLP$ as defined in Definition 2.2.3 is dependent on the description of the relaxed feasible set. An interesting question that arises from this dependency is whether a problem is* granularity representable*, that is, if* some *set $\widetilde{M}$ fulfilling Equation* (3.1) *with a nonempty inner parallel set $\widetilde{M}^-$ exists.*

Our main focus, however, will be developing conditions which are algorithmically testable for specific initial representations $\widehat{M}$ and we hence link the concept of granularity to particular enlargement ideas that depend on the initial functional description of $\widehat{M}$. Therefore, our notion of granularity is in a sense context-dependent. Finding a granular problem in this context-dependent sense implies its granular representability, yet the reverse implication is generally not true and non-granular problems might well be granularity representable. This connection will become more apparent in Example 3.2.5.

In fact, let us consider enlargements of $\widehat{M}$ resulting from constant additive relaxations of its constraints $Ax + By \leqslant b$ and $g(x, y) \leqslant 0$, that is, we consider the relaxed constraints $Ax + By \leqslant b + \sigma$ and $g(x, y) \leqslant \tau$ with appropriately chosen vectors $\sigma, \tau \geqslant 0$. This approach maintains algorithmically attractive properties like the polyhedrality of D and differentiability or convexity of the functions g_i, $i \in I$. We set

$$D_\sigma \;:=\; \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m |\; Ax + By \leqslant b + \sigma\}$$

as well as

$$G_\tau \;:=\; \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m |\; g(x, y) \leqslant \tau\}.$$

Clearly, for each $\rho := (\sigma, \tau) \geqslant 0$ the set

$$\widehat{M}_\rho := D_\sigma \cap G_\tau \tag{3.2}$$

satisfies $\widehat{M} \subseteq \widehat{M}_\rho$. Let us denote the appropriate choices of ρ for (3.1) by

$$R \;:=\; \{\rho \in \mathbb{R}^p \times \mathbb{R}^q |\; \rho \geqslant 0,\; M = \widehat{M}_\rho \cap (\mathbb{R}^n \times \mathbb{Z}^m)\}. \tag{3.3}$$

Then, as in the derivation of Proposition 2.2.2, for each $\rho \in R$ any rounding of any element of $\widehat{M}_\rho^-$ lies in M. Furthermore, we have $\widehat{M}^- \subseteq \widehat{M}_\rho^-$, so that $\widehat{M}_\rho^-$ is more likely to be nonempty than $\widehat{M}^-$. In fact, after preprocessing $\widehat{M}$ to $\widehat{M}_\rho$ for some $\rho \in R$, according to Definition 2.2.3 the set M and the problem $MINLP$ are granular and, thus, consistent if the enlarged inner parallel set $\widehat{M}_\rho^-$ is nonempty. Note that due to (3.2) we may write $\widehat{M}_\rho^- = D_\sigma^- \cap G_\tau^-$ with

$$D_\sigma^- \;=\; \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m |\; Ax + By \leqslant b + \sigma - \tfrac{1}{2}\|\beta\|_1\}. \tag{3.4}$$

The following example illustrates how a granularity proof for nonlinear binary problems can benefit from this construction.

Example 3.1.2. *Consider the nonlinear problem*

$$\min_{(x,y)\in\mathbb{R}^n\times\mathbb{B}^m} f(x, y) \quad s.t. \quad g(x, y) \;\leqslant\; 0$$

with binary variables $y_j \in \mathbb{B} = \{0,1\}$, $j = 1,\dots,m$. By rewriting the set $\mathbb{B}^m$ as $\{y \in \mathbb{Z}^m | \, 0 \leqslant y \leqslant e\}$, we obtain a problem of type MINLP with

$$D := \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m | \; -y \leqslant 0, \; y \leqslant e\}.$$

This results in the continuously relaxed feasible set $\widehat{M} = D \cap G$ of M with the inner parallel set

$$\begin{aligned} \widehat{M}^- &= D^- \cap G^- = \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m | \; -y \leqslant -\tfrac{1}{2}e, \; y \leqslant \tfrac{1}{2}e\} \cap G^- \\ &= (\mathbb{R}^n \times \{\tfrac{1}{2}e\}) \cap G^- \end{aligned}$$

which is likely to be inconsistent. However, for any $0 \leqslant \sigma^\ell, \sigma^u < e$ we may also write

$$\mathbb{B}^m = \{y \in \mathbb{Z}^m | \; -\sigma^\ell \leqslant y \leqslant e + \sigma^u\}$$

and define

$$D_\sigma := \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m | \; -y \leqslant \sigma^\ell, \; y \leqslant e + \sigma^u\}$$

with $\sigma = (\sigma^\ell, \sigma^u)$. With $\rho = (\sigma, 0)$ this leads to the relaxed feasible set $\widehat{M}_\rho = D_\sigma \cap G$ satisfying $M = \widehat{M}_\rho \cap \mathbb{Z}^m$, so that $\rho = (\sigma, 0)$ lies in R for any $0 \leqslant \sigma^\ell, \sigma^u < e$. Due to

$$\begin{aligned} \widehat{M}^- \subseteq \widehat{M}_\rho^- &= \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m | \; -y \leqslant \sigma^\ell - \tfrac{1}{2}e, \; y \leqslant \tfrac{1}{2}e + \sigma^u\} \cap G^- \\ &= (\mathbb{R}^n \times [\tfrac{1}{2}e - \sigma^\ell, \tfrac{1}{2}e + \sigma^u]) \cap G^- \end{aligned}$$

the chance for consistency of the enlarged inner parallel set $\widehat{M}_\rho^-$ is larger than this chance for $\widehat{M}^-$, where entries of σ^ℓ and σ^u close to one are particularly beneficial.

This shows that even nonlinear binary problems may be granular in the sense of Definition 2.2.3.

3.2 Polyhedral Constraints

Let us extend Example 3.1.2 to general polyhedral constraints. The focus of this section will be on one specific enlargement procedure which is ready to implement algorithmically. In fact, as we shall see next, enlarging D is efficiently possible for any constraint i which is only posed on integral variables and possesses integer coefficients, that is, for $\alpha_i = 0$ and $\beta_i \in \mathbb{Z}^m \backslash \{0\}$ where the vectors α_i^T, $i = 1,\dots,p$, denote the rows of the matrix A. A proof for the following result may be found, e.g., in [22, Corollary 1.9].

Lemma 3.2.1. *For $i \in \{1,\dots,p\}$ let ω_i denote the greatest common divisor of the entries of $\beta_i \in \mathbb{Z}^m \backslash \{0\}$. Then all values of $\beta_i^\mathsf{T} y$ with $y \in \mathbb{Z}^m$ are multiples of ω_i.*

In the following, for $a_i \in \mathbb{R}$ and $\omega_i \in \mathbb{N}$, let

$$\lfloor a_i \rfloor_{\omega_i} := \max\{z \in \omega_i \mathbb{Z} | \; z \leqslant a_i\}, \;\text{ and }\; \lfloor a_i \rfloor_0 := a_i,$$

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

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

Lemma 3.2.1 suggests that for an inequality constraint i with $\alpha_i = 0$, we may relax the components b_i of the right-hand side vector arbitrarily close to $\lfloor b_i \rfloor_{\omega_i} + \omega_i$ without

admitting any additional solution. Setting ω_i to zero for any $i \in \{1, \dots, p\}$ for which $\alpha_i \neq 0$ or $\beta_i \notin \mathbb{Z}^m$ holds, and with $\omega := (\omega_1, \dots, \omega_p)^\intercal \in \mathbb{N}_0^p$, we define the set

$$D_{\sigma(\delta)} := \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m \mid Ax + By \leqslant \lfloor b \rfloor_\omega + \delta\omega,\ y^\ell - \delta e \leqslant y \leqslant y^u + \delta e\} \tag{3.5}$$

with $\delta \in (0, 1)$. For future reference we will abbreviate $D_{\sigma(\delta)}$ by D_δ.

The previous considerations give rise to the following result.

Lemma 3.2.2. *For any $\delta \in (0, 1)$, we have*

$$M = D_\delta \cap G \cap (\mathbb{R}^m \times \mathbb{Z}^m)\,.$$

Proof. Take any row of the constraint matrix with $\alpha_i = 0$ and $\beta_i \in \mathbb{Z}^m$. From Lemma 3.2.1, it follows that $\beta_i^\intercal y = \lfloor b_i \rfloor_{\omega_i} + \delta\omega_i$ has no integral solution for any $\delta \in (0, 1)$. Thus we may rewrite any inequality constraint $\beta_i^\intercal y \leqslant b_i$ as $\beta_i^\intercal y \leqslant \lfloor b_i \rfloor_{\omega_i} + \delta\omega_i$ without admitting any additional solution. Moreover, the box constraints on the integral variables in (3.5) correspond to the special case of Lemma 3.2.1, where ω is the m-dimensional vector of ones and the right-hand side vectors correspond to $-y^\ell = \lfloor -y^\ell \rfloor_\omega$ and $y^u = \lfloor y^u \rfloor_\omega$, respectively. For $\alpha_i \neq 0$ or $\beta_i \notin \mathbb{Z}^m$, an inequality constraint i remains unchanged so that overall the assertion is shown. □

Note that in the special case $\omega = 0$, the enlarged inner parallel set collapses to the inner parallel set. Otherwise, a restriction i is relaxed only if

$$\delta \geqslant \frac{b_i - \lfloor b_i \rfloor_{\omega_i}}{\omega_i}$$

holds. As we want to ensure $D \subseteq D_\delta$ (and thus $D^- \subseteq D_\delta^-$), even if ω is not the zero vector, we need to add a lower bound

$$\delta_e := \begin{cases} \max\left\{ \frac{b_i - \lfloor b_i \rfloor_{\omega_i}}{\omega_i} \,\middle|\, i = 1, \dots, p, \omega_i \neq 0 \right\}, & \omega \neq 0, \\ 0 & \text{else} \end{cases} \tag{3.6}$$

on δ, that is, $\delta \in [\delta_e, 1)$. Note that $\delta_e < 1$ always holds and that the interval $[\delta_e, 1)$ is hence never empty. By using this construction, we indeed always obtain an *enlarged* inner parallel set if $\omega \neq 0$ holds.

By Equation (2.2) the closed-form expression of the enlarged inner parallel set of the polyhedral constraints may be written as

$$\begin{aligned} D_\delta^- = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m \mid\ & Ax + By \leqslant \lfloor b \rfloor_\omega + \delta\omega - \tfrac{1}{2} \|\beta\|_1\,, \\ & y^\ell + (\tfrac{1}{2} - \delta)e \leqslant y \leqslant y^u + (\delta - \tfrac{1}{2})e\}. \end{aligned} \tag{3.7}$$

The following main result in this section immediately follows from Lemmata 2.2.1 and 3.2.2.

Theorem 3.2.3. *For mixed-integer linear optimization problems ($I = \varnothing$), for any $\delta \in [\delta_e, 1)$, any rounding of any point from D_δ^- lies in M.*

Figure 3.1 depicts the construction of the inner parallel set $\widehat{M^-}$ (left) and the enlarged inner parallel set $\widehat{M_\delta^-}$ (right) for a two dimensional purely integer linear optimization problem. Note that from this example it follows that we do not nec-

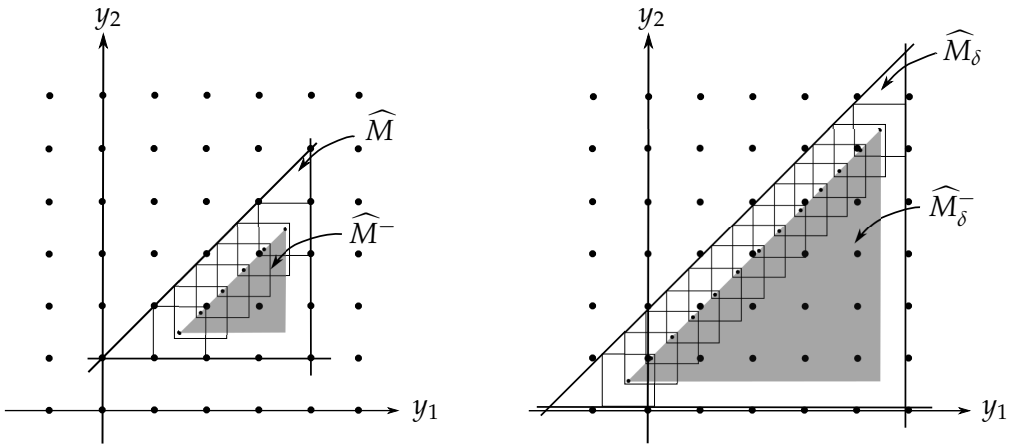


FIGURE 3.1: Construction of the (enlarged) inner parallel set for an $MILP$

essarily have $\widehat{M}_\delta^- \subseteq \widehat{M}$, in particular under the occurrence of box constraints. However, the consistency of $\widehat{M}_\delta^-$ still guarantees consistency of $\widehat{M}$: if $\widehat{M}_\delta^- \neq \emptyset$ holds, then we may choose some $(x, y) \in \widehat{M}_\delta^-$ and for the corresponding rounding we have $(\check{x}, \check{y}) \in M \subseteq \widehat{M}$.

The next example emphasizes the importance of enlarging the inner parallel set.

Example 3.2.4. *Consider the following (binary) knapsack problem, where for the purpose of illustration we are restricted to one item only:*

$$KP: \quad \max_{y \in \mathbb{Z}^m} d^\intercal y \quad s.t. \quad \sum_{i=1}^{m} y_i \leqslant 1, \quad 0 \leqslant y \leqslant e.$$

At first glance, the restriction $\sum_{i=1}^m y_i \leqslant 1$ appears to inhibit granularity, as $\|\beta\|_1 = m$ increases linearly with dimensionality m and hence the enlarged inner parallel set

$$\widehat{M}_\delta^- = \{y \in \mathbb{R}^m | \sum_{i=1}^{m} y_i \leqslant 1 + \delta - \tfrac{m}{2},\ y \in [\tfrac{1}{2} - \delta, \tfrac{1}{2} + \delta]^m\}$$

appears to be shrinking (eventually being empty) for increasing dimensionality.

However, note that for the enlarged inner parallel set, all m components of y may also take a negative value up to $(\frac{1}{2} - \delta)$ (with δ close to 1). Therefore, the point $(\frac{1}{2} - \delta)e$ lies in $\widehat{M}_\delta^-$ if and only if

$$m(\frac{1}{2} - \delta) \leqslant 1 + \delta - \frac{m}{2}$$

holds, which is the case for any $\delta \geqslant \frac{m-1}{m+1}$. Consequently, the problem KP is granular, independently of the dimensionality m.

In fact, for a sufficiently large value $\delta < 1$, we may even construct a point $y^k \in \widehat{M}_\delta^-$ such that $\check{y}^k$ is the optimal solution of KP as follows. Assume the k-th item is optimal for KP. Then the unique rounding $\check{y}^k$ of the vector $y^k \in \mathbb{R}^m$, which takes the value 1 at position k and $(\frac{1}{2} - \delta)$ elsewhere, is an optimal point. Moreover, we have $y^k \in \widehat{M}_\delta^-$ if and only if

$$\sum_{i=1}^{m} y_i^k = (m-1)(\frac{1}{2} - \delta) + 1 \leqslant 1 + \delta - \frac{m}{2},$$

which is the case exactly for $\delta \in [1 - \frac{1}{2m}, 1)$. Hence, we may construct y^k independently of the dimensionality m.

This shows that even feasible sets with only a relatively small number of discrete points can be granular and that we might actually be able to compute good *points by rounding points of $\widehat{M}_\delta^-$.*

Note that the arguments for the construction of an optimal point of Example 3.2.4 generalize to an arbitrary size of the knapsack.

Example 3.2.5. *To demonstrate the general flexibility of enlargements and that granularity is actually dependent on modeling techniques, let us again consider the problem KP from Example 3.2.4, where we denote its feasible set by*

$$K := \{y \in \mathbb{Z}^m | \sum_{i=1}^{m} y_i \leqslant 1,\ y \in [0,1]^m\},$$

and that of its equivalent set cover formulation by

$$K^{sc} := \{y \in \mathbb{Z}^m | \ y_i + y_j \leqslant 1,\ i = 1,\dots,m,\ j = 1,\dots,m,\ i \neq j,\ y \in [0,1]^m\}.$$

Although these two sets are identical, a comparison of their enlarged inner parallel sets

$$(\widehat{K}^{sc})_\delta^- = \{y \in \mathbb{R}^m | \ y_i + y_j \leqslant \delta, i = 1,\dots,n,\ j = 1,\dots,n,\ i \neq j,\ y \in [\tfrac{1}{2} - \delta, \tfrac{1}{2} + \delta]^m\},$$

and

$$\widehat{K}_\delta^- = \{y \in \mathbb{R}^m | \sum_{i=1}^{m} y_i \leqslant 1 + \delta - \tfrac{m}{2},\ y \in [\tfrac{1}{2} - \delta, \tfrac{1}{2} + \delta]^m\}$$

reveals that, for $m > 2$, the set cover formulation is better suited for our purpose due to $\widehat{K}_\delta^- \subseteq (\widehat{K}^{sc})_\delta^-$. In fact, for any $y \in \widehat{K}_\delta^-$, we have $y_i \geqslant \frac{1}{2} - \delta$ and hence for all $i \neq j$

$$1 + \delta - \frac{m}{2} \ \geqslant \ \sum_{i=1}^{m} y_i \ \geqslant \ y_i + y_j + (m-2)(\frac{1}{2} - \delta).$$

Therefore, for all $i \neq j$ and $\delta \in (0,1)$, any $y \in \widehat{K}_\delta^-$ satisfies

$$y_i + y_j \leqslant (m-2)(\delta - 1) + \delta < \delta,$$

and, thus, $y \in (\widehat{K}^{sc})_\delta^-$. Moreover, for $\bar{y} := (\delta, 0, \dots, 0)^\intercal$, we have $\bar{y} \in (\widehat{K}^{sc})_\delta^-$, but $\bar{y} \notin \widehat{K}_\delta^-$ and hence $\widehat{K}_\delta^- \subsetneq (\widehat{K}^{sc})_\delta^-$ holds.

This indicates that partitioning dense constraints into several sparse constraints may generally promote granularity.

The next example shows that presolving techniques, as commonly applied by MILP-Solvers, can also influence granularity.

Example 3.2.6. *Consider the feasible set of a purely integer linear optimization problem*

$$M = \{y \in \mathbb{Z}^2 | \ y_1 + y_2 \leqslant \tfrac{1}{2},\ y_1 + y_2 \geqslant 0, y \geqslant 0\}.$$

The constraint $y_1 + y_2 \geqslant 0$ is redundant and can be removed in a presolving step which yields the enlarged inner parallel set

$$\widehat{M}_\delta^- = \{y_1 + y_2 \leqslant \delta - 1,\ y \geqslant (\tfrac{1}{2} - \delta)e\},$$

which contains the point $(-\frac{1}{6}, -\frac{1}{6})^\intercal$ *for* $\delta \geqslant \frac{2}{3}$ *and is hence nonempty for these values of* δ. *Yet, keeping the redundant constraint results in the constraints*

$$\begin{aligned} y_1 + y_2 &\leqslant \delta - 1 \\ -y_1 - y_2 &\leqslant \delta - 1 \end{aligned}$$

which are not satisfiable for any $\delta < 1$.

3.3 Nonlinear Constraints

The next example shows that enlargement ideas may also be applied to the setting of nonlinear inequality constraints.

Example 3.3.1. *For* $n = 0$ *let us consider the set* $M = \{y \in D \cap \mathbb{Z}^m \mid g(y) \leqslant 0\}$ *with a nonempty polyhedron* $D \subseteq \mathbb{R}^m$ *and the function* $g(y) = \pi(y) - p_0$. *We assume that* π *is a real-valued multivariate polynomial*

$$\pi(y) = \sum_{\alpha \in A} p_\alpha y^\alpha,$$

where $A \subseteq \mathbb{N}_0^m \backslash \{0\}$ *is a finite set,* $\alpha \in \mathbb{N}_0^m$ *denotes a multi-index, and* y^α *stands for the product* $y_1^{\alpha_1} \cdots y_m^{\alpha_m}$. *Furthermore, for the coefficients we assume* $p_0 \in \mathbb{R}$ *and* $p_\alpha \in \mathbb{Z}$, $\alpha \in A$. *Then* π *does not only map from* $\mathbb{Z}^m$ *to* $\mathbb{Z}$, *but the values in* $\pi(\mathbb{Z}^m)$ *may even form a proper subset of* $\mathbb{Z}$. *In fact, let* ω *denote the greatest common divisor of the coefficients* p_α, $\alpha \in A$. *Then, since for each* $y \in \mathbb{Z}^m$ *and each* $\alpha \in A$ *the value* y^α *is integer, all elements of* $\pi(\mathbb{Z}^m)$ *are multiples of* ω *(cf. Lemma 3.2.1 and, e.g., [22, Corollary 1.9]).*

To take advantage of this fact, we define

$$\lfloor p_0 \rfloor_\omega := \max\{z \in \omega\mathbb{Z} \mid z \leqslant p_0\}. \tag{3.8}$$

Then, like in the polyhedral case, we may relax p_0 *to any value arbitrarily close to* $\lfloor p_0 \rfloor_\omega + \omega$ *without admitting any additional solution of the inequality constraint. More explicitly, for every* $\tau \in [0, \lfloor p_0 \rfloor_\omega + \omega - p_0)$ *the relaxation*

$$G_\tau := \{y \in \mathbb{R}^m \mid \pi(y) - p_0 \leqslant \tau\}$$

of the set $G = \{y \in \mathbb{R}^m \mid \pi(y) - p_0 \leqslant 0\}$ *leads to the relaxation* $\widehat{M}_\rho := D \cap G_\tau$ *of* $\widehat{M}$ *with* $\rho = (0, \tau)$, *and this relaxation satisfies* $M = \widehat{M}_\rho \cap \mathbb{Z}^m$.

We emphasize that the above enlargement properties may even hold for $\tau \geqslant \lfloor p_0 \rfloor_\omega + \omega - p_0$, *as for any* $\alpha \in \mathbb{N}_0^n$ *with* $\alpha_j \neq 1$, $j = 1, \ldots, m$, *also the set* $\{y^\alpha \mid y \in \mathbb{Z}^m\}$ *is a proper subset of* $\mathbb{Z}$. *In fact, consider the enlargement for the set* $M = \{y \in D \cap \mathbb{Z}^m \mid \pi(y) \leqslant p_0\}$ *with a quadratic polynomial* $\pi(y) = y^\intercal Q y + \beta^\intercal y$. *Then for any* $k \in \mathbb{N}$ *it is possible to decide whether the quadratic equation* $\pi(y) = p_0 + k$ *has a solution in integers [33, 70]. Hence, with* $\widetilde{k} \in \mathbb{N}$ *being the first number for which the equation is solvable with the right-hand side* $p_0 + \widetilde{k}$, *and with* $\tau \in [0, \widetilde{k})$, *we may enlarge the right-hand side of the inequality constraint to* $p_0 + \tau$ *without admitting additional integer solutions.*

The subsequent example will show that $p_0 + \widetilde{k}$ *may be strictly larger than the value* $\lfloor p_0 \rfloor_\omega + \omega$ *from the above construction for general polynomials. However, we are only aware of effective implementations of procedures for the determination of* $\widetilde{k}$ *in special cases, e.g., when* $m = 2$ *holds, or when* π *is a sum of squares.*

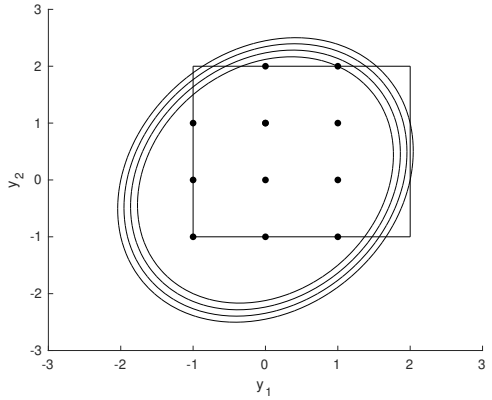


FIGURE 3.2: Enlargement for a quadratic polynomial

Example 3.3.2. *The convex-quadratic inequality constraint* $3y_1^2 - y_1y_2 + 2y_2^2 \leqslant 9$ *is of the form considered in Example 3.3.1 with* $A = \{(2,0),(1,1),(0,2)\}$, $p_{(2,0)} = 3$, $p_{(1,1)} = -1$, $p_{(0,2)} = 2$ *and* $p_{(0,0)} = 9$. *Moreover, let* $D = \{y \in \mathbb{R}^2 |\ -e \leqslant y \leqslant 2e\}$. *Figure 3.2 shows the box D, level lines of the function* $\pi(y) = 3y_1^2 - y_1y_2 + 2y_2^2$ *to the levels 9, 10, 11, and 12, and the set* $M = \{y \in D \cap \mathbb{Z}^2 |\ \pi(y) \leqslant 9\}$.

The first enlargement construction from Example 3.3.1 yields a greatest common divisor $\omega = 1$ *for the coefficients of* π *and, thus, the possibility to extend the right-hand side* 9 *of the inequality constraint to any value strictly below* $|9|_1 + 1 = 10$. *Hence, for any* $\tau \in [0,1)$ *the pair* $\rho = (0,\tau)$ *lies in R, and we obtain the enlarged inner parallel set*

$$\widehat{M}_\rho^- = D^- \cap G_\tau^- = \{y \in \mathbb{R}^2 |\ -\tfrac{1}{2}e \leqslant y \leqslant \tfrac{3}{2}e\} \cap G_\tau^-.$$

On the other hand, by inspection of Figure 3.2, the second enlargement construction from Example 3.3.1 yields $\widetilde{k} = 3$ *with the additional integer solutions* $(2,0)$, $(2,1)$ *and, thus, the possibility to even choose* $\tau \in [0,3)$.

Next, let us discuss the effect of the proposed enlargement ideas on the inner approximation T^- of $\widehat{M}^-$. We start by considering the case $\sigma = 0$, that is, only the right-hand side of the nonlinear inequality constraint $g(x,y) \leqslant 0$ is relaxed by some $\tau \geqslant 0$, and we have $\rho = (0,\tau)$. Then Lemma 2.3.3 yields the inner approximation

$$T_\rho^- \ = \ \{(x,y) \in D^- |\ g(x,y) + \tfrac{1}{2}L_\infty \leqslant \tau\}$$

of $\widehat{M}_\rho^-$, where the entries of the vector L_∞ are Lipschitz constants of the functions $g_i(x,y) - \tau_i$, $i \in I$, on D in the sense of Assumption 2.3.2. These Lipschitz constants, of course, coincide with those of the functions g_i, $i \in I$, so that the vector L_∞ does not depend on τ.

Example 3.3.3. *In Example 3.3.2 the corresponding Lipschitz constant for* $\pi(y) = 3y_1^2 - y_1y_2 + 2y_2^2$ *on D may be computed with the aid of Example 2.3.7, namely as the value*

$$L_\infty \ = \ \max_{y \in \mathrm{vert}D} \left\| \begin{pmatrix} 6 & -1 \\ -1 & 4 \end{pmatrix} y \right\|_1 \ = \ \left\| \begin{pmatrix} 6 & -1 \\ -1 & 4 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \end{pmatrix} \right\|_1 \ = \ 19.$$

This yields the inner approximation

$$\begin{aligned} T_\rho^- &= \{y \in \mathbb{R}^2 |\ -\tfrac{1}{2}e \leqslant y \leqslant \tfrac{3}{2}e,\ 3y_1^2 - y_1y_2 + 2y_2^2 - 9 + \tfrac{19}{2} \leqslant \tau\} \\ &= \{y \in \mathbb{R}^2 |\ -\tfrac{1}{2}e \leqslant y \leqslant \tfrac{3}{2}e,\ 3y_1^2 - y_1y_2 + 2y_2^2 + \tfrac{1}{2} \leqslant \tau\} \end{aligned}$$

of $\widehat{M}^-_\rho = D^- \cap G^-_\tau$ *for any* $\tau \in [0,3)$. *Hence,* T^-_ρ *is empty for all* $\tau \in [0,1/2)$, *but nonempty for* $\tau \in [1/2,3)$. *In particular, the above enlargement approach based on the computation of the greatest common divisor* ω *is already sufficient to enlarge the empty inner approximation* T^- *of* $\widehat{M}^-$ *to a nonempty inner approximation, e.g., to* $T^-_{(0,3/4)}$ *and, thus, to show granularity of the set* M.

Note that, as in Example 3.3.3, also in general the monotonicity property $T^-_{\rho^1} \subseteq T^-_{\rho^2}$ holds for all $\rho^1, \rho^2 \in R$ with $\rho^1 \leqslant \rho^2$ as long as σ^1 and σ^2 vanish.

After this consideration of the case $\sigma = 0$ let us turn to the effect of general enlargements with $\rho = (\sigma, \tau) \in R$ on the set T^-. By Lemma 2.3.3 the set

$$T^-_\rho = \{(x,y) \in D^-_\sigma \mid g(x,y) + \tfrac{1}{2}L_\infty \leqslant \tau\}$$

still is an inner approximation of $\widehat{M}^-_\rho$ if the vector L_∞ of Lipschitz constants is chosen according to Assumption 2.3.2. However, L_∞ depends on σ, which implies the undesirable effect that for $\rho^1, \rho^2 \in R$ with $\rho^1 \leqslant \rho^2$ one may no longer expect the inclusion $T^-_{\rho^1} \subseteq T^-_{\rho^2}$. The following section studies this issue in detail.

3.4 The Interplay Between Enlargements and Lipschitz Constants

For $\rho \in R$ Assumption 2.3.2 requires for all $i \in I$ the existence of some $L^{\sigma,i}_\infty$ such that for all $x \in \mathrm{pr}_x D_\sigma$ and all $y^1, y^2 \in D_\sigma(x)$ the estimate

$$|(g_i(x,y^1) - \tau_i) - (g_i(x,y^2) - \tau_i)| = |g_i(x,y^1) - g_i(x,y^2)| \leqslant L^{\sigma,i}_\infty \|y^1 - y^2\|_\infty$$

holds. While this condition does not depend on τ, for $\sigma^1 \leqslant \sigma^2$ the larger set D_{σ^2} requires potentially larger Lipschitz constants than D_{σ^1}. More precisely, the vector L^σ_∞ is monotonic in σ in the sense that $\sigma^1 \leqslant \sigma^2$ implies $L^{\sigma^1}_\infty \leqslant L^{\sigma^2}_\infty$ whenever the smallest known Lipschitz constants are chosen. For later reference let us explicitly state this modified version of Assumption 2.3.2.

Assumption 3.4.1. *For all* $i \in I$ *there exists some* $L^{\sigma,i}_\infty \geqslant 0$ *such that for all* $x \in \mathrm{pr}_x D_\sigma$ *and all* $y^1, y^2 \in D_\sigma(x)$ *we have*

$$|g_i(x,y^1) - g_i(x,y^2)| \leqslant L^{\sigma,i}_\infty \|y^1 - y^2\|_\infty.$$

The σ-dependence of L^σ_∞ leads to the inner approximation

$$\Gamma^-_\rho := \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid g(x,y) + \tfrac{1}{2}L^\sigma_\infty \leqslant \tau\}$$

of G^-_τ which does not only depend on τ, but also on σ, and to the inner approximation

$$T^-_\rho = \{(x,y) \in D^-_\sigma \mid g(x,y) + \tfrac{1}{2}L^\sigma_\infty \leqslant \tau\} = D^-_\sigma \cap \Gamma^-_\rho \tag{3.9}$$

of $\widehat{M}^-_\rho$. The mentioned undesirable effect can now be formulated by the observation that, on the one hand, for $\sigma^1 = \sigma^2$ and $\tau^1 \leqslant \tau^2$ we have $\Gamma^-_{\rho^1} \subseteq \Gamma^-_{\rho^2}$ whereas, on the other hand, for $\sigma^1 \leqslant \sigma^2$ and $\tau^1 = \tau^2$ the monotonicity of the Lipschitz constants leads to the reverse inclusion $\Gamma^-_{\rho^1} \supseteq \Gamma^-_{\rho^2}$. Hence, for arbitrary $\rho^1 \leqslant \rho^2$ the inclusion $T^-_{\rho^1} \subseteq T^-_{\rho^2}$ cannot be expected.

At least we can state that an inner approximation $T_\rho^- = D_\sigma^- \cap \Gamma_\rho^-$ becomes larger when the vector τ is increased (component-wise), and that there is trade-off between increasing D_σ^- and shrinking Γ_ρ^- when σ is increased (component-wise). If the size of R permits, of course one may compensate the undesirable effect in the behavior of L_∞^σ by choosing $\tau \geqslant L_\infty^\sigma/2$ for given σ. In general, however, the determination of some $\rho = (\sigma, \tau)$ with $T_\rho^- \neq \varnothing$ is problem dependent and potentially nontrivial.

Example 3.4.2. *In Example 3.3.2 the set $D = \{y \in \mathbb{R}^2 |\ -e \leqslant y \leqslant 2e\}$ is described by box constraints and may, hence, be enlarged by the technique from Example 3.1.2. We choose to enlarge all four constraints simultaneously by $\delta \in [0,1)$, that is, we put $D_\delta := \{y \in \mathbb{R}^2 |\ -e-\delta e \leqslant y \leqslant 2e+\delta e\}$. By Example 2.3.7 the corresponding Lipschitz constant for $\pi(y) = 3y_1^2 - y_1y_2 + 2y_2^2$ on D_δ is*

$$L_\infty^\delta \;=\; \max_{y \in \mathrm{vert} D_\delta} \left\| \begin{pmatrix} 6 & -1 \\ -1 & 4 \end{pmatrix} y \right\|_1 \;=\; \left\| \begin{pmatrix} 6 & -1 \\ -1 & 4 \end{pmatrix} \begin{pmatrix} 2+\delta \\ -1-\delta \end{pmatrix} \right\|_1 \;=\; 19 + 12\delta$$

which yields the inner approximation

$$T_{(\delta,\tau)}^- = \{y \in \mathbb{R}^2 |\ -\tfrac{1}{2}e - \delta e \leqslant y \leqslant \tfrac{3}{2}e + \delta e,\ 3y_1^2 - y_1y_2 + 2y_2^2 + \tfrac{1}{2} + 6\delta \leqslant \tau\}$$

of $\widehat{M}_{(\delta,\tau)}^- = D_\delta^- \cap G_\tau^-$ for any $\delta \in [0,1)$ and $\tau \in [0,3)$. As we have seen in Example 3.3.3, the set $T_{(0,3/4)}^-$ is nonempty. For increasing δ, however, instead of growing larger the set $T_{(\delta,3/4)}^-$ shrinks to a singleton for $\delta = 1/24$ and then becomes empty.

3.5 Pseudo-Granularity

In the previous section we have seen that the size and behavior of the vector of Lipschitz constants L_∞^σ from Assumption 3.4.1 may have a strong effect on the chances for consistency of the inner approximation

$$T_\rho^- \;=\; \{(x,y) \in D_\sigma^- |\ g(x,y) + \tfrac{1}{2}L_\infty^\sigma \leqslant \tau\}$$

of $\widehat{M}_\rho^-$ and, thus, on the chances for the algorithmic employment of granularity. For this reason, the present section will introduce a way to work with smaller Lipschitz constants, albeit at the price of modifying the geometrically intuitive idea of granularity to the algorithmically more attractive concept of pseudo-granularity. We remark that we will only modify appearing Lipschitz constants in the construction of T^- and hence this will only make a difference under the occurrence of nonlinear constraint functions. In particular, pseudo-granularity will coincide with granularity for MILPs.

Notice that replacing the set D_σ by a smaller set in Assumption 3.4.1 would actually allow to compute better Lipschitz constants and hence larger sets T_δ^-. The present section will show how this idea can be employed if we do without inner parallel sets and their inner approximations in the first place.

In fact, in this section let us write

$$M \;=\; \{(x,y) \in \mathbb{R}^n \times \mathbb{Z}^m |\ Ax + By \leqslant b,\ y^\ell \leqslant y \leqslant y^u,\ g(x,y) \leqslant 0\}.$$

We assume that no further box constraints on integer variables are modeled by the system $Ax + By \leqslant b$, that is, with α_i^T and β_i^T, $i = 1, \ldots, p$, denoting the rows of A

and B, respectively, in the case $\alpha_i = 0$ the vector β_i contains at least two nonzero real entries.

For the enlargement of D we choose $\sigma := (\sigma^b, \sigma^\ell, \sigma^u) \geqslant 0$ and define

$$D_\sigma := \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid Ax + By \leqslant b + \sigma^b,\ y^\ell - \sigma^\ell \leqslant y \leqslant y^u + \sigma^u\}$$

with the inner parallel set

$$D_\sigma^- := \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid Ax + By \leqslant b + \sigma^b - \tfrac{1}{2}\|\beta\|_1,\ y^\ell - \sigma^\ell + \tfrac{1}{2}e \leqslant y \leqslant y^u + \sigma^u - \tfrac{1}{2}e\}.$$

Note that in contrast to (3.7) we allow for different enlargement parameters. As will become apparent shortly, this enables dealing with the shrinking effects of Γ_ρ^- with the enlargement of D_σ more effectively.

As above, we also consider the sets $G_\tau = \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid g(x,y) \leqslant \tau\}$ and $\widehat{M}_\rho = D_\sigma \cap G_\tau$ and, as in (3.3), collect the enlargement vectors $\rho = (\sigma, \tau) \geqslant 0$ with $M = \widehat{M}_\rho \cap (\mathbb{R}^n \times \mathbb{Z}^m)$ in the set R. Recall that the main arguments from the previous sections relied on considering the inner parallel set $\widehat{M}_\rho^-$ and, in particular, the inclusion $T_\rho^- \subseteq \widehat{M}_\rho^-$. In the present section we follow a different route and define a set $\widetilde{T}_\rho^- \supseteq T_\rho^-$ for which we directly show that roundings of its elements lie in $\widehat{M}_\rho$, and thus in M.

First of all, from the observations in Example 3.1.2 it is clear that we may choose $\rho \in R$ with $0 \leqslant \sigma^\ell, \sigma^u < e$. Also the vector σ^b may be constructed along the lines of (3.7) which yields

$$\sigma_i^b \in \begin{cases} [0, \lfloor b_i \rfloor_{\omega_i} + \omega_i - b_i), & \text{if } \omega_i > 0, \\ \{0\}, & \text{else.} \end{cases} \tag{3.10}$$

Now let us focus on choices $e/2 \leqslant \sigma^\ell, \sigma^u < e$. The crucial observation for the following is that then elements from D_σ^- and their roundings not only lie in the set D_σ but even in its subset

$$\widetilde{D}_\sigma := \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid Ax + By \leqslant \lfloor b \rfloor_\omega,\ y^\ell - \sigma^\ell + \tfrac{1}{2}e \leqslant y \leqslant y^u + \sigma^u - \tfrac{1}{2}e\}.$$

Lemma 3.5.1. *For $D = \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid Ax + By \leqslant b,\ y^\ell \leqslant y \leqslant y^u\}$ and any $\sigma = (\sigma^b, \sigma^\ell, \sigma^u)$ with σ^b satisfying* (3.10) *as well as $e/2 \leqslant \sigma^\ell, \sigma^u < e$, the following assertions are true:*

a) The chain of inclusions $D_\sigma^- \subseteq \widetilde{D}_\sigma \subseteq D_\sigma$ holds.

b) Any rounding $(\check{x}, \check{y})$ of any point $(x,y) \in D_\sigma^-$ lies in $\widetilde{D}_\sigma$.

Proof. For the proof of the first inclusion in part a we show $b + \sigma^b - \|\beta\|_1/2 \leqslant \lfloor b \rfloor_\omega$. By (3.10) this relation is clear in all components with $\omega_i = 0$. Moreover, we have $\sigma_i^b < \lfloor b_i \rfloor_{\omega_i} + \omega_i - b_i$ for all i with $\omega_i > 0$ and, thus, it suffices to show $\omega_i \leqslant \|\beta_i\|_1/2$ for these i. As the greatest common divisor of the entries of β_i, each of these entries upper bounds ω_i, and since we assume at least two nonzero entries of β_i, the required inequality follows. The second inclusion in part a immediately follows from $\lfloor b \rfloor_\omega \leqslant b$ and $\sigma^b \geqslant 0$.

For the proof of part b let $(x,y) \in D_\sigma^-$ and let $(\check{x}, \check{y})$ denote one of its roundings. The constraints $y^\ell - \sigma^\ell + e/2 \leqslant y \leqslant y^u + \sigma^u - e/2$ with $\sigma^\ell, \sigma^u < e$ enforce $y^\ell \leqslant \check{y} \leqslant y^u$ and, due to $\sigma^\ell, \sigma^u \geqslant e/2$, also $y^\ell - \sigma^\ell + e/2 \leqslant \check{y} \leqslant y^u + \sigma^u - e/2$.

It remains to show the validity of the constraint $A\breve{x} + B\breve{y} \leqslant \lfloor b \rfloor_\omega$. In fact, the enlargement construction yields $(\breve{x}, \breve{y}) \in D_\sigma \cap (\mathbb{R}^n \times \mathbb{Z}^m) = D \cap (\mathbb{R}^n \times \mathbb{Z}^m) \subseteq D$ which implies $A\breve{x} + B\breve{y} \leqslant b$ and, thus, the asserted constraint for each component with $\omega_i = 0$. Moreover, in the case $\omega_i > 0$ we have $\alpha_i = 0$ and $\beta_i \in \mathbb{Z}^m \backslash \{0\}$ so that for $\breve{y} \in \mathbb{Z}^m$ the right-hand side b_i of the constraint can be reduced to $\lfloor b_i \rfloor_{\omega_i}$. The assertion hence is shown. □

In view of Lemma 3.5.1 we may replace the Lipschitz conditions on the set D_σ from Assumption 3.4.1 by Lipschitz conditions on the smaller set $\widetilde{D}_\sigma$ to obtain potentially smaller Lipschitz constants. The relaxed Assumption 3.4.1 reads as follows.

Assumption 3.5.2. *For $D = \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid Ax + By \leqslant b,\ y^\ell \leqslant y \leqslant y^u\}$ and $\sigma = (\sigma^b, \sigma^\ell, \sigma^u)$ with σ^b satisfying* (3.10) *as well as $e/2 \leqslant \sigma^\ell, \sigma^u < e$, let for all $i \in I$ exist some $\widetilde{L}_\infty^{\sigma,i} \geqslant 0$ such that for all $x \in \mathrm{pr}_x \widetilde{D}_\sigma$ and all $y^1, y^2 \in \widetilde{D}_\sigma(x)$ we have*

$$|g_i(x,y^1) - g_i(x,y^2)| \ \leqslant\ \widetilde{L}_\infty^{\sigma,i} \|y^1 - y^2\|_\infty.$$

Notice that, to ensure the validity of Lemma 3.5.1b for $0 \leqslant \sigma^\ell, \sigma^u < e/2$, we would need to include at least unmodified box constraints $y^\ell \leqslant y \leqslant y^u$ in $\widetilde{D}_\sigma$. This implies that in Assumption 3.5.2 choices $0 \leqslant \sigma^\ell, \sigma^u < e/2$ yield no advantage compared to $\sigma^\ell = \sigma^u = e/2$ and that considering only $e/2 \leqslant \sigma^\ell, \sigma^u < e$ therefore covers all relevant cases.

With the aid of the Lipschitz constants from Assumption 3.5.2, and with some τ such that $\rho = (\sigma, \tau)$ lies in R, we define the sets

$$\widetilde{\Gamma}_\rho^- \ :=\ \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid g(x,y) + \tfrac{1}{2}\widetilde{L}_\infty^\sigma \leqslant \tau\}$$

and

$$\widetilde{T}_\rho^- := \{(x,y) \in D_\sigma^- \mid g(x,y) + \tfrac{1}{2}\widetilde{L}_\infty^\sigma \leqslant \tau\} = D_\sigma^- \cap \widetilde{\Gamma}_\rho^-.$$

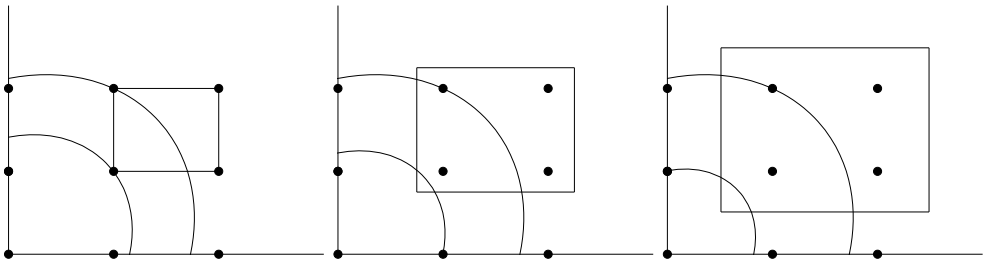
Note that $\widetilde{T}_\rho^-$ differs from the set T_ρ^- in (3.9) only by the choice of the vector of Lipschitz constants. Recall that these Lipschitz constants may be too small to conduct the proof of Lemma 2.3.3. In fact, examples show that $\widetilde{T}_\rho^-$ is no longer necessarily an inner approximation of the inner parallel set $\widehat{M}_\rho^-$ and, hence, the consistency of $\widetilde{T}_\rho^-$ may not be used as a sufficient condition for the granularity of M. Since subsequently it will still turn out to be very useful, we formulate the following concept.

Definition 3.5.3. *We call the set M* pseudo-granular *if for some $\rho \in R$ and for the Lipschitz constants from Assumption 3.5.2 the set $\widetilde{T}_\rho^-$ is nonempty. Moreover, we call a problem MINLP pseudo-granular if its feasible set M is pseudo-granular.*

We mention that consistency of the set T_ρ^- from (3.9) is sufficient for granularity of the problem $MINLP$ (by Theorem 2.3.4) as well as for its pseudo-granularity (in view of $\widetilde{L}_\infty^\sigma \leqslant L_\infty^\sigma$ and, thus, $T_\rho^- \subseteq \widetilde{T}_\rho^-$). However, neither granularity nor pseudo-granularity implies $T_\rho^- \neq \varnothing$.

As the next result shows, it is possible to maintain the assertions of Theorem 2.3.4 when granularity is replaced by pseudo-granularity.

Theorem 3.5.4. *Let $D = \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid Ax + By \leqslant b,\ y^\ell \leqslant y \leqslant y^u\}$, let $\rho = (\tau, \sigma) \in R$ with $\sigma = (\sigma^b, \sigma^\ell, \sigma^u)$, σ^b satisfying* (3.10) *as well as $e/2 \leqslant \sigma^\ell, \sigma^u < e$, and let Assumption 3.5.2 hold. Then the following assertions are true:*



(a) $\delta = 0.5$ (b) $\delta = 0.75$ (c) $\delta = 0.99$

FIGURE 3.3: $\widetilde{T}^-_{(\delta,\tau)}$ for different enlargement parameters δ and $\tau = 2.9$

a) For any point $(x, y) \in \widetilde{T}^-_\rho$, any of its roundings $(\breve{x}, \breve{y})$ lies in M.

b) If the problem MINLP is pseudo-granular, then it is consistent.

Proof. In the case $\widetilde{T}^-_\rho = \varnothing$ the assertion of part a trivially holds. Otherwise, let $(x, y) \in \widetilde{T}^-_\rho = D^-_\sigma \cap \widetilde{\Gamma}^-_\rho$. We have to show that any rounding $(\breve{x}, \breve{y})$ of (x, y) lies in $M = D_\sigma \cap G_\tau \cap (\mathbb{R}^n \times \mathbb{Z}^m)$.

From the definition of a rounding, $(\breve{x}, \breve{y}) \in \mathbb{R}^n \times \mathbb{Z}^m$ is clear. Furthermore, $(x, y) \in D^-_\sigma$ implies $(\breve{x}, \breve{y}) \in D_\sigma$.

It remains to show $(\breve{x}, \breve{y}) \in G_\tau$, that is, $g_i(\breve{x}, \breve{y}) \leqslant \tau_i$ for all $i \in I$. In fact, for any $i \in I$ we have $g_i(\breve{x}, \breve{y}) \leqslant g_i(x, y) + |g_i(\breve{x}, \breve{y}) - g_i(x, y)|$, where Lemma 3.5.1 implies $(\breve{x}, \breve{y}), (x, y) \in \widetilde{D}_\sigma$. Hence, with the Lipschitz constant $\widetilde{L}^{\sigma,i}_\infty$ from Assumption 3.5.2 we obtain

$$g_i(\breve{x}, \breve{y}) \leqslant g_i(x, y) + \widetilde{L}^{\sigma,i}_\infty \|\breve{y} - y\|_\infty \leqslant g_i(x, y) + \tfrac{1}{2}\widetilde{L}^{\sigma,i}_\infty \leqslant \tau_i$$

where the last inequality stems from the definition of $\widetilde{T}^-_\rho$. This shows $(\breve{x}, \breve{y}) \in G_\tau$ and, thus, the assertion of part a. In view of Definition 3.5.3, part b is an immediate consequence of part a. □

Example 3.5.5. *The setting of Example 3.4.2 is suitable for the application of the above techniques. In fact, to make this example more challenging and the occurring effects more apparent, let us change the lower bounds of both variables from -1 to 1. Thus, we obtain $D = \{y \in \mathbb{R}^2 \,|\, e \leqslant y \leqslant 2e\}$ and the feasible set $M = D \cap G = \{(1, 1)^\intercal, (1, 2)^\intercal\}$ contains only two points. The square in Figure 3.3a actually illustrates the shape of the set D, and the "outer" nonlinear level curve corresponds to the condition $\pi(y) = 9$, that is, to the boundary points of the set G.*

Since all inequality constraints in the description of D are box constraints, the set $\widetilde{D}_\delta = \{y \in \mathbb{R}^2 \,|\, 3e/2 - \delta e \leqslant y \leqslant 3e/2 + \delta e\}$ turns out to coincide with the enlarged inner parallel set D^-_δ. Moreover, the condition $e/2 \leqslant \sigma^\ell, \sigma^u < 1$ is satisfied for all choices $\delta \in [1/2, 1)$, and from Example 3.3.2 we know that with all choices $\tau \in [0, 3)$ we have $\rho = (\delta, \tau) \in R$.

The benefit of the pseudo-granularity concept is that for given $\rho \in R$ we do not have to compute the Lipschitz constant of the function π on the set $D_\delta = \{y \in \mathbb{R}^2 \,|\, e - \delta e \leqslant y \leqslant 2e + \delta e\}$, but only on the smaller set $\widetilde{D}_\delta$. This Lipschitz constant from Assumption 3.5.2 is computed to be

$$\widetilde{L}^\delta_\infty = \max_{y \in \operatorname{vert} \widetilde{D}_\delta} \left\| \begin{pmatrix} 6 & -1 \\ -1 & 4 \end{pmatrix} y \right\|_1 = \left\| \begin{pmatrix} 6 & -1 \\ -1 & 4 \end{pmatrix} \begin{pmatrix} 3/2 + \delta \\ 3/2 + \delta \end{pmatrix} \right\|_1 = 12 + 8\delta.$$

It yields the sets

$$\widetilde{\Gamma}^-_{(\delta,\tau)} = \{y \in \mathbb{R}^2 |\ 3y_1^2 - y_1y_2 + 2y_2^2 - 3 + 4\delta \leqslant \tau\}$$

and $\widetilde{T}_{(\delta,\tau)} = D^-_\delta \cap \widetilde{\Gamma}^-_{(\delta,\tau)}$. *The structure of the latter set is illustrated in Figure 3.3 for the cases* $\tau = 2.9$ *and* $\delta \in \{0.5, 0.75, 0.99\}$, *where the "inner" nonlinear level curve visualizes the shrinking behavior of the set* $\widetilde{\Gamma}^-_{(\delta,\tau)}$ *with increasing values of* δ.

Since the coordinates of the lower left vertex of $D^-_{0.5}$ *violate the inequality in the definition of* $\widetilde{\Gamma}^-_{(0.5,\tau)}$ *for any* $\tau < 3$, *the set* $\widetilde{T}^-_{(0.5,2.9)} = D^-_{0.5} \cap \widetilde{\Gamma}^-_{(0.5,2.9)}$ *is empty. This corresponds to the situation illustrated in Figure 3.3a. However, in this example the enlarging effect of* D^-_δ *with increasing values of* δ outweighs *the shrinking effect of* $\widetilde{\Gamma}^-_{(\delta,2.9)}$ *so that* $\widetilde{T}^-_{(\delta,2.9)}$ *becomes larger with increasing values of* δ. *In particular,* $\widetilde{T}^-_{(\delta,2.9)}$ *is nonempty for any* $\delta \in (0.5, 1)$, *which proves the pseudo-granularity of the feasible set* M. *Figure 3.3 also illustrates that the rounding of any point from the nonempty set* $\widetilde{T}^-_{(\delta,2.9)}$ *turns out to be the point* $(1,1)^\intercal \in M$.

We remark that, while pseudo-granularity of M *holds, one cannot prove granularity of* M *by showing the consistency of the set* $T^-_{(\delta,\tau)}$ *for some* $\delta \in [0,1)$ *and* $\tau \in [0,1)$. *Indeed, computing the Lipschitz constant of* π *on* D_δ *yields* $L^\delta_\infty = 16 + 8\delta$ *and hence*

$$T^-_{(\delta,\tau)} = \{y \in D^-_\delta |\ 3y_1^2 - y_1y_2 + 2y_2^2 - 1 + 4\delta \leqslant \tau\}.$$

Again, since the lower left vertex of D^-_δ *does not satisfy the nonlinear constraint for any* $\delta \in [0,1)$ *and* $\tau \in [0,3)$, *the set* $T^-_{(\delta,\tau)}$ *is empty for all these parameters.*

We remark that Theorem 3.5.4 has important implications for algorithmic considerations. While we observe opposing effects regarding pseudo-granularity when entries of σ^u and σ^ℓ vary between $1/2$ and 1, the values for σ^b should *always* be chosen *as large as possible*. Indeed, Theorem 3.5.4 eliminates all negative effects on the Lipschitz constant that previously occurred with increasing values of σ^b, since the set $\widetilde{D}_\sigma$ and the vector of corresponding Lipschitz constants $\widetilde{L}^\sigma_\infty$ do not depend on σ^b. Moreover, as highlighted in Example 3.5.5, the tighter box constraints in the definition of the set $\widetilde{D}_\sigma$ compared to D_σ yield another crucial advantage.

Chapter 4

Using Inner Parallel Sets for Computing Feasible Points

This chapter discusses the concrete algorithmic employment of the (pseudo-)granularity concept and provides numerical results. In Section 4.1 we summarize the findings from the previous chapters by developing a specific algorithmic scheme that can be applied to mixed-integer linear and nonlinear optimization problems. Subsequently, we conduct a comprehensive computational study on problems from standard libraries. Sections 4.2 and 4.3 discuss results for mixed-integer linear and nonlinear optimization problems, respectively, and Section 4.4 offers a brief summary.

4.1 Algorithmic Considerations

Before we introduce a feasibility test and a method for the construction of a good feasible point, let us first summarize the necessary computation of parameters.

Computation of Pseudo-Granularity Parameters:

1. Set σ^ℓ and σ^u to some value satisfying $e/2 \leqslant \sigma^\ell, \sigma^u \leqslant e$, and σ^b to some value satisfying (3.10). We recommend using Equation (3.7) with δ close to one.
2. For each $i \in I$, compute a Lipschitz constant $\widetilde{L}_\infty^{\sigma,i}$ as an upper bound for the optimal value of

   $$\max \|\nabla_y g_i(x,y)\|_1 \text{ s.t. } Ax + By \leqslant \lfloor b \rfloor_\omega,\ y^\ell - \sigma^\ell + \tfrac{1}{2}e \leqslant y \leqslant y^u + \sigma^u - \tfrac{1}{2}e$$

   (cf. Section 2.3.2 and Section 3.5).
3. Determine valid nonlinear enlargement parameters τ_i, $i \in I$, such that $\rho = (\sigma, \tau) \in R$ (cf. (3.3)).

After the computation of the parameters σ, τ and $\widetilde{L}_\infty^\sigma$ all necessary data for computing the set

$$\widetilde{T}_\rho^- := \{(x,y) \in D_\sigma^- \mid g(x,y) + \tfrac{1}{2}\widetilde{L}_\infty^\sigma \leqslant \tau\}$$

is available.

According to Theorem 3.5.4b, a straightforward sufficient condition for the consistency of *MINLP* is a successful feasibility test for the latter set, for example in the following form.

FRA-SLOR (Feasible Rounding Approach by Shrink-Lift-Optimize-Round):

1. Compute the pseudo-granularity parameters σ, τ and $\widetilde{L}_\infty^\sigma$.
2. Compute an optimal point (x^f, y^f) of the *feasibility* problem

$$\begin{aligned} F^- : \quad \min_{(x,y,z)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}} \; z \quad \text{s.t.} \quad & Ax + By - ze \leqslant b + \sigma^b - \tfrac{1}{2}\|\beta\|_1, \\ & y^\ell - \sigma^\ell + \tfrac{1}{2}e - ze \leqslant y \leqslant y^u + \sigma^u - \tfrac{1}{2}e + ze, \\ & g(x,y) - ze \leqslant \tau - \tfrac{1}{2}\widetilde{L}_\infty^\sigma, \\ & z \geqslant -1. \end{aligned}$$

3. If v_{F^-} is nonpositive, then $MINLP$ is pseudo-granular and we can round (x^f, y^f) to a feasible point $(\breve{x}^f, \breve{y}^f)$.

Note that if all functions $g_i, i \in I$, are smooth and convex, the feasibility test FRA-SLOR can be performed efficiently once all Lipschitz constants and enlargement parameters are determined. Under such assumptions, the test provides an efficiently computable sufficient condition for the existence of feasible points for $MINLP$. As indicated earlier, for the practical purpose of computing a feasible point of good quality it is beneficial to explicitly take the objective function f into account.

FRA-SOR (Feasible Rounding Approach by Shrink-Optimize-Round):

1. Compute the pseudo-granularity parameters σ, τ and $\widetilde{L}_\infty^\sigma$.
2. Compute an optimal point (x^{ob}, y^{ob}) of f over $\widetilde{T}_\rho^-$, that is, of the *objective based* problem

$$\begin{aligned} P_\rho^{ob} : \quad \min_{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m} \; f(x,y) \quad \text{s.t.} \quad & Ax + By \leqslant b + \sigma^b - \tfrac{1}{2}\|\beta\|_1, \\ & y^\ell - \sigma^\ell + \tfrac{1}{2}e \leqslant y \leqslant y^u + \sigma^u - \tfrac{1}{2}e, \\ & g(x,y) \leqslant \tau - \tfrac{1}{2}\widetilde{L}_\infty^\sigma. \end{aligned}$$

3. Round (x^{ob}, y^{ob}) to $(\breve{x}^{ob}, \breve{y}^{ob}) \in M$.

For MILPs, Step 1 of FRA-SLOR and FRA-SOR are simplified since from the three steps in the computation of pseudo-granularity parameters only Step 1 has to be performed. We also mention that the consistency of the optimization problem P_ρ^{ob} in Step 2 implies that the problem $MINLP$ is pseudo-granular. Hence, the existence of some optimal point of P_ρ^{ob} in particular proves pseudo-granularity of $MINLP$.

For FRA-SLOR we emphasize that even if $MINLP$ is not pseudo-granular, the point $(\breve{x}^f, \breve{y}^f)$ can nevertheless be feasible for $MINLP$. Even if it is infeasible, it might at least be a good starting point for heuristic procedures, e.g., for pumping cycles of the feasibility pump. Therefore, even for a non-granular problem $MINLP$, FRA-SLOR might prove useful for the computation of feasible points.

Clearly this is a crucial advantage of FRA-SLOR over FRA-SOR, as well as the fact that FRA-SLOR does not involve a potentially complicating objective function. On the other hand we expect $(\breve{x}^{ob}, \breve{y}^{ob})$ to have a better objective value than $(\breve{x}^f, \breve{y}^f)$.

As one is usually interested in good points in M, this is a significant advantage of FRA-SOR over FRA-SLOR. In any case, one can initially check (pseudo-)granularity by using FRA-SLOR and, in the (pseudo-)granular case, subsequently apply FRA-SOR, using (x^f, y^f) as an initial feasible point.

In the computation of the set $D^-_{\sigma(\delta)}$ (cf. Equation (3.7)), if in some row i we only have $\alpha_i = 0$ but not $\beta_i \in \mathbb{Z}^m$, we may either multiply β_i and b_i with a constant to make all entries integer (which always is possible for rational entries) or, otherwise, simply set ω_i to zero. Moreover, although ω can be computed efficiently, we remark that setting $\omega_i := 1$ for all rows with $\alpha_i = 0$ and $\beta_i \in \mathbb{Z}^m \backslash \{0\}$ also is a possible choice which computationally comes for free and may thus be useful for practical applications.

Remark 4.1.1. *Due to rounding effects, we generally cannot expect the generated feasible point to lie on the boundary of M. If continuous variables are present ($n > 0$) we might therefore end up with an idle potential for the objective function to be improved, which is accessible by optimizing over the continuous variables with integer variables being fixed to the values of the generated feasible point. We did not utilize this potential in the subsequent computational analysis of this chapter, but shall formally introduce and explicitly use this step in later chapters.*

4.2 Computational Study in the Polyhedral Case

Our first computational study comprises results from a test bed of optimization problems from the MIPLIB 2003 [3] and the MIPLIB 2010 [46]. The intention of our computational study is fourfold.

Firstly, our algorithmic considerations imply that finding feasible points for granular mixed-integer linear problems is easy, at the very least if we apply the feasible rounding approaches. Hence, by checking problems from our test bed for granularity in Section 4.2.1 we obtain a certificate for the consistency of the feasible set and for the possibility to efficiently compute a feasible point of the corresponding problem.

Secondly, in contrast to other approaches from the literature, the feasible rounding approaches yield *polynomial worst case complexity* for computing feasible points for these problems. It is natural to ask, however, if granularity coincides with a structure, which is (implicitly) exploited not only by the feasible rounding approaches, but also by commonly used heuristics. We aim to answer this question in Section 4.2.2 by a comparison of the feasible rounding approaches against Gurobi, which we specifically tune on finding feasible points as quickly as possible. We stress that the aim is *not* to benchmark our approaches against Gurobi's vast arsenal of heuristics. Rather, we wish to provide a proof of concept for our methods and to determine if there are problems for which it is indeed advantageous to make explicit use of the granularity concept.

Thirdly, in Section 4.2.3 we investigate if FRA-SLOR is also able to compute feasible points for nongranular applications.

We emphasize that solvers like Gurobi apply a presolving step to optimization models before actually solving them (cf. [2] for a recent survey on different techniques). In order to also test the potential usefulness of our approaches within the framework of such a solver, we conduct the main part of our computational study (Sections 4.2.1 - 4.2.3) on presolved models.

Finally, in Section 4.2.4, we conclude our numerical analysis by investigating the impact of this presolving step on granularity. Here, we evaluate in what way our

results change when we apply the feasible rounding approaches to models which are *not* presolved. Recall that we illustrated the potential influence of different modeling techniques on granularity in Section 3.2. We will demonstrate that similar effects occur when presolving techniques are applied.

Both feasible rounding approaches are implemented in Matlab R2016b and all arising optimization problems are solved with Gurobi 7. All tests are run on a personal computer with two cores à 2.3 GHz and 8 GB RAM. Unless stated otherwise, we use Gurobi's default parameters, including a feasibility tolerance of 10^{-6}. We constructed the enlarged inner parallel set in accordance to Equation (3.7) and set δ to $1 - 10^{-4}$, so that our roundings do not produce infeasible points, even if Gurobi returns a 10^{-6}-(in)feasible point.

As mentioned in Remark 3.1.1, while a nonempty inner parallel set for a fixed value of δ shows granularity and thus also the granular-representability of M, an empty inner parallel set does not imply that M is not granularity-representable. In this section we call a problem granular (not granular), if the enlarged inner parallel set $\widehat{M}_\delta^- = D_\delta^-$ is nonempty (empty) for the enlargement parameter $\delta = 1 - 10^{-4}$.

As indicated above, we only consider problems which do not contain equality constraints on integral variables and leave studying these problems to Chapter 6. On the other hand, equality constraints which only contain continuous variables can be incorporated unaltered in the construction of the inner parallel set. Hence, such instances are included in the current study.

Moreover, some problems from the libraries are formulated with non-integral values in the constraint matrix B (at rows i where $\alpha_i = 0$ holds). With rational numbers, it is straightforward to scale these numbers such that the corresponding rows of B are integral. However, their representation as floating point numbers makes this scaling algorithmically intractable without changing the corresponding problem by introducing computational errors. Therefore, we did not enlarge the respective restrictions for these problems in our computational study.

4.2.1 Granular Optimization Problems from the MIPLIB Libraries

Applying Gurobi's presolving step and excluding problems with equality constraints containing integral variables from the MIPLIB libraries results in 151 problems which we may efficiently test for δ-granularity by using the feasible rounding approaches. Out of these, 137 are from the MIPLIB 2010, 19 from the MIPLIB 2003, and 5 problems occur in both libraries.

We report that **81** problems from our test bed are granular. Recall that this immediately proves consistency for these problems. Table 4.1 lists these problems, together with the number of continuous variables n, the number of integer variables m, the number of binary variables b ($\leqslant m$) and the number of rows p of the constraint matrix. Problems marked with * are from the MIPLIB 2003 and problems marked with ** occur in both libraries. The remaining problems are from the MIPLIB 2010. We remark that, due to Gurobi's presolving step, the values of n, m, b and p often differ from the ones stated in [3] and [46].

Remarkably, we find that even problems with a significant number of binary variables are granular, which coincides with the intuition given in Examples 3.1.2 and 3.2.4. In fact, some problems like *ex1010-pi* only use binary variables. Therefore, our first computational finding indicates that granularity is a characteristic which may be expected in various real world problems. On the other hand, theoretical results from [62] predict a better performance (especially in terms of the objective value

name	n	m (b)	p
30_70_4.5_0.95_100	0	10959(10958)	12503
50v-10	366	1647(1464)	233
a1c1s1**	2297	192(192)	2053
b2c1s1	2389	288(288)	2546
beasleyC3	852	852(852)	1153
bg512142	527	230(230)	897
buildingenergy	128688	26287(0)	277589
core2536-691	0	11053(11052)	1892
core4872-1529	0	14954(14954)	3981
cov1075	0	120(120)	637
dfn-gwin-UUM	846	90(0)	156
dg012142	1299	600(600)	1987
ex1010-pi	0	11568(11568)	1466
fast0507*	0	20334(20334)	440
fixnet6*	499	378(378)	477
g200x740i	740	740(740)	940
ger50_17_trans	4320	18062(0)	498
germany50-DBM	8078	88(0)	2510
go19	0	361(361)	361
iis-100-0-cov	0	100(100)	3831
iis-bupa-cov	0	337(337)	4796
iis-pima-cov	0	698(698)	7122
janos-us-DDM	2095	84(0)	755
k16x240	240	240(240)	256
m100n500k4r1	0	500(500)	100
macrophage	0	1889(1889)	2708
manna81*	0	3321(18)	6480
mc11	1517	1518(1518)	1917
methanosarcina	0	7865(7865)	14538
mik.250-1-100.1	1	250(100)	100
modglob*	256	98(98)	286
n15-3	152360	780(0)	29494
n3-3	8236	348(0)	2194
n3700	5000	5000(5000)	5150
n3705	5000	5000(5000)	5150
n370a	5000	5000(5000)	5150
n4-3	2950	150(0)	976
n9-3	6864	234(0)	2082
neos-1112782	2025	2025(2025)	2070
neos-1112787	1600	1600(1600)	1640
neos-1225589	625	625(625)	650

name	n	m(b)	p
neos-1616732	0	200(200)	1026
neos-932816	148	6216(6216)	2568
neos-933638	800	8087(8087)	8053
neos-933966	800	7432(7432)	6590
neos-934278	767	7354(7354)	7238
neos15	545	154(154)	460
npmv07	158248	1880(1880)	60723
ns4-pr3	8045	58(0)	1936
ns4-pr9	6741	39(0)	1910
opm2-z10-s2	0	5940(5940)	65667
opm2-z11-s8	0	7636(7636)	88677
opm2-z12-s14	0	10323(10323)	124615
opm2-z12-s7	0	10328(10328)	124780
opm2-z7-s2	0	1892(1892)	15793
p100x588b	588	588(588)	688
p6b	0	451(451)	502
p80x400b	396	396(396)	474
pb-simp-nonunif	0	11710(11710)	122652
pp08a*	170	64(64)	133
pp08aCUTS*	171	64(64)	239
qiu**	792	48(48)	1192
queens-30	0	900(900)	900
r80x800	800	800(800)	880
ramos3	0	2187(2187)	2187
ran14x18.disj-8	252	252(252)	447
ran14x18	252	252(252)	284
ran16x16	256	256(256)	288
set1ch*	408	235(235)	423
set3-10	2501	176(176)	2481
set3-15	2501	176(176)	2537
set3-20	2501	176(176)	2537
seymour.disj-10	0	1022(987)	4800
seymour**	0	893(893)	4369
stockholm	9891	829(825)	21968
sts405	0	405(405)	27270
sts729	0	729(729)	88452
tanglegram1	0	32705(32705)	65152
tanglegram2	0	4058(4058)	7976
toll-like	0	2570(2570)	4038
zib54-UUE	4958	80(80)	1745

TABLE 4.1: Granular presolved instances from the MIPLIB libraries

of FRA-SOR) for problems where integer variables are not binary but are bounded by some larger boxes.

Moreover, we find that only one problem has non-integral values in the constraint matrix B at rows i where $\alpha_i = 0$ holds. Finally we report that determining the vector ω is generally an easy task as even an iterative implementation takes, for most problems, not longer than a couple of seconds. However, we stress once more that we can also simply set ω to $\bar{\omega}$, with

$$\bar{\omega}_i := \begin{cases} 1, & \alpha_i = 0, \beta_i \in \mathbb{Z}^m, \\ 0, & \text{else} \end{cases}$$

which computationally comes for free. In fact, for all presolved models from our test bed the "true" ω actually coincides with $\bar{\omega}$. This is also the case for almost all

unaltered problems, as we shall further discuss in Section 4.2.4.

4.2.2 Comparison with Gurobi for Granular Optimization Problems

To determine if granularity is exploited not only by the feasible rounding approaches but also by other commonly used heuristic techniques and to establish a proof of concept for our methods, we compare the run time of the feasible rounding approaches to the time which Gurobi needs to find the first feasible point when tuned to finding a feasible solution as fast as possible. We achieve this behavior by setting the parameter values of *SolutionLimit* and *MIPFocus* to one, where Gurobi recommends the latter for finding feasible solutions quickly. For a generalization of our results we experiment with the parameter *Heuristic* which determines the amount of time Gurobi spends in MIP heuristics.

As mentioned in the previous section, after the application of a presolving step we can simply use $\bar{\omega}$ for enlarging the relaxed feasible set. Therefore, the main effort in the computation of the data (Step 1 of our methods) is composed of (i) the calculation of $\|\beta\|_1$ and (ii) the identification of the rows i where $\beta_i \in \mathbb{Z}^m$ and $\alpha_i = 0$ hold. These two steps run within milliseconds for small and medium-sized problems and take at most 0.02 seconds time for a few larger problems in our Matlab implementation. Thus, the main effort of our method occurs in Step 2 in which we need to solve a linear optimization problem. As we do not wish to account for interfacing times with Gurobi, we henceforth report run times of this main effort (Step 2) only. This allows us to report the run time of the feasible rounding approaches by querying Gurobi's parameter *runtime*. We also use this parameter to report Gurobi's run time, and hence this yields a fair basis for a comparison.

Tables 4.2 and 4.3 report the run times and objective values of both feasible rounding approaches and Gurobi. Moreover, our results are summarized in Figure 4.1, where we report the number of problems for which the different approaches yield feasible points within a given time.

	FRA-SOR		FRA-SLOR		Gurobi	
name	time	objective	time	objective	time	objective
30_70_4.5_0.95_100	2.60	8878.00	0.31	19600.00	0.02	1033.00
50v-10	0.02	199236.75	0.00	1.420549e+07	0.00	31966.50
a1c1s1	0.03	23395.48	0.05	33278.38	0.08	19747.08
b2c1s1	0.06	77216.60	0.05	109582.11	0.19	63271.52
beasleyC3	0.00	4945.00	0.00	4945.00	0.05	922.00
bg512142	0.06	3.775710e+06	0.00	3.161667e+09	0.00	1.089073e+08
buildingenergy	14.94	44205.67	3.49	54541.38	0.06	1.624119e+07
core2536-691	1.88	10020.00	1.50	10554.00	0.00	892.00
core4872-1529	3.73	12826.00	2.57	13600.00	0.00	1934.00
cov1075	0.09	77.00	0.00	120.00	0.00	56.00
dfn-gwin-UUM	0.00	227208.00	0.00	488984.00	0.00	168924.00
dg012142	0.25	7.368098e+07	0.02	6.958340e+09	0.00	6.149438e+08
ex1010-pi	1.20	4641.00	0.02	11568.00	0.00	497.00
fast0507	0.52	18240.00	0.02	39500.00	0.00	321.00
fixnet6	0.00	93205.00	0.00	94980.01	0.00	22917.00
g200x740i	0.00	194475.00	0.02	197794.00	0.00	53393.00
ger50_17_trans	0.36	555975.25	0.05	657955.72	0.00	28622.18
germany50-DBM	0.92	610470.00	0.02	1.272850e+06	0.00	3.475250e+06
go19	0.05	297.00	0.00	361.00	0.00	95.00
iis-100-0-cov	0.03	100.00	0.00	100.00	0.00	35.00
iis-bupa-cov	0.17	107.00	0.02	337.00	0.00	48.00
iis-pima-cov	0.45	151.00	0.02	698.00	0.02	44.00
janos-us-DDM	0.00	1.508461e+06	0.00	3.437444e+06	0.00	5.628448e+06
k16x240	0.00	177473.00	0.00	185233.00	0.00	24175.00
m100n500k4r1	0.02	-11.00	0.00	0.00	0.00	-18.00
macrophage	0.06	1409.00	0.02	1409.00	0.00	581.00
manna81	0.02	-12869.00	0.00	0.00	0.00	-6954.00
mc11	0.02	128518.00	0.00	128518.00	0.11	13509.00
methanosarcina	1.20	7270.00	0.11	7270.00	0.00	5045.00
mik.250-1-100.1	0.00	0.00	0.00	2.000000e+07	0.00	446229.00
modglob	0.00	2.153798e+07	0.00	3.733806e+07	0.00	3.618051e+07
n15-3	18.81	89491.00	17.29	2.179122e+08	430.90	66291.00
n3-3	0.12	39030.00	0.05	6.960872e+07	0.72	22030.00
n3700	0.02	8.057113e+07	0.03	8.065223e+07	0.02	3.684658e+06
n3705	0.02	7.980784e+07	0.03	7.987856e+07	0.00	3.420830e+06
n370a	0.02	8.075517e+07	0.04	8.080776e+07	0.02	3.472063e+06
n4-3	0.03	18175.00	0.02	3.000411e+07	0.02	20115.00
n9-3	0.12	28425.00	0.05	4.680798e+07	0.52	19025.00
neos-1112782	0.02	2.477444e+13	0.00	2.477444e+13	0.00	2.248990e+12
neos-1112787	0.00	2.178675e+13	0.00	2.178675e+13	0.00	1.702341e+12
neos-1225589	0.00	9.819371e+10	0.00	9.819371e+10	0.00	3.866793e+09

TABLE 4.2: A comparison of the feasible rounding approaches and Gurobi with regard to time (seconds) and objective value on presolved models (I)

	FRA-SOR		FRA-SLOR		Gurobi	
name	time	objective	time	objective	time	objective
neos-1616732	0.05	197.00	0.00	200.00	0.00	173.00
neos-932816	0.05	5.723447e+06	0.00	6.728378e+06	0.02	513000.00
neos-933638	1.69	5.071322e+06	0.03	6.077942e+06	0.00	484000.00
neos-933966	1.09	5.071338e+06	0.02	6.078073e+06	0.00	484000.00
neos-934278	1.52	4.617128e+06	0.03	5.576040e+06	0.00	468000.00
neos15	0.00	145050.91	0.00	2.270029e+09	0.00	2.832237e+07
npmv07	2.23	1.049283e+11	1.05	5.832567e+11	0.05	4.608013e+11
ns4-pr3	0.11	38120.00	0.02	1.191060e+06	0.00	112045.00
ns4-pr9	0.11	36475.00	0.02	743380.00	0.00	94935.00
opm2-z10-s2	15.98	-1118.00	7.84	-3017.00	0.02	-8104.00
opm2-z11-s8	28.34	-1611.00	18.05	-2515.00	0.02	-9433.00
opm2-z12-s14	56.93	-1306.00	31.54	-3351.00	0.00	-11994.00
opm2-z12-s7	53.07	-1653.00	29.32	-4781.00	0.02	-12375.00
opm2-z7-s2	1.08	-1359.00	0.48	-955.00	0.00	-3515.00
p100x588b	0.00	448087.00	0.00	454481.00	0.00	94028.00
p6b	0.11	-4.00	0.00	0.00	0.00	-52.00
p80x400b	0.00	308006.00	0.00	313331.00	0.00	77458.00
pb-simp-nonunif	6.75	170.00	1.28	170.00	0.02	94.00
pp08a	0.00	18100.00	0.00	27519.09	0.00	13390.00
pp08aCUTS	0.00	20030.46	0.00	25880.21	0.00	15770.00
qiu	0.02	2603.42	0.02	4127.36	0.09	1120.97
queens-30	1.47	0.00	0.00	0.00	0.00	-29.00
r80x800	0.00	26891.00	0.00	40827.00	0.00	23943.00
ramos3	11.46	1077.00	0.00	2187.00	0.00	542.00
ran14x18.disj-8	0.02	42607.74	0.00	43141.00	0.00	7657.00
ran14x18	0.00	42659.02	0.02	43246.99	0.00	7657.00
ran16x16	0.00	42914.02	0.02	43727.02	0.00	7994.00
set1ch	0.00	169016.59	0.00	214781.71	0.00	133709.75
set3-10	0.03	1.993635e+06	0.02	3.192430e+06	0.00	5.086785e+06
set3-15	0.03	1.927219e+06	0.02	3.042305e+06	0.00	5.112993e+06
set3-20	0.05	1.917558e+06	0.02	2.912400e+06	0.00	4.396281e+06
seymour.disj-10	0.52	632.00	0.02	1111.00	0.00	362.00
seymour	1.91	648.00	0.00	1082.00	0.00	492.00
stockholm	1.09	785.00	7.54	838.00	17.57	150.00
sts405	0.06	405.00	0.03	405.00	0.00	357.00
sts729	0.47	729.00	0.14	729.00	0.00	665.00
tanglegram1	17.04	32576.00	0.11	32576.00	0.02	7505.00
tanglegram2	1.17	3989.00	0.02	3989.00	0.00	2022.00
toll-like	0.27	2055.00	0.02	2055.00	0.00	1134.00
zib54-UUE	0.28	2.404391e+07	0.02	2.404391e+07	0.00	2.404391e+07

TABLE 4.3: A comparison of the feasible rounding approaches and Gurobi with regard to time (seconds) and objective value on presolved models (II)

Firstly, notice that both feasible rounding approaches compute feasible points for almost all problems from our test bed within 15 seconds. Moreover, in half the cases (41/81) FRA-SLOR computes a feasible point at least as fast as Gurobi. Exceptions are the binary problems *opm2-z10-s2* to *opm2-z7-s2* where the involved linear auxiliary problems seem to be particularly hard to solve. For these problems Gurobi computes feasible points much faster.

Secondly, note that Gurobi yields feasible points within only one second for most granular optimization problems. In 40 cases Gurobi is even faster than FRA-SLOR, presumably due to the use of various primal heuristics. Interestingly, in 64 cases Gurobi even yields a feasible point with a better objective value than that of the point constructed by FRA-SOR, which is better in only 17 cases. Therefore, we may conclude that for many granular practical applications, standard software already

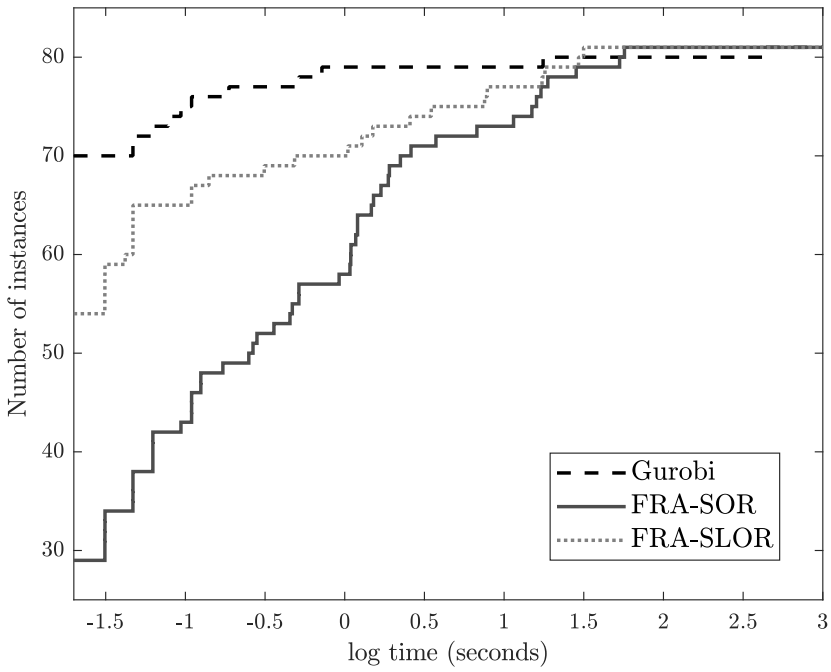


FIGURE 4.1: Number of instances for which feasible points are computed

computes feasible points quite quickly and that, in general, we cannot expect the objective value of FRA-SOR to outperform that of commonly used heuristic techniques.

However, this does not hold true for all problems, most notably for *n15-3*, *stockholm* and *buildingenergy*. For *n15-3* and *stockholm* it takes Gurobi much longer than both feasible rounding approaches to obtain a feasible solution. In fact, for *n15-3* it takes Gurobi 430.9 seconds to obtain the first feasible point, whereas FRA-SOR and FRA-SLOR terminate after only 18.81 seconds and 17.29 seconds, respectively.

Let us next consider the problem *buildingenergy* which, like *n15-3*, contains no binary variables. Here, Gurobi yields a feasible point within a fraction of a second, whereas the feasible rounding approaches run 14.94 and 3.49 seconds, respectively. However, the corresponding feasible point obtained by FRA-SOR has a significantly better objective value compared to that of Gurobi. In fact, for this particular problem (with the above mentioned parameters) it takes Gurobi more than 40 seconds until generating a feasible point with better objective value than that of FRA-SOR.

Aside from the problems *buildingenergy* and *n15-3* our test bed only contains 11 problems where at least half of the integer variables are not binary. We emphasize that for these problems FRA-SOR and Gurobi run within roughly the same time and that the feasible points obtained by our method actually yield better objective values in 7 (out of 11) cases.

We report that our results do not qualitatively change when we vary the parameter *Heuristic* from 0.05 (default) to 1. In particular, Gurobi *always* spends more than 390 seconds until generating the first feasible point for the problem *n15-3*.

Therefore, although for the majority of practical problems exploiting granularity does not seem to be necessary for quickly finding feasible points, at least for some instances the granularity concept provides significant savings in run time. Moreover,

for most practical problems the run time of both feasible rounding approaches exhibits the same order of magnitude as Gurobi. Finally we stress that the performance of FRA-SOR is promoted by the presence of non-binary integer variables, which are often even complicating for other approaches. Thus, applying FRA-SOR on problems containing relatively many non-binary integer variables, like it is the case for *buildingenergy* and *n15-3*, seems to be particularly promising.

In an additional experiment we evaluated if rounded feasible points with improved objective value can be found along the line segment connecting the point generated by FRA-SOR and the optimal point of the continuous relaxation. We systematically checked this using an integer line search technique similar to the ones presented in [12, 39]. Apart from 16 cases, this small effort always yielded an (often significant) improvement. We remark that other methods of locally improving solutions may work even better. Hence FRA-SOR may also be beneficial to generate start solutions of local search heuristics.

One might argue that a-priori it is not known whether a general mixed-integer linear optimization problem is granular. However, the feasible rounding approaches *always run efficiently*, either terminating with a feasible point or reporting the nongranularity of the problem. Therefore, we suggest to take both feasible rounding approaches into consideration when choosing a method to come up with an initial feasible point for mixed-integer linear optimization problems.

4.2.3 Comparison with Gurobi for Nongranular Optimization Problems

As mentioned before, FRA-SLOR can also yield a feasible point for optimization problems which are not granular. Our computational study shows that this is the case for 10 out of 70 nongranular optimization problems. These problems are listed in Table 4.4 where we also present the run time for the computation and the objective value of the feasible points obtained by FRA-SLOR and Gurobi, as well as the objective z^f of the problem P^f.

				FRA-SLOR			Gurobi	
name	n	m (b)	p	time	z^f	objective	time	objective
neos-1171692	819	819(819)	4239	0.03	0.00008	0.00	0.00	0.00
neos-1171737	1170	1170(1170)	4179	0.03	2.5	0.00	0.00	0.00
neos-1311124	546	546(546)	1643	0.00	1.75	0.00	0.00	0.00
neos-1426635	260	260(260)	796	0.00	1.75	0.00	0.00	0.00
neos-1426662	416	416(416)	1914	0.00	0.4	0.00	0.00	0.00
neos-1436709	338	338(338)	1417	0.02	0.00007	0.00	0.00	0.00
neos-1440460	234	234(234)	989	0.00	0.00008	0.00	0.00	0.00
neos-1442119	364	364(364)	1524	0.02	0.00008	0.00	0.00	0.00
neos-1442657	312	312(312)	1310	0.00	0.00008	0.00	0.00	0.00
neos13	12	1815(1815)	20852	0.55	21.065	0.00	0.02	0.00

TABLE 4.4: Nongranular instances for which FRA-SLOR yields a feasible point

Note that, in contrast to Section 4.2.2, we no longer have a solid theoretical foundation which explains why FRA-SLOR is able to compute a feasible point. However, the optimal value z^f might be an indicator, as well as the norm of each vector β_i, where $\alpha_i^\intercal x^f + \beta_i^\intercal y^f = z^f$ holds. In fact, as shown in Table 4.4, for most of the problems z^f is quite close to zero. Yet we stress that only $z^f = 0$ guarantees the output of a feasible point.

Concerning the run time and objective value, the performance of FRA-SLOR is quite similar to that of Gurobi. One may wonder why the objective value is zero for both approaches and all optimization problems. In fact, for the problems *neos-1171692* to *neos-1442657* Gurobi and FRA-SLOR compute the (feasible) zero vector, and for the problem *neos13* the objective vector only contains 12 non-zero entries, and the corresponding entries of the points obtained by FRA-SLOR and Gurobi are zero.

4.2.4 Influence of Presolving Techniques

In Chapter 3 we illustrated that different modeling techniques may generally influence granularity. So far we have tested the feasible rounding approaches on presolved models of optimization problems from the MIPLIB libraries. By also examining unaltered models, we shall see next that presolving techniques affect granularity in the same manner.

Let us briefly discuss some rather apparent effects that occur when a presolving step is applied. Firstly, the elimination of rows in the constraint matrix promotes granularity because, as Example 3.2.6 shows, redundant constraints can become active in the enlarged inner parallel set. Secondly, equality constraints may be eliminated in a presolving step via simple or regular probing. Thirdly, constraint tightening techniques appear to have a negative influence on granularity. However, the enlargement step in the construction of the enlarged inner parallel set actually reverses many of them so that consequently they do not influence granularity. Moreover, some constraint coefficients may become integral which further promotes granularity, as it enables the enlargement of the corresponding constraint. Clearly these few examples are not encompassing and we cannot use them to certainly predict the influence of a presolving step on granularity.

However, our first glance suggests a granularity promoting influence of a presolving step: Indeed, for unaltered models, we can test our methods on only 134 (instead of 151) problems, as for the remaining 17 problems the presolving step eliminated equality constraints which were posed on integral variables. Together with the other effects discussed above this leads to the fact that six problems, *beasleyC3*, *core2536-691*, *core4872-1529*, *dg012142*, *go19* and *pb-simp-nonunif*, are only granular when Gurobi's presolving step is applied. On the other hand, we report two additional granular unaltered problems, *mas74* and *mas76*, for which only Gurobi's presolving step prevented granularity. In summary we obtain that 77 out of 134 unaltered optimization problems from the MIPLIB libraries are granular.

Quite similarly to the presolved case, the nonzero entries of ω only contain ones for almost all granular optimization problems. The only exception is the problem *bg512142*. Here, some of the (only 12) nonzero entries of ω obtain a value of 119. In this case an exact computation of ω is crucial while for all other problems our results also hold with the simple choice $\omega = \bar{\omega}$. Finally, in contrast to the presolved models, we find that actually eight problems possess non-integral values in the constraint matrix B at rows where $\alpha_i = 0$ holds. For these models we did not enlarge the corresponding constraints and hence the presolving step indeed promoted granularity.

We report run times and objective values for the unaltered problems in Tables A.1 and A.2 in the appendix. We find that the enlarged inner parallel set changes for many optimization problems when a presolving step is applied, as we often obtain slightly different objective values using the feasible rounding approaches on the unaltered models (compared to the presolved models). Hence our results for unaltered

models slightly differ from those reported in Tables 4.2 and 4.3, but qualitatively remain unaffected.

4.3 Computational Study in the Nonlinear Case

Our next computational study investigates the applicability of the feasible rounding approach FRA-SOR to mixed-integer quadratically constrained quadratic problems (MIQCQPs) from practice. With our study, we primarily intend to provide a proof of concept of the method also in the nonlinear case. In particularly, we wish to shed light on the following questions:

1. Under what circumstances is the computation of Lipschitz constants feasible for MIQCQPs from practice?
2. Is pseudo-granularity a characteristic that can be expected in practical applications?
3. How "good" is the objective value of the feasible points computed by FRA-SOR for pseudo-granular problems?
4. Are these results sensitive to variations of the enlargement values σ^ℓ and σ^u for the box constraints?

To address these questions, we collected MIQCQPs from the MINLPLib [17]. We looked for problems which are separable in x and y, that is, all quadratic constraints are posed as

$$x^\intercal Q_x x + y^\intercal Q_y y + \beta_x^\intercal x + \beta_y^\intercal y \leqslant q,$$

with an (n,n)-matrix Q_x, an (m,m)-matrix Q_y, $\beta_x \in \mathbb{R}^n$, $\beta_y \in \mathbb{R}^m$ and $q \in \mathbb{R}$.

This not only allowed us to compute the Lipschitz constants using the problem *MILP* from Example 2.3.7, but also to quickly obtain upper bounds for the Big-M values using (2.5) and (2.6).

We implemented FRA-SOR in Python 3.6 using the Pyomo framework [36]. We used Gurobi 8 for solving the auxiliary MILP and IPOPT [73] implemented in the Coin infrastructure [21] for solving the auxiliary problem P_ρ^{ob}. All tests were run on a desktop computer with an Intel i7 processor with 8 cores à 3.6 GHz and 32 GB RAM.

We computed the greatest common divisor ω of β_y (and all entries of Q_y) for constraints where $Q_x = 0$ and $\beta_x = 0$ held for obtaining the enlargement parameters σ^b (τ) for the corresponding (non)linear constraint. In fact, in the (non)linear case, we set $\sigma^b(\tau)$ to $\lfloor q \rfloor_\omega + (1 - 10^{-4})\omega$.

For our main analysis, we set $\sigma^\ell = \sigma^u = \delta e$, with $\delta = 1 - 10^{-4}$, but we shall make further remarks on the effects of changing these values in Section 4.3.3.

Our test bed contains 210 problems. Let us distinguish three cases, which vary in their significance for our analysis. First, for 68 problems integer variables only appear in linear constraints, that is, all nonlinear constraints are posed on continuous variables only. For these problems, FRA-SOR basically reduces to the approach presented in the polyhedral case, the only novelty being that nonlinear constraints on x may be incorporated unaltered in the auxiliary optimization problem.

Secondly, in 10 instances integer variables appear in nonlinear constraints, but only in a linear fashion, that is $Q_x \neq 0$, $Q_y = 0$, $\beta_y \neq 0$. For such problems, we could compute the Lipschitz constant in accordance with Example 2.3.6, which yields a first major generalization compared to the purely polyhedral case discussed in the

previous Section. Notice that for both cases we can clearly answer Question 1 affirmatively, as the computation of $\|\beta\|_1$ requires a relatively small number of operations.

Finally, the third and most interesting case leaves 132 instances, which contain nonlinear constraints that have a nonlinear part in y. These problems are the most challenging ones for the construction of the pseudo-granularity parameters and thus also for the applicability of our approach.

4.3.1 Computation of Lipschitz Constants

In order to answer Question 1 comprehensively, let us start our analysis by investigating the task of computing all Lipschitz constants. In fact, we were able to compute all Lipschitz constants for 183 out of the 210 problems within a time limit of 30 minutes. We further report that, if successful, this task needed more than 5 seconds for only 10 optimization problems. We list these problems in Table 4.5a next to 10 (out of 27) exemplary problems where we were not able to compute all Lipschitz constants within 30 minutes (Table 4.5b).

	m	p	time
edgecross10-040	90	480	26.10
edgecross10-050	90	480	51.44
edgecross10-070	90	480	1095.28
edgecross14-176	182	1456	240.80
sporttournament18	153	0	10.12
sporttournament20	190	0	16.81
sporttournament22	231	0	22.10
sporttournament24	276	0	188.42
sporttournament26	325	0	228.91
sporttournament28	378	0	470.62

(a) Lipschitz constant computable

	m	p
edgecross14-058	182	1456
edgecross20-040	380	4560
edgecross22-048	462	6160
edgecross24-057	552	8096
faclay20h	190	2280
faclay30h	435	8120
faclay33	528	10912
faclay35	595	13090
sporttournament30	435	0
sporttournament50	1225	0

(b) Lipschitz constant not computable

TABLE 4.5: Instances where the computation of Lipschitz constants is especially difficult

We remark that all problems from Table 4.5 share a characteristic which makes them particularly suitable for our analysis. All problems are epigraph reformulations, in which we minimize an additional variable α, adding the constraint $f(x, y) \leqslant \alpha$ to the "original" model. Moreover, in all instances $f(x, y) \leqslant \alpha$ is the *only* nonlinear constraint, *almost all* variables appear in this constraint, and the Matrix Q_y is therefore comparably dense (details on the sparsity can be obtained from the MINLPLib website [54]). This means that the auxiliary MILP is much more difficult to solve compared to problems where nonlinear constraints contain only a few integer variables with nonzero coefficients.

In Table 4.5 we also report the number of integer variables (m) and the number of linear constraints (p) for all problems. Table 4.5a additionally lists the time needed to solve the auxiliary MILP in seconds.

Observe that the number of variables occurring in a constraint seems to influence our ability to compute the Lipschitz constant, as well as the number of linear constraints. However, a closer look actually reveals that the sparsity of Q_y is more important. Indeed, we need significantly more time to solve the auxiliary MILP for *edgecross10-070* compared to *edgecross14-176*, although the latter has more linear constraints and more integer variables. Yet, the matrix Q_y of *edgecross10-070* is considerably more dense, which appears to be the reason for the observed behavior.

Finally, we remark that the only real difficulties in the computation of the Lipschitz constant occurred with instances where a nonlinear constraint $f(x,y) \leqslant \alpha$ resulted from an epigraph reformulation. These problems were perfectly suitable for analyzing the behavior of the time needed to solve *MILP* for different problem sizes. However, we stress that a constraint coming from an epigraph reformulation can be satisfied even without the computation of a Lipschitz constant. All we need to do is to apply a post processing step, which sets the epigraph variable α to $f(\breve{x}^{ob}, \breve{y}^{ob})$, the objective value generated by FRA-SOR.

4.3.2 Pseudo-Granular Problems and Quality of the Generated Feasible Points

In this section we investigate Questions 2 and 3, that is, we look for pseudo-granular instances and evaluate the quality of the feasible points generated by FRA-SOR for such problems. We use three different methods implemented in Bonmin [13] for our comparison. First, the outer-approximation based branch-and-cut algorithm (B-Hyb), which is the recommended algorithmic choice for solving *convex* MINLPs using Bonmin. Secondly, the branch-and-bound algorithm (B-BB), which is Bonmin's recommendation for nonconvex MINLPs and, finally, the iterated feasibility pump (B-iFP), which is suggested for quickly finding good solutions to very hard convex MINLPs [16].

We set the time limit for FRA-SOR to 30 minutes, including the time needed to solve the auxiliary MILP as well as the auxiliary NLP. We also used 30 minutes as a time limit for all competing methods.

We report that we were able to use pseudo-granularity for the computation of feasible points for 52 out of 183 problems. This indicates that pseudo-granularity is a characteristic that can be expected to hold for a relevant size of practical applications.

Before we proceed with a comparison that evaluates the quality of the generated feasible points, let us initially remark that several problems were an equivalent epigraph reformulation. Note that the output $(\breve{x}^{ob}, \breve{y}^{ob}, \breve{\alpha}^{ob})$ of FRA-SOR applied to an epigraph problem may not be expected to satisfy $f(\breve{x}^{ob}, \breve{y}^{ob}) = \breve{\alpha}^{ob}$ exactly, but $\breve{\alpha}^{ob}$ will usually be a strict upper bound for $f(\breve{x}^{ob}, \breve{y}^{ob})$. However, for the original MIQCQP the objective value $f(\breve{x}^{ob}, \breve{y}^{ob})$ then provides more useful information than the auxiliary value $\breve{\alpha}^{ob}$, so that it makes sense to update $\breve{\alpha}^{ob}$ to $f(\breve{x}^{ob}, \breve{y}^{ob})$ in a post processing step after applying the FRA-SOR to epigraph reformulated problems.

Moreover, 13 of the 52 pseudo-granular problems are instances of the problems *sporttournament* and *autocorr_bern*. Yet, these problems actually correspond to an epigraph reformulation of an unconstrained problem. Note that for such problems pseudo-granularity is an apparent characteristic and the point obtained by FRA-SOR is actually not very interesting. In fact, it is easily verified that in such cases the solution of FRA-SOR corresponds to the rounding of the solution of the original unconstrained continuously relaxed problem. Therefore, we excluded these problems from further comparison.

To obtain comparability for the remaining 39 pseudo-granular problems, we investigated the time which B-Hyb, B-BB and B-iFP need for computing a point that is at least as good as the one provided by FRA-SOR. To do so, we set the parameter *"cutoff"* to $f(\breve{x}^{ob}, \breve{y}^{ob})$, the objective value obtained by the FRA-SOR. This implicitly added the constraint $f(x,y) \leqslant f(\breve{x}^{ob}, \breve{y}^{ob})$ to the model, instructing the other methods to only search for feasible points of the prescribed quality.

We wish to point out the slightly biased nature of this experiment but also remark that our ambition is *not* to benchmark FRA-SOR against all other methods. Rather, we wish to find out if the objective values of the points generated by FRA-SOR provide valuable information in the sense that they can potentially be used to help speeding up exact algorithms.

The detailed results are shown in Table 4.6, where we also list a summary of the problem data. Problems where all appearing functions are proven to be convex are marked with an asterisk, and the columns of the table read as follows

- **variables**: overall number of variables, number of integer variables, number of binary variables,
- **constraints**: overall number of constraints, number of nonlinear constraints, number of nonlinear constraints where integer variables appear, number of nonlinear constraints where integer variables appear in a nonlinear fashion,
- **objective**: objective value obtained by FRA-SOR and cutoff value for the other methods,
- **time (**$MILP$**)**: cumulative time in seconds needed to solve all auxiliary MILPs for the computation of the Lipschitz constants,
- **time (**P_ρ^{ob}**)**: time needed to compute an optimal point of the purely continuous nonlinear problem P_ρ^{ob},
- **time BB/B-Hyb/B-iFP**: time needed for BB/B-Hyb/B-iFP to compute a feasible point that meets the cutoff value in the column "objective".

	variables	constraints	objective	time				
				MILP	P_ρ^{ob}	B-BB	B-Hyb	B-iFP
cvxnonsep_normcon20r*	(40, 10, 0)	(21, 20, 10, 10)	-14.65	1.37	0.05	0.09	0.20	0.14
cvxnonsep_normcon30r*	(60, 15, 0)	(31, 30, 15, 15)	-14.80	1.73	0.02	0.14	0.08	0.06
ex1223a*	(7, 4, 4)	(9, 4, 4, 0)	6.07	0.00	0.05	0.03	0.09	0.08
ex4*	(36, 25, 25)	(30, 25, 25, 0)	-6.70	0.00	0.03	0.08	0.81	0.83
genpooling_lee1	(49, 9, 9)	(82, 20, 0, 0)	-4309.83	0.00	0.70	-	-	-
genpooling_lee2	(53, 9, 9)	(92, 30, 0, 0)	-3849.24	0.00	0.37	0.45	0.42	0.64
genpooling_meyer10	(394, 187, 187)	(423, 33, 0, 0)	5129659.13	0.00	0.72	5.67	5.36	2.81
genpooling_meyer15	(734, 352, 352)	(768, 48, 0, 0)	6581050.03	0.00	1.35	57.19	6.92	4.28
ndcc12	(644, 46, 46)	(237, 46, 0, 0)	108.11	0.00	0.14	0.30	1.77	1.69
ndcc13	(630, 42, 42)	(254, 42, 0, 0)	107.57	0.00	0.23	2.23	6.75	6.14
ndcc14	(864, 54, 54)	(305, 54, 0, 0)	143.31	0.00	0.59	0.80	27.48	-
ndcc15	(680, 40, 40)	(306, 40, 0, 0)	102.42	0.00	0.30	0.33	4.53	3.56
ndcc16	(1080, 60, 60)	(377, 60, 0, 0)	145.89	0.00	0.28	5.40	48.21	-
nous1	(50, 2, 2)	(43, 28, 0, 0)	1.57	0.00	0.08	-	-	-
nous2	(50, 2, 2)	(43, 28, 0, 0)	0.63	0.00	0.36	0.48	0.33	0.36
ringpack_10_1	(70, 50, 50)	(385, 330, 330, 0)	-4.17	0.00	0.22	0.12	-	-
ringpack_10_2	(80, 60, 60)	(475, 420, 420, 0)	0.00	0.00	0.44	0.14	2.38	-
ringpack_20_1	(215, 175, 175)	(2547, 2337, 2337, 0)	-4.17	0.00	161.93	0.61	-	-
ringpack_20_2	(235, 195, 195)	(2927, 2717, 2717, 0)	0.00	0.00	401.47	623.76	-	-
ringpack_30_1	(433, 373, 373)	(7898, 7433, 7433, 0)	-6.26	0.00	400.71	2.25	29.98	-
smallinvDAXr1b150-165*	(31, 30, 0)	(4, 1, 1, 1)	100.66	0.14	0.05	0.20	11.79	-
smallinvDAXr1b200-220*	(31, 30, 0)	(4, 1, 1, 1)	175.19	0.12	0.05	0.16	11.61	-
smallinvDAXr2b150-165*	(31, 30, 0)	(4, 1, 1, 1)	100.66	0.12	0.06	0.25	12.01	-
smallinvDAXr2b200-220*	(31, 30, 0)	(4, 1, 1, 1)	175.19	0.14	0.05	0.12	12.34	-
smallinvDAXr3b150-165*	(31, 30, 0)	(4, 1, 1, 1)	100.66	0.14	0.03	0.23	11.70	-
smallinvDAXr3b200-220*	(31, 30, 0)	(4, 1, 1, 1)	175.19	0.12	0.06	0.13	12.11	-
smallinvDAXr4b150-165*	(31, 30, 0)	(4, 1, 1, 1)	100.66	0.12	0.05	0.20	12.79	-
smallinvDAXr4b200-220*	(31, 30, 0)	(4, 1, 1, 1)	175.19	0.15	0.05	0.12	11.84	-
smallinvDAXr5b150-165*	(31, 30, 0)	(4, 1, 1, 1)	100.66	0.12	0.05	0.23	12.12	-
smallinvDAXr5b200-220*	(31, 30, 0)	(4, 1, 1, 1)	175.19	0.12	0.05	0.16	12.73	-
sonet17v4	(136, 136, 136)	(2057, 17, 17, 17)	1816146.00	2.35	6.37	9.34	30.51	-

Continued on next page

	variables	constraints	objective	time				
				MILP	P_ρ^{ob}	B-BB	B-Hyb	B-iFP
sonet18v6	(153, 153, 153)	(2466, 18, 18, 18)	6925440.00	2.56	19.75	36.06	77.93	13.21
sonet19v5	(171, 171, 171)	(2926, 19, 19, 19)	4806792.00	2.75	20.45	33.84	63.97	-
sonet20v6	(190, 190, 190)	(3440, 20, 20, 20)	9245610.00	3.03	36.54	182.47	119.61	-
sonet21v6	(210, 210, 210)	(4011, 21, 21, 21)	10674610.00	3.32	63.06	38.74	12.63	-
sonet22v4	(231, 231, 231)	(4642, 22, 22, 22)	3408192.00	3.60	122.41	48.88	13.43	-
sonet23v6	(253, 253, 253)	(5336, 23, 23, 23)	12764340.00	3.91	161.96	244.36	264.25	-
sonet24v2	(276, 276, 276)	(6096, 24, 24, 24)	21512599.00	4.26	636.22	-	874.83	-
supplychain	(27, 3, 3)	(30, 6, 0, 0)	2288.16	0.00	0.12	0.09	0.20	0.08

TABLE 4.6: Comparison of FRA-SOR with three algorithms implemented in Bonmin for 39 instances from the MINLPLib

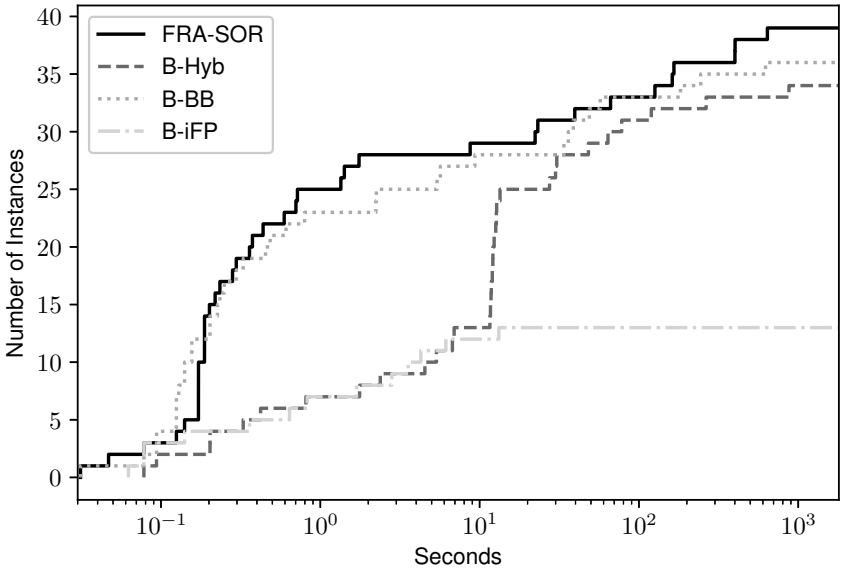


FIGURE 4.2: Number of instances solved to the cutoff value given by FRA-SOR

Table 4.6 indicates that, similar to the findings for the linear case in the previous section, the occurrence of binary variables does not prohibit pseudo-granularity for mixed-integer nonlinear optimization problems. Moreover, in several cases we can show pseudo-granularity for problems with a large number of linear and nonlinear constraints and, in many cases, the explicit use of pseudo-granularity is indeed advantageous. In fact, in 22 out of 39 instances, all other methods need additional time to compute a feasible point of the same quality.

In some cases, pseudo-granularity seems to be particularly useful. For example, for several instances of the problem *sonet* it takes all other methods significantly longer to compute a point of the prescribed quality as compared to using FRA-SOR.

On the other hand, in 15 cases B-BB finds a point of similar quality in less time. In these cases, P_ρ^{ob} often seems to be quite difficult to solve to optimality, like it is the case for *ringpack_30_1*. Notice that these results could potentially be improved by not computing an optimal point of P_ρ^{ob}, but terminating earlier with some good feasible point of P_ρ^{ob}.

A summary of our comparison is visualized in Figure 4.2, where we plot the number of instances solved to the quality prescribed by FRA-SOR over time. This figure reveals that FRA-SOR terminates for many instances already within one second, whereas all other methods need significantly more time to compute points of similar quality for the same share of problems. This is especially true for B-Hyb and B-iFP. Hence, our results indicates that, if the computation of the Lipschitz constant is quick, using pseudo-granularity explicitly can be very beneficial.

We remark once more that our benchmark is performed only on pseudo-granular instances and that the cutoff-value is given by the point obtained by FRA-SOR rather than by some external value. Nonetheless, Figure 4.2 reveals that the application of FRA-SOR provides valuable information *if a pseudo-granular structure is present*, in the sense that we are then able to construct feasible points of a quality which is not easily obtainable using other methods.

4.3.3 Effects of Different Enlargement Parameters for Box Constraints

Let us next investigate how the above results depend on different values of σ^ℓ and σ^u. In fact, with $\sigma^\ell = \sigma^u = \delta e$, we repeated parts of the above analysis setting δ to 0.5 and 0.75, and observed three different patterns.

First, smaller values of δ made it easier for Gurobi to solve the auxiliary MILP. Table 4.7 illustrates that, with $\delta = 0.5$, we were able to compute Lipschitz constants for constraints in much higher dimensions compared to $\delta = 1 - 10^{-4}$. This seems to indicate that using lower values for δ can be beneficial.

	m	p	time
edgecross14-058	182	1456	520.11
edgecross20-040	380	4560	1.98
edgecross22-048	462	6160	3.36
edgecross24-057	552	8096	14.25
faclay20h	190	2280	148.15
sporttournament30	435	0	10.40
sporttournament32	496	0	14.11

	m	p	time
sporttournament34	561	0	87.51
sporttournament36	630	0	154.54
sporttournament38	703	0	283.66
sporttournament40	780	0	363.60
sporttournament42	861	0	1140.25
sporttournament46	1035	0	946.82
sporttournament48	1128	0	783.33

TABLE 4.7: Instances where the computation of Lipschitz constants is possible for $\delta = 0.5$, but not for $\delta = 1 - 10^{-4}$

Yet, our second observation was that decreasing values of δ actually made it less likely to obtain a nonempty set $\widetilde{T}_\rho^-$. In fact, while for $\delta = 1 - 10^{-4}$ this set was nonempty for 52 out of 183 tested instances, for $\delta = 0.75$ this was only the case for 41 out of 184 instances with available Lipschitz constants. Setting $\delta = 0.5$ resulted in 42 problems with a nonempty set $\widetilde{T}_\rho^-$ out of 197 instances with available Lipschitz constants. Recall that pseudo-granularity for instances of the problem *sporttournament* is actually trivial, so that this indeed shows a decreasing number of pseudo-granular problems for decreasing values of δ.

Finally, different values of δ resulted in different feasible points. Setting $\delta = 0.75$, we were able to compute a feasible point for 27 of the 39 instances presented in Section 4.3.2. Out of these, the objective value of the obtained feasible point was similar to that presented in Section 4.3.2 in 6, better in 12, and worse in 9 cases.

Let us conclude this section with a short summary. First, the computation of Lipschitz constants is possible for almost all separable MIQCQPs from practice. Secondly, we were able to use pseudo-granularity for the computation of feasible points for roughly one out of four tested problems. This indicates that pseudo-granularity may be expected in various practical applications. Thirdly, if pseudo-granularity is present, then its explicit use is beneficial in a majority of cases.

Finally, we wish to remark that applying the presented ideas in combination with other primal heuristics might greatly enhance the presented results. For instance, postprocessing techniques like local search heuristics could be useful to improve the quality of the generated feasible points.

4.4 Conclusions and Outlook

In this chapter, we introduced a feasibility test and a feasible rounding approach for mixed-integer (nonlinear) optimization problems. In their basic form they make use of a structural property called granularity which, for the nonlinear case, we extended to the algorithmically more attractive concept of pseudo-granularity.

Indeed, tests on a subset of instances from the MIPLIB and the MINLPLib illustrate that granularity in the linear setting, and pseudo-granularity in the nonlinear setting may be expected in various practical applications. Furthermore, our results indicate that exploiting pseudo-granularity explicitly by applying the proposed methods can be beneficial.

In comparison to mixed-integer nonlinear optimization problems, mixed-integer linear optimization problems are very well studied. Hence it is no surprise that the feasible rounding approaches seem to add more value in the the nonlinear case, even though the possibility to explicitly compute inner parallel sets in the linear case makes the method particularly suitable for these problems.

We stressed that feasible rounding approaches need not to be viewed as standalone concepts and that the combination with other primal heuristics might greatly improve their effectiveness. Moreover, the sizes of the Lipschitz constants are crucial for the applicability of the presented approaches, and calculating at least good upper bounds for them is of major importance. In special cases, for example for functions in the description of the feasible set being separable in x and y as well as linear in y, even the precise Lipschitz moduli can be computed. This makes these problems especially suitable for the feasibility tests and feasible rounding approaches, particularly if used as a standalone concept.

If, on the other hand, the constraints do not fulfill such special properties, using single, global Lipschitz constants may be insufficient to ensure pseudo-granularity. Here, dividing the feasible set into multiple boxes, each with its own *local* Lipschitz constant, increases the chance of finding a good feasible point. Therefore, combining a feasible rounding approach with any concept like branch-and-bound, where optimization problems have to be solved over multiple boxes of decreasing size, seems to be particularly promising. We shall partly come back to these topics and investigate them more fully in Chapter 7.

Chapter 5

Inner Parallel Cuts

In this chapter we discuss mixed-integer convex optimization problems (MICPs) and develop a specialized cutting plane method that uses inner parallel sets as its basis. It is structured as follows.

In Section 5.1 we outline the special role of MICP, discuss some basic solution techniques and thus embed the ideas introduced in this chapter in the literature. Section 5.2 revisits the concept of granularity and dwells on the difference between the linear and the nonlinear case as a central motivation for the development of a cutting plane method that uses inner parallel sets as its basis.

Subsequently, Section 5.3 presents an inner parallel cutting plane method (IPCP), analyzes its properties with regard to convergence and discusses ways in which it can be incorporated into state-of-the-art methods that aim to solve MICPs. In Section 5.4 we provide worst case bounds on the objective values of the points obtained by this approach. Its applicability is studied in computational experiments on optimization problems from the literature in Section 5.5, before Section 5.6 concludes this chapter with some final remarks.

5.1 The Special Role of Mixed-Integer Convex Optimization

Mixed-Integer convex optimization problems occur in a variety of optimal decision problems from science, engineering, and economic applications (cf. the application matrix provided in [10]). This modeling framework allows us to accurately account for the fact that causal relations are often nonlinear in practice. Equally important, its practical complexity is just at the margin in the sense that we are able to compute optimal solutions to decision problems of relevant sizes. This makes a contribution in this field particularly appealing.

Leveraging on the development of powerful MILP-software which we have outlined in Chapter 1, one fruitful approach for solving MICPs is *outer approximation* (OA) [25, 28], where convex constraints are relaxed by a finite number of half-spaces. Based on the same principle, cutting-plane methods have been presented, including the use of extended cutting planes [74] and extended supporting hyperplanes [48]. In all these approaches, solving MICPs is reduced to solving a finite number of MILPs (and continuous nonlinear optimization problems) for which effective solvers exist.

For OA-based approaches to work well, convex constraints need to be well approximated by a preferably small number of half-spaces. While this can sometimes be achieved, e.g. by constructing the approximation in a higher dimensional space [38, 51], clearly good half-spaces that tightly outer-approximate convex constraints of general MICPs are crucially important for solving MICPs.

A second important aspect for solving MICPs is the availability of good feasible points. As already mentioned earlier, they help to speed up convergence by providing an upper bound for the objective value - especially in hybrid approaches that combine outer approximation with branch and bound concepts. Moreover, due to the notorious difficulty of solving the latter, the time needed for convergence of exact processes might sometimes not be practical. In such cases, a feasible point can be considered a solution, although its objective value might not be optimal. Recall that several existing ideas for the construction of feasible points were already outlined in Chapter 2.

In this chapter, we suggest an inner parallel cutting plane method (IPCP) that neatly combines these two important aspects, i.e. quickly constructing a feasible point and generating valid cutting planes for OA-based methods. Remarkably, it only needs to solve linear optimization problems (LPs) which ensures fast convergence for practical applications. The proposed method is inspired by feasible rounding approaches for MILPs introduced in the previous chapters and Kelley's (extended) cutting plane method [44, 74]. Before we introduce the method formally, let us next motivate its development by revisiting the advantages of the granularity concept in the linear compared to the nonlinear case.

5.2 Setting and Motivation

In this chapter we consider mixed-integer convex optimization problems of the form

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

with $c \in \mathbb{R}^n$ and $d \in \mathbb{R}^m$ and real-valued convex constraint functions g_i, $i \in I$, defined on $\mathbb{R}^n \times \mathbb{R}^m$. Note that the assumption of a linear objective function is not restrictive. In fact, in the case of a nonlinear objective function f, we can minimize an additional variable $z \in \mathbb{R}$ with the additional constraint $f(x,y) \leqslant z$ which fits the model $MICP$ (cf. Section 5.3.4 for further discussion).

In this chapter, we write the purely continuous relaxation of the feasible set M of $MICP$ as

$$\widehat{M} = D^0 \cap G,$$

where the set $D^0 := D$ includes all linear and the set G all nonlinear constraint functions. Here we replace set D by D^0 to be able to capture the iteratively changing polyhedral outer approximation of the relaxed feasible set. To guarantee convergence of the method, in this chapter we impose the assumption that the set described by the linear restrictions D^0 is bounded.

We will make frequent use of the monotonicity of inner parallel sets and their compatibility with intersections. To be more precise, for $S_1 \subseteq S_2 \subseteq (\mathbb{R}^n \times \mathbb{R}^m)$ we have $S_1^- \subseteq S_2^-$ and $(S_1 \cap S_2)^- = S_1^- \cap S_2^-$.

We specify the method for the enlarged inner parallel set $(D_\delta^0)^-$ from Equation (3.7) but stress that also other enlargement techniques are possible. Using this technique for some fixed $\delta \in [0,1)$, a problem is granular if the enlarged inner parallel set

$$\widehat{M}_\delta^- := (D_\delta^0)^- \cap G$$

is nonempty.

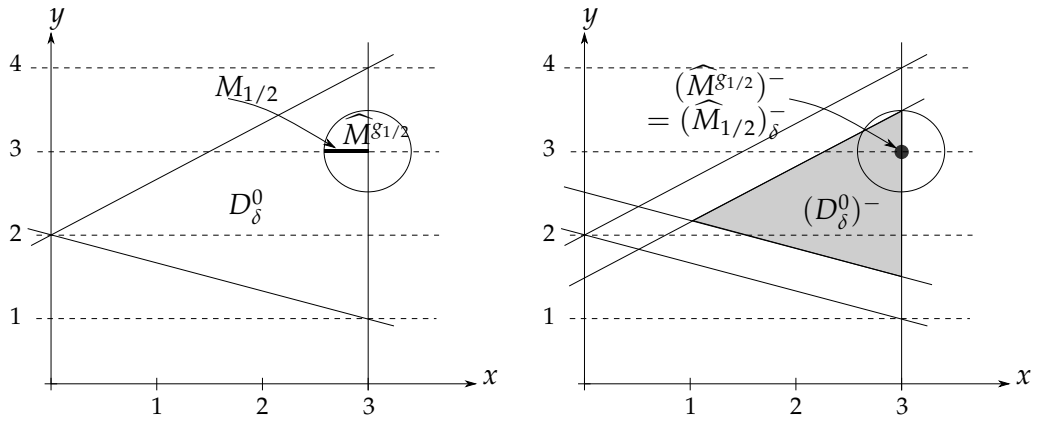


FIGURE 5.1: Construction of the enlarged inner parallel set for an *MICP*

Remark 5.2.1. *The developments in this section would be equally possible after applying an enlargement step to the nonlinear constraints of the set G as presented in Section 3.3. Yet, we propose not to enlarge nonlinear constraints in this context mainly because this will allow* deeper *cutting planes and thus a faster convergence of the method we present.*

The developments in this chapter only need as a foundation the concept of granularity (and not that of pseudo-granularity). This is because, instead of computing approximations of the inner parallel set via Lipschitz constants, we will be able to work with polyhedral approximations directly for which a closed-form expression is available. Let us next elaborate this more fully.

Recall that, while consistency of T_ρ^- (cf. Equation (3.9)) for some ρ entailed granularity of the underlying MICP, there are granular MICPs with empty sets T_ρ^- for any $\rho \in R$. We illustrate this by the following example which we shall also revisit in Section 5.3.1.

Example 5.2.2. *Let us consider the two dimensional mixed-integer set*

$$M_r := \{(x,y) \in \mathbb{R} \times \mathbb{Z} |\ (x-3)^2 + (y-3)^2 \leqslant r^2,\ -\tfrac{2}{3}x + y \leqslant 2,\ \tfrac{1}{3}x + y \geqslant 2,\ x \leqslant 3\}$$

with $r \geqslant 0$, which is illustrated on the left-hand side of Figure 5.1 for $r = 1/2$. In this notation we may write $\widehat{M}_r = D^0 \cap G_r$ with

$$D^0 := \{(x,y) \in \mathbb{R}^2 |\ -\tfrac{2}{3}x + y \leqslant 2,\ \tfrac{1}{3}x + y \geqslant 2,\ x \leqslant 3\}$$

and $g_r(x,y) := (x-3)^2 + (y-3)^2 - r^2$. Observe that an enlargement of D^0 and G_r is not possible, as all constraint functions contain continuous variables with nonzero coefficients. Consequently, all sets D^0_δ and $(D^0_\delta)^-$ coincide with D^0 and $(D^0)^-$, respectively. This further implies that the sets T_r and $\widetilde{T}_{r,\rho}$ coincide and that the notion of pseudo-granularity hence offers no advantage in this example.

Let us first study the case $r = 1/2$ illustrated on the right-hand side of Figure 5.1. Using the relation

$$\{(3,3)^\intercal\} + \{0\} \times B_\infty(0, \tfrac{1}{2}) = \{3\} \times [\tfrac{5}{2}, \tfrac{7}{2}] \subseteq G_{1/2}$$

we obtain $(3,3)^\intercal \in G^-_{1/2}$. Moreover, with $g_{1/2}(3, \frac{5}{2}) = g_{1/2}(3, \frac{7}{2}) = 0$ it is easily verified that $G^-_{1/2}$ actually coincides with the singleton $\{(3,3)^\intercal\}$. With $\|\beta\|_1 = (\frac{1}{2}, \frac{1}{2})^\intercal$ we thus

obtain the (enlarged) inner parallel set

$$\widehat{M}^-_{1/2} = \{(x,y) \in \mathbb{R}^2 |\ -\tfrac{2}{3}x + y \leqslant \tfrac{3}{2},\ \tfrac{1}{3}x + y \geqslant \tfrac{5}{2},\ x \leqslant 3\} \cap \{(3,3)^\intercal\} = \{(3,3)^\intercal\}.$$

Hence, the set $M_{1/2}$ is granular. With similar arguments it is not hard to see that M_r is granular for all $r \geqslant 1/2$, but not granular for any $0 \leqslant r < 1/2$.

Furthermore, for any $r \geqslant 0$ the function g_r possesses the Lipschitz modulus $L^g_\infty = 4$ on D^0 in the sense of Assumption 2.3.2. This results in

$$T^-_r = \{(x,y) \in \mathbb{R}^2 |\ -\tfrac{2}{3}x + y \leqslant \tfrac{3}{2},\ \tfrac{1}{3}x + y \geqslant \tfrac{5}{2},\ x \leqslant 3,\ (x-3)^2 + (y-3)^2 \leqslant r^2 - 2\}$$

for all $r \geqslant 0$ so that T is empty for any $0 \leqslant r < \sqrt{2}$.

Altogether this shows that for all $r \in [1/2, \sqrt{2})$ the set M_r is granular, while T_r is empty. As indicated earlier, none of the enlargement techniques outlined in Chapter 3 are available in our example and the fact that T^-_r is empty for any $r \in [1/2, \sqrt{2})$ hence also implies that the problem is not pseudo-granular for these choices of r.

Note that the undesirable effect in the computation of T_r stems from the fact that we use *global* information (i.e. Lipschitz constants) to *locally* approximate the enlarged inner parallel set. In contrast, the IPCP will only use local information at given points and thereby iteratively create polyhedral *outer* approximations of the enlarged inner parallel set. This, in addition to the cutting planes generated on the fly, will turn out to be the main potential of the proposed method.

5.3 The Inner Parallel Cutting Plane Method

In this section we present the inner parallel cutting plane method and outline how inner parallel cuts can be used to support OA-based methods. We show convergence of the method, demonstrate how cuts can be modified so that they are valid for $MICP$ and conclude with some remarks on the effects of a nonlinear objective function.

5.3.1 Statement of the Algorithm

The IPCP is specified in detail in Algorithm 1 and may be summarized as follows. We start by minimizing $c^\intercal x + d^\intercal y$ over $(D^0_\delta)^-$, denoting the optimization problem by LP^0. If $(D^0_\delta)^-$ is empty, then we can certify that $\widehat{M}^-_\delta$ is empty as well. Otherwise, we compute an optimal point $(\hat{x}^0, \hat{y}^0)$ of LP^0 and round it to $(\check{x}^0, \check{y}^0)$. Note that this is possible due to our boundedness assumption on D^0. In view of $(\check{x}^0, \check{y}^0) \in D^0_\delta \cap (\mathbb{R}^n \times \mathbb{Z}^m)$ and

$$M = D^0_\delta \cap G \cap (\mathbb{R}^n \times \mathbb{Z}^m),$$

we call the point $(\check{x}^0, \check{y}^0)$ ϵ-feasible with a prescribed tolerance $\epsilon \geqslant 0$ if

$$g_i(\check{x}^0, \check{y}^0) \leqslant \epsilon,\ i \in I,$$

holds. If $(\check{x}^0, \check{y}^0)$ is ϵ-feasible, then the algorithm terminates.

Otherwise, we update the set $(D^0_\delta)^-$ to $(D^1_\delta)^-$ by adding an *inner parallel cut* $(C^0)^-$. The latter corresponds to the inner parallel set of a usual Kelley cutting plane C^0, but is defined via the *rounded* point $(\check{x}^0, \check{y}^0)$ instead of the optimal point $(\hat{x}^0, \hat{y}^0)$.

Note that this is necessary, as defining a cutting plane via $(\hat{x}^0, \hat{y}^0)$ might just add a redundant constraint in cases where we have $g_i(\hat{x}^0, \hat{y}^0) \leqslant \epsilon$, $i \in I$, but $g_i(\check{x}^0, \check{y}^0) > \epsilon$

Algorithm 1: Inner parallel cutting plane method (IPCP)

Data: A problem $MICP$ with bounded set D^0, an enlargement parameter $\delta \in [0,1)$ and a feasibility tolerance $\epsilon > 0$

Result: A good ϵ-feasible point $(\breve{x}^\star, \breve{y}^\star) \in M$, or a certificate for $\widehat{M}_\delta^- = \varnothing$

1 **begin**

2 set $k \leftarrow -1$ and $(D_\delta^0)^- \leftarrow \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid Ax + By \leqslant \lfloor b \rfloor_\omega + \delta\omega - \frac{1}{2}\|\beta\|_1,\ y^\ell + (\frac{1}{2} - \delta)e \leqslant y \leqslant y^u + (\delta - \frac{1}{2})e\}$

3 **if** $(D_\delta^0)^- = \varnothing$ **then**

4 **return** $\widehat{M}_\delta^- = \varnothing$

5 **end**

6 **repeat**

7 set $k \leftarrow k + 1$

8 compute an optimal point $(\widehat{x}^k, \widehat{y}^k)$ of the problem

$$LP^k: \quad \min_{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m} c^\intercal x + d^\intercal y \quad \text{s.t.} \quad (x,y) \in (D_\delta^k)^-$$

and round it to $(\breve{x}^k, \breve{y}^k)$. Choose some $\ell_k \in I$ with

$$g_{\ell_k}(\breve{x}^k, \breve{y}^k) = \max_{i\in I} g_i(\breve{x}^k, \breve{y}^k)$$

if $g_{\ell_k}(\breve{x}^k, \breve{y}^k) > \epsilon$ **then**

9 set $(D_\delta^{k+1})^- \leftarrow (D_\delta^k)^- \cap (C^k)^-$, with

$$(C^k)^- \leftarrow \Big\{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid g_{\ell_k}(\breve{x}^k, \breve{y}^k) + \Big\langle \nabla g_{\ell_k}(\breve{x}^k, \breve{y}^k), \begin{pmatrix} x - \breve{x}^k \\ y - \breve{y}^k \end{pmatrix} \Big\rangle$$

10 $$+ \tfrac{1}{2}\left\|\nabla_y g_{\ell_k}(\breve{x}^k, \breve{y}^k)\right\|_1 \leqslant 0\Big\}$$

11 **end**

12 **until** $g_{\ell_k}(\breve{x}^k, \breve{y}^k) \leqslant \epsilon$ **or** $(D_\delta^{k+1})^- = \varnothing$;

13 **if** $(D_\delta^{k+1})^- = \varnothing$ **then**

14 **return** $\widehat{M}_\delta^- = \varnothing$

15 **else**

16 **return** $(\breve{x}^\star, \breve{y}^\star) \leftarrow (\breve{x}^k, \breve{y}^k)$

17 **end**

18 **end**

for some i. Moreover, due to the polyhedrality of a cutting plane, we have a closed-form expression (cf. line 10 of Algorithm 1) for the corresponding inner parallel set. The steps described above are then repeated, until we either find $(D_\delta^{k+1})^- = \varnothing$ and thus $\widehat{M}_\delta^- = \varnothing$, or an ϵ-feasible point $(\breve{x}^k, \breve{y}^k)$.

Let us emphasize that, by the intersection compatibility of inner parallel sets, for each k the set

$$(D_\delta^{k+1})^- = (D_\delta^0)^- \cap \bigcap_{j=0}^{k} (C^j)^- \tag{5.1}$$

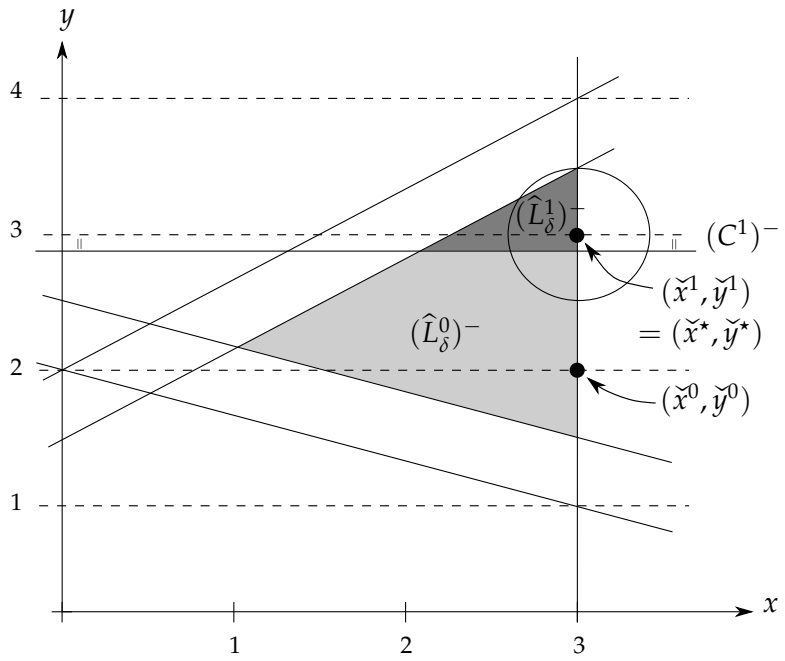


FIGURE 5.2: Idea of the IPCP for the MICP from Example 5.3.1

generated in line 10 of Algorithm 1 is the inner parallel set of

$$D_\delta^{k+1} := D_\delta^0 \cap \bigcap_{j=0}^{k} C^j, \tag{5.2}$$

with the Kelley cutting planes

$$C^j := \left\{ (x,y) \in \mathbb{R}^n \times \mathbb{R}^m \,|\, g_{\ell_j}(\breve{x}^j, \breve{y}^j) + \left\langle \nabla g_{\ell_j}(\breve{x}^j, \breve{y}^j), \begin{pmatrix} x - \breve{x}^j \\ y - \breve{y}^j \end{pmatrix} \right\rangle \leqslant 0 \right\} \tag{5.3}$$

to G, which can be used for OA-based methods.

Example 5.3.1. *The IPCP is visualized in Figure 5.2 for an MICP with the feasible set $M_{1/2}$ from Example 5.2.2 and the objective function $-x + y$. In the first iteration, the algorithm computes the infeasible point $(\breve{x}^0, \breve{y}^0) = (3, 2)$ and adds the inner parallel cut*

$$(C^1)^- = \{(x,y) \in \mathbb{R}^2 \,|\, y \geqslant 2.875\}.$$

It then terminates in the second iteration, returning the feasible point $(\breve{x}^\star, \breve{y}^\star) = (3,3)^\intercal$.

While the analysis of Section 5.3.2 will expose that granularity ($\widehat{M}_\delta^- \neq \varnothing$) is a sufficient condition for the IPCP to compute an ϵ-feasible point, Example 5.3.1 already shows that granularity is not necessary. In fact, while none of the sets M_r with $0 \leqslant r < 1/2$ from Example 5.2.2 is granular, the IPCP still computes the feasible point $(\breve{x}^\star, \breve{y}^\star) = (3,3)^\intercal$ after two iterations.

Note that for $I = \varnothing$ the MICP is an MILP, and the IPCP then corresponds to the FRA-SOR from Chapter 4. In particular, this entails that it terminates after the first iteration. Moreover, for $m = 0$, the IPCP collapses to Kelley's cutting plane method.

Remark 5.3.2. *We already mentioned in Remark 4.1.1 that when continuous variables are present we might be able to improve the objective value when applying a postprocessing step.*

In fact, fixing the integer variables to $\breve{y}^\star$, we may update $\breve{x}^\star$ by computing an optimal point of the continuous convex optimization problem

$$CP(\breve{y}^\star): \quad \min_{x\in\mathbb{R}^n} c^\intercal x \quad s.t. \quad (x,\breve{y}^\star)\in\widehat{M}.$$

Indeed, in our computational study we observed that this postprocessing step was useful for many applications.

5.3.2 Convergence of the Algorithm

Before we proceed by studying the convergence of the algorithm, let us initially establish the validity of the two termination criteria of the IPCP for the claimed outputs (either finding an ϵ-feasible point or certifying $\widehat{M}_\delta^- = \varnothing$).

Firstly, if $g_{\ell_k}(\breve{x}^k,\breve{y}^k)\leqslant\epsilon$ holds in some iteration k, then due to $(\breve{x}^k,\breve{y}^k)\in D^0$ and $\breve{y}^k\in\mathbb{Z}^m$ the output from line 16 of Algorithm 1 is indeed an ϵ-feasible point.

Secondly, the subsequent Lemma 5.3.3 will show that for any k the set $(D_\delta^k)^-$ provides an outer approximation of the enlarged inner parallel set $\widehat{M}_\delta^-$ so that also the statement of lines 13 and 14 in Algorithm 1 is correct.

Lemma 5.3.3. *$\widehat{M}_\delta^- \subseteq (D_\delta^k)^-$ holds for all $k\in\mathbb{N}_0$.*

Proof. Due to the convexity of the functions g_i, $i\in I$, and the resulting outer approximation property of the sets C^j, $0\leqslant j\leqslant k$, for G we have

$$\widehat{M}_\delta = D_\delta^0\cap G\subseteq D_\delta^0\cap\bigcap_{j=0}^{k-1}C^j = D_\delta^k \tag{5.4}$$

and, by the monotonicity of inner parallel sets, $\widehat{M}_\delta^-\subseteq(D_\delta^k)^-$. □

Lemma 5.3.3 substantiates the already mentioned advantage of the IPCP: While determining the set T_δ^- required the computation of *global* Lipschitz constants which lead to an *inner* approximation, inner parallel sets of Kelley cutting planes are explicitly computable from *local* information and thus create *outer* approximations of the enlarged inner parallel set.

Let us next investigate the convergence properties of the IPCP. We will initially establish that the inner parallel set of a cutting plane not only excludes the point $(\breve{x}^k,\breve{y}^k)$, but also all other points that are potentially rounded to $(\breve{x}^k,\breve{y}^k)$.

Lemma 5.3.4. *For any iterate $(\breve{x}^k,\breve{y}^k)$ and any rounding $(\breve{x},\breve{y})$ of any point $(\widehat{x},\widehat{y})\in(C^k)^-$ we have $(\breve{x},\breve{y})\neq(\breve{x}^k,\breve{y}^k)$.*

Proof. Firstly, $(\widehat{x},\widehat{y})\in(C^k)^-$ and rounding properties imply

$$(\breve{x},\breve{y})\in(\widehat{x},\widehat{y})+K\subseteq C^k.$$

Moreover, because C^k corresponds to a cutting plane that is generated via $(\breve{x}^k,\breve{y}^k)$ (cf. (5.3)), clearly we also have $(\breve{x}^k,\breve{y}^k)\notin C^k$ so that the assertion is shown. □

Note that the proof of Lemma 5.3.4 does not use specific information about the cutting planes, but only requires that the point $(\breve{x}^k,\breve{y}^k)$ is excluded from subsequent iterations. Hence, this result is actually equally valid for methods that generate a sequence of inner parallel sets of other (not necessarily gradient based) cutting planes.

Proposition 5.3.5. *The sequence* $\big((\breve{x}^k, \breve{y}^k)\big)$ *contains pairwise different points.*

Proof. For any $(\breve{x}^{k_1}, \breve{y}^{k_1})$, $(\breve{x}^{k_2}, \breve{y}^{k_2}) \in \big((\breve{x}^k, \breve{y}^k)\big)$ with $k_2 > k_1$ we have $(\hat{x}^{k_2}, \hat{y}^{k_2}) \in (C^{k_1})^-$ which, together with Lemma 5.3.4, shows the assertion. □

We can now prove our first main result for purely integer convex optimization problems ($n = 0$).

Theorem 5.3.6. *For any purely integer convex optimization problem, any* $\epsilon \geqslant 0$*, and any* $\delta \in [0, 1)$ *the IPCP terminates in finitely many steps with either an* ϵ*-feasible point or a certificate for* $\widehat{M}_\delta^- = \varnothing$*.*

Proof. If Algorithm 1 terminates after finitely many steps due to $(D_\delta^{k+1})^- = \varnothing$, this is a certificate for $\widehat{M}_\delta^- = \varnothing$. Otherwise, assume infinitely many iterations and let $(D_\delta^{k+1})^- \neq \varnothing$ hold in all iterations. Then Proposition 5.3.5 and the fact that the sequence of iterates $(\breve{y}^k)$ lies in the bounded set D lead to a contradiction. Consequently, the IPCP terminates after finitely many steps. □

Hence, for granular integer convex optimization problems, we may even set ϵ to zero and find a feasible point after finitely many iterations. Let us next extend the finding from Theorem 5.3.6 to the general mixed-integer case where, like in the purely continuous case, the presence of continuous variables leads to the necessity to choose a positive termination tolerance ϵ.

Theorem 5.3.7. *For any mixed-integer convex optimization problem, any* $\epsilon > 0$*, and any* $\delta \in [0, 1)$ *the IPCP terminates in finitely many steps with either an* ϵ*-feasible point or a certificate for* $\widehat{M}_\delta^- = \varnothing$*.*

Proof. As in the proof of Theorem 5.3.6 we may assume $(D_\delta^{k+1})^- \neq \varnothing$ in all iterations.

Then, as D^0 is bounded, there exists a subsequence $\big((\breve{x}^k, \breve{y}^k)\big)$ with $\lim_k (\breve{x}^k, \breve{y}^k) = (\widetilde{x}, \widetilde{y}) \in D^0 \cap (\mathbb{R}^n \times \mathbb{Z}^m)$ and $g_{\ell_k}(\breve{x}^k, \breve{y}^k) > \epsilon$ for all $k \in \mathbb{N}_0$. Moreover, since the index set I is finite, we may choose this subsequence so that $\ell_k = \widetilde{\ell} \in I$ holds for all $k \in \mathbb{N}_0$. Taking the limit then results in $g_{\widetilde{\ell}}(\widetilde{x}, \widetilde{y}) \geqslant \epsilon$.

On the other hand, for any $k \in \mathbb{N}_0$ the point $(\breve{x}^{k+1}, \breve{y}^{k+1})$ lies in C^k, as it is a rounding of the optimal point $(\hat{x}^{k+1}, \hat{y}^{k+1}) \in (D_\delta^{k+1})^- \subseteq (C^k)^-$. Hence it satisfies the inequality

$$0 \geqslant g_{\widetilde{\ell}}(\breve{x}^k, \breve{y}^k) + \left\langle \nabla g_{\widetilde{\ell}}(\breve{x}^k, \breve{y}^k), \begin{pmatrix} \breve{x}^{k+1} - \breve{x}^k \\ \breve{y}^{k+1} - \breve{y}^k \end{pmatrix} \right\rangle$$

which, in the limit, yields the contradiction $0 \geqslant g_{\widetilde{\ell}}(\widetilde{x}, \widetilde{y})$. □

We shall demonstrate next that we indeed need $\epsilon > 0$ to guarantee convergence of the inner parallel cutting plane method in the general mixed integer case.

Example 5.3.8. *Consider the two dimensional mixed-integer problem*

$$MICP: \quad \min_{x,y} -x \quad s.t. \quad g(x,y) \leqslant 0, \;\; 0 \leqslant x \leqslant 1, \;\; 0 \leqslant y \leqslant 1, \;\; y \in \mathbb{Z}$$

with $g(x,y) := x^2 + (y - \frac{1}{2})^2 - \frac{1}{4}$*. Note that the feasible set of MICP consists merely of the two points* $\{(0,0), (0,1)\}$*, and that the enlarged inner parallel set* $\widehat{M}_\delta^-$ *contains only the point* $(\bar{x}, \bar{y}) = (0, \frac{1}{2})$ *for any* $\delta \in [0, 1)$*. By Theorem 5.3.7, this implies that for any* $\epsilon > 0$ *the IPCP finds an* ϵ*-feasible point in finitely many iterations.*

Assume that the algorithm even terminates for $\epsilon = 0$ in any iteration $k \geqslant 0$. This can only happen if $(\hat{x}^k, \hat{y}^k) \in \{0\} \times [\frac{1}{2} - \delta, \frac{1}{2} + \delta]$ holds. Then the point $(\bar{x}, \bar{y}) \in \widehat{M}_\delta^- \subseteq (D_\delta^k)^-$ (Lemma 5.3.3) possesses the same objective function value as the optimal point $(\hat{x}^k, \hat{y}^k)$ of LP^k and is, thus, also optimal for LP^k. We will show, however, that $(\bar{x}, \bar{y})$ cannot be optimal for any problem LP^k with $k \geqslant 0$, leading to a contradiction.

In fact, independently of the choice of δ, $(\bar{x}, \bar{y})$ is clearly not optimal for the problem LP^0 with the feasible set

$$(D_\delta^0)^- = \{(x, y) \in \mathbb{R}^2 |\ 0 \leqslant x \leqslant 1,\ -\delta + \tfrac{1}{2} \leqslant y \leqslant \delta + \tfrac{1}{2}\}.$$

Therefore, for any $k \geqslant 1$ the optimality of $(\bar{x}, \bar{y})$ for LP^k requires the activity of at least one of the cutting planes $(C^j)^-$, $j \in \{0, \ldots, k-1\}$, constructed so far. For the corresponding index j this means

$$(\breve{x}^j)^2 + (\breve{y}^j - \tfrac{1}{2})^2 - \tfrac{1}{4} + \left\langle \begin{pmatrix} 2\breve{x}^j \\ 2\breve{y}^j - 1 \end{pmatrix}, \begin{pmatrix} 0 - \breve{x}^j \\ \frac{1}{2} - \breve{y}^j \end{pmatrix} \right\rangle + \tfrac{1}{2}|2\breve{y}^j - 1| = 0.$$

Due to $\breve{y}^j \in \{0, 1\}$ this condition implies $\breve{x}^j = 0$, so that $(\breve{x}^j, \breve{y}^j)$ lies in M and the algorithm must already have terminated in iteration j.

Hence, for purely integer convex optimization problems with an nonempty enlarged inner parallel set, the IPCP terminates with a feasible point, whereas in the mixed-integer case, it may produce only an outer approximation of a feasible point.

Remark 5.3.9. *Central to the convergence results of the IPCP is the assumption that the points $(\hat{x}^k, \hat{y}^k)$ are feasible for LP^k, which ensures that the sequence of iterates $((\breve{x}^k, \breve{y}^k))$ contains pairwise different points (cf. Lemma 5.3.4). Because LP-solvers work with feasibility tolerances, this assumption may sometimes be violated in practice. This will indeed become relevant in our numerical study in Section 5.5. The next arguments illustrate that convergence of the method can still be guaranteed, as long as the feasibility tolerance ϵ_{LP} of the LP-solver is smaller than the feasibility tolerance ϵ of IPCP.*

More specifically, for iterations $k_1, k_2 \in \mathbb{N}$ with $k_1 < k_2$ let $(\hat{x}^{k_2}, \hat{y}^{k_2})$ be ϵ_{LP}-feasible for $(C^{k_1})^-$, that is,

$$g_{\ell_{k_1}}(\breve{x}^{k_1}, \breve{y}^{k_1}) + \left\langle \nabla g_{\ell_{k_1}}(\breve{x}^{k_1}, \breve{y}^{k_1}), \begin{pmatrix} \hat{x}^{k_2} - \breve{x}^{k_1} \\ \hat{y}^{k_2} - \breve{y}^{k_1} \end{pmatrix} \right\rangle + \tfrac{1}{2} \left\| \nabla_y g_{\ell_k}(\breve{x}^{k_1}, \breve{y}^{k_1}) \right\|_1 \leqslant \epsilon_{LP} \quad (5.5)$$

holds and let $\epsilon_{LP} < \epsilon$ hold.

Now assume that two iterates generated by the IPCP are identical, that is $(\breve{x}^{k_1}, \breve{y}^{k_1}) = (\breve{x}^{k_2}, \breve{y}^{k_2})$. This yields $\hat{x}^{k_2} = \breve{x}^{k_1}$ and $|\hat{y}^{k_2} - \breve{y}^{k_1}| (= |\hat{y}^{k_2} - \breve{y}^{k_2}|) \leqslant \frac{1}{2}$ and thus by Hölder's inequality

$$\left\langle \nabla g_{\ell_{k_1}}(\breve{x}^{k_1}, \breve{y}^{k_1}), \begin{pmatrix} \hat{x}^{k_2} - \breve{x}^{k_1} \\ \hat{y}^{k_2} - \breve{y}^{k_1} \end{pmatrix} \right\rangle + \tfrac{1}{2} \left\| \nabla_y g_{\ell_k}(\breve{x}^{k_1}, \breve{y}^{k_1}) \right\|_1 \geqslant 0. \quad (5.6)$$

Inequalities (5.5) and (5.6) imply $g_{\ell_{k_1}}(\breve{x}^{k_1}, \breve{y}^{k_1}) \leqslant \epsilon_{LP}$, which contradicts $g_{\ell_{k_1}}(\breve{x}^{k_1}, \breve{y}^{k_1}) > \epsilon$, the condition for generating the cutting plane $(C^{k_1})^-$.

This recovers Proposition 5.3.5. Hence, we may simply reuse all arguments presented in Theorem 5.3.6 to show convergence in the purely integer case for $0 < \epsilon_{LP} < \epsilon$. The proof of Theorem 5.3.7 can also be extended to such cases with small modifications. Indeed, perturbing the right-hand sides of the constraints of $(D_\delta^j)^-$, $j = 1, \ldots, k+1$, by ϵ_{LP} allows us to reuse all presented arguments.

5.3.3 Using Inner Parallel Cuts in Outer Approximation Based Methods

There are different possibilities to integrate the IPCP into OA-based methods. In particular, it seems very promising to initially run the IPCP with the post processing step from Remark 4.1.1 and then to initialize the auxiliary $MILP$ with a number of reversed inner parallel cutting planes C^k. To be more specific, let

$$(C^k)^- = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m \mid \alpha_k^\intercal x + \beta_k^\intercal y \leqslant b_k\}$$

be some cutting plane provided by the IPCP, then by (5.4) the reversed inner parallel cut (RIPC)

$$\alpha_k^\intercal x + \beta_k^\intercal y \leqslant b_k + \tfrac{1}{2} \|\beta_k\|_1$$

is valid for $MICP$ and may help to better approximate the latter. In addition, if the method successfully returned a feasible point $(\breve{x}^\star, \breve{y}^\star)$, we can incorporate into the auxiliary $MILP$ further potentially useful information, like the objective bound

$$c^\intercal x + d^\intercal y \leqslant c^\intercal \breve{x}^\star + d^\intercal \breve{y}^\star$$

and the linearizations

$$g_\ell(\breve{x}^\star, \breve{y}^\star) + \left\langle \nabla g_\ell(\breve{x}^\star, \breve{y}^\star), \begin{pmatrix} x - \breve{x}^\star \\ \breve{y} - y^\star \end{pmatrix} \right\rangle \leqslant 0$$

of some (almost) active constraints g_ℓ.

We further remark that the IPCP can be started in any iteration of OA-based methods, where D^0 then corresponds to the relaxed feasible set of the auxiliary $MILP$ of the current iteration. Moreover, instead of waiting for the method to converge, one could also run it for a certain number of iterations until some criterion concerning the quality of enhancement of the outer approximation is satisfied.

5.3.4 The Effect of a Nonlinear Objective Function

Within the scope of solving practical problems, one often has to deal with a nonlinear convex objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$. If the latter can be, from a practical perspective, efficiently minimized over a polyhedron, we may just apply the IPCP as formulated in Algorithm 1, replacing $c^\intercal x + d^\intercal y$ with f. Often, however, the polyhedrality of the sub-problems LP^k is beneficial. Then we may apply the IPCP to the equivalent epigraph reformulation of $MICP$, where we minimize an additional continuous variable z under the constraints $(x, y, z) \in M \times \mathbb{R}$ and $f(x, y) \leqslant z$. The following proposition shows that this reformulation preserves granularity.

Proposition 5.3.10. *The mixed-integer convex optimization problem*

$$MICP: \quad \min_{x,y} f(x, y) \quad s.t. \quad (x, y) \in M$$

is granular if and only if its epigraph reformulation

$$MICP_{\text{epi}}: \quad \min_{x,y,z} z \quad s.t. \quad (x, y) \in M,\ f(x, y) \leqslant z$$

is granular.

Proof. First, the granularity of $MICP_{\text{epi}}$ implies the existence of some $\delta \in [0, 1)$ and $(\bar{x}, \bar{y}) \in \widehat{M}_\delta^-$ which immediately entails the granularity of $MICP$. It remains to show that the granularity of $MICP$ implies the granularity of $MICP_{\text{epi}}$.

In fact, if $MICP$ is granular, then there exist some $\delta \in [0,1)$ and $(\bar{x},\bar{y}) \in \widehat{M}_{\delta}^{-}$. Hence, it suffices to show the existence of some $\bar{z} \in \mathbb{R}$ which satisfies $f(\bar{x},\bar{y}+\eta) \leqslant \bar{z}$ for all $\eta \in B_{\infty}(0,\frac{1}{2})$. Since, being convex on $\mathbb{R}^n \times \mathbb{R}^m$, the function f is continuous, and since the set $B_{\infty}(0,\frac{1}{2})$ is nonempty and compact, the assertion follows from the Weierstrass theorem. □

We remark that the convexity of $MICP$ does not play a role for the validity of Proposition 5.3.10 and that the epigraph reformulation hence generally preserves granularity for continuous objective functions.

To have the epigraph reformulation $MICP_{\text{epi}}$ meet the input requirements of the IPCP, in addition to its assumptions on $MICP$, lower and upper bounds for the auxiliary variable z must be available which do not interfere with the values of f on M. A lower bound for z, which is necessary for the boundedness of the problem LP^0, can be obtained by computing the minimal value of f over the polyhedral set $D^0 \supseteq M$. For obtaining an upper bound, we may, e.g., employ techniques from interval arithmetic (cf., e.g., [35, 57]). From a practical perspective, however, the explicit knowledge of this upper bound is not necessary because z is minimized in each iteration of the IPCP.

Let us briefly discuss the case when the objective function f is the only source of nonlinearity in the problem $MICP$, that is, its feasible set M is polyhedral. Then, although the feasible set of $MICP_{\text{epi}}$ is nonpolyhedral, all iterates of the IPCP applied to $MICP_{\text{epi}}$ are feasible for $MICP$. Yet, the IPCP iteratively incorporates more information from the objective function by adding inner parallel objective cuts

$$f(\check{x}^k,\check{y}^k) + \left\langle \nabla f(\check{x}^k,\check{y}^k), \begin{pmatrix} x-\check{x}^k \\ y-\check{y}^k \end{pmatrix} \right\rangle + \tfrac{1}{2}\left\|\nabla_y f(\check{x}^k,\check{y}^k)\right\|_1 \leqslant z$$

until the termination criterion $f(\check{x}^k,\check{y}^k) \leqslant \check{z}^k + \epsilon$ is met. Any premature termination of the IPCP hence at least yields a feasible point for $MICP$, but with an idle potential to improve its objective value.

We shall further discuss these effects in our computational study in Section 5.5 for several practical applications with a nonlinear objective function.

5.4 Bounds on the Objective Value

In this section we give a theoretical analysis for the objective values of the iterates generated by the IPCP. Note that, if successful, the IPCP does not heuristically generate some arbitrary ϵ-feasible point of the problem $MICP$, but takes into account its objective function, as well as a set that is closely linked to its feasible set M. Clearly, due to the necessary modifications of the feasible set on the transition from $MICP$ to LP^k, as well as due to rounding effects, we cannot expect the IPCP to compute an optimal point of $MICP$. We can, however, derive bounds on the objective value of the iterates that merely depend on the problem data. To evaluate the quality of some IPCP iterate $(\check{x}^k,\check{y}^k)$, we will compare its objective value $\check{v}^k := c^\intercal \check{x}^k + d^\intercal \check{y}^k$ with the optimal value v of $MICP$.

5.4.1 A-Posteriori Bounds

As the value v is unknown, a computable upper bound of $\check{v}^k - v$ relies on a computable lower bound for v. Let us mention three possibilities to establish such

bounds which are a-posteriori in the sense that, upon termination of the IPCP, additional auxiliary optimization problems have to be solved.

Firstly, we may compare $\breve{v}^k$ to the optimal value v_{CP} of the continuous relaxation

$$CP: \quad \min_{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m} c^\intercal x + d^\intercal y \quad \text{s.t.} \quad (x,y) \in D^0,\ g_i(x,y) \leqslant 0,\ i \in I,$$

of $MICP$ and obtain the upper bound

$$\breve{v}^k - v \leqslant \breve{v}^k - v_{CP}. \tag{5.7}$$

Observe that for the construction of the relaxed problem CP it would not make sense to replace D^0 by an enlargement D^0_δ with $\delta > 0$ since this would lead to a worse lower bound $v_{CP,\delta} \leqslant v_{CP}$. On the other hand, in the term $\breve{v}^k - v_{CP}$ the value $\breve{v}^k$ does depend on δ as it is computed via an enlarged inner parallel set in line 8 of Algorithm 1.

In this approach it would neither make sense to add the reversed inner parallel cuts (RIPCs) available in iteration k to the feasible set $\widehat{M}$ of CP, that is, to replace the set D^0 by $D^k := D^0 \cap \bigcap_{j=1}^{k-1} C^j$. In fact, by (5.4) the set D^k forms an outer approximation for $\widehat{M}$ and the RIPCs are, thus, redundant in this construction.

A second possibility to lower bound v is the computation of the optimal value of the MILP relaxation of $MICP$. While employing a positive enlargement parameter δ still would not make sense in this approach, including the available RIPCs can be expected to improve the bound since in the MILP relaxation the original nonlinear constraints g_i, $i \in I$, are dropped. By the relaxation property of RIPCs from (5.4) the optimal value v^k_{MILP} of the problem

$$\overline{MILP}^k: \quad \min_{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m} c^\intercal x + d^\intercal y \quad \text{s.t.} \quad (x,y) \in D^k \cap (\mathbb{R}^n \times \mathbb{Z}^m)$$

is a valid lower bound for v, leading to the estimate

$$\breve{v}^k - v \leqslant \breve{v}^k - v^k_{MILP}. \tag{5.8}$$

Note that the gap $v - v^k_{MILP}$ decreases with increasing values of k when useful RIPCs are added to D^0. A tight bound $\breve{v}^k - v^k_{MILP}$ thus not only demonstrates that the generated feasible point is of good quality, but also points to the fact that the generated cutting planes are probably helpful. Moreover, $\breve{v}^k - v^k_{MILP}$ can be compared to $\breve{v}^k - v^0_{MILP}$ to determine more specifically if the RIPCs are useful. We shall come back to this comparison in our computational study in Section 5.5.

Thirdly, dropping the integrality constraints of $\overline{MILP}^k$ yields the continuous linear problem

$$\overline{LP}^k: \quad \min_{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m} c^\intercal x + d^\intercal y \quad \text{s.t.} \quad (x,y) \in D^k \tag{5.9}$$

with minimum value v^k_{LP}. This value satisfies $v^k_{LP} \leqslant v^k_{MILP}$ and, thus,

$$\breve{v}^k - v \ \leqslant\ \breve{v}^k - v^k_{LP} \tag{5.10}$$

is another option for a computable upper bound. Whereas the explicit computations of the upper bounds (5.7) and (5.8) need the solution of a CP and an MILP, respectively, for the potentially larger bound from (5.10) we only need to solve an LP.

5.4.2 A-Priori Bounds

In contrast to a-posteriori bounds, a-priori bounds do not rely on the solution of additional optimization problems, but they merely depend on the problem data. Such bounds facilitate the sensitivity analysis of the algorithmic output with respect to changes in the input data. Since they usually involve constants which are hard to compute explicitly (see below for details), a-priori bounds often are rather of theoretical than of computational interest.

Recall that in the feasible rounding approach FRA-SOR for general MINLPs from Chapter 4, the computed optimal point (x^{ob}, y^{ob}) of f over $\widetilde{T}_\rho^-$ is rounded to $(\breve{x}^{ob}, \breve{y}^{ob})$ with objective value $\breve{v}^{ob} := f(\breve{x}^{ob}, \breve{y}^{ob})$. In [62] an a-priori bound for the deviation $\breve{v}^{ob} - v$ is given. We will state this bound for the setting of the present MICP and show that an analogous a-priori bound also holds for $\breve{v}^k - v$.

Earlier, we emphasized that a crucial advantage of the IPCP is that it does not necessarily need granularity for the computation of a feasible point (cf., e.g., Example 5.3.1). Yet, the bound from [62] is stated under more restrictive assumptions, including that of a nonempty inner approximation of the enlarged inner parallel set. To adapt this bound to the present setting and to show its validity for the IPCP, we will initially also rely on these assumptions. Subsequently we shall demonstrate that it is possible to state a similar, but slightly different bound for $\breve{v}^k - v$ under significantly milder assumptions.

For the statement of the bound from [62] let $\delta = 0$ and let L_∞ denote the vector of Lipschitz constants from Assumption 2.3.2. We may thus write

$$T^- := T_0^- = \{(x, y) \in (D^0)^- \mid g(x, y) + \tfrac{1}{2} L_\infty \leqslant 0\}$$

and assume $T^- \neq \varnothing$.

For the objective function $f(x, y) = c^\intercal x + d^\intercal y$ and any optimal point $(\hat{x}, \hat{y})$ of CP [62, Lemma 1] yields the estimate

$$\breve{v}^{ob} - v \leqslant \tfrac{1}{2} \|d\|_1 + \|(c, d)\|_2 \operatorname{dist}(\,(\hat{x}, \hat{y}), T^-), \tag{5.11}$$

where $\operatorname{dist}(\,(\hat{x}, \hat{y}), T^-) := \inf_{(x,y)\in T^-} \|(x, y) - (\hat{x}, \hat{y})\|_2$ denotes the Euclidean distance of $(\hat{x}, \hat{y})$ from T^-.

A further estimate of $\operatorname{dist}(\,(\hat{x}, \hat{y}), T^-)$ in terms of the problem data is possible by a global error bound

$$\begin{aligned} &\operatorname{dist}(\,(x, y), T^-) \\ &\leqslant \gamma \left\| \left(Ax + By - b + \tfrac{\|\beta\|_1}{2} \right)_+, \left(y^\ell + \tfrac{e}{2} - y \right)_+, \left(y - y^u + \tfrac{e}{2} \right)_+, \left(g(x, y) + \tfrac{L_\infty}{2} \right)_+ \right\|_\infty \end{aligned} \tag{5.12}$$

for any $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$, where $\gamma > 0$ is a Hoffman constant and, with the componentwise positive-part operator $a_+ := (\max\{0, a_1\}, \ldots, \max\{0, a_q\})^\intercal$ for vectors $a \in \mathbb{R}^q$, the second factor on the right-hand side of (5.12) is a penalty function for the set T^-.

Due to the seminal work [40] of Hoffman, in the polyhedral setting a constant γ with (5.12) exists without any further assumptions. Its computation, on the other hand, is often intricate even in the polyhedral case. Several suggestions are given [45, 49, 50, 53, 75], often based on the global solution of some nonconvex optimization problem. For global error bounds of broader problem classes see, for example, the surveys [5, 64]. To cite an early result for the nonpolyhedral case from [66], if for convex functions g_i, $i \in I$, the set T^- is bounded and satisfies Slater's condition,

then the global error bound (5.12) holds with some $\gamma > 0$. As in our application the boundedness of T^- follows from the boundedness of D^0, assuming Slater's condition in T^- guarantees (5.12).

Since in the present approach the considered optimal point $(\hat{x}, \hat{y})$ of CP lies in $\widehat{M}$, the estimate (5.12) can be shown [62, 72] to reduce to

$$\operatorname{dist}(\,(\hat{x},\hat{y}), T^-) \leqslant \tfrac{1}{2}\gamma \left\|(\|\beta\|_1, e, e, L_\infty)\right\|_\infty = \tfrac{1}{2}\gamma \max\{\|(\|\beta\|_1)\|_\infty, 1, \|L_\infty\|_\infty\}. \quad (5.13)$$

The combination of (5.11) and (5.13) yields the a-priori bound

$$\breve{v}^{ob} - v \leqslant \tfrac{1}{2}\left(\|d\|_1 + \|(c,d)\|_2\, \gamma \max\{\|(\|\beta\|_1)\|_\infty, 1, \|L_\infty\|_\infty\}\right) \quad (5.14)$$

as an adaption of [62, Theorem 3] to the present setting, whenever T^- satisfies Slater's condition.

As announced, next we shall show that in (5.14) the value $\breve{v}^{ob}$ may be replaced by $\breve{v}^k$. It suffices to show that this replacement is possible in (5.11). To this end, recall that $\breve{v}^k$ depends on the enlargement parameter δ. To ensure $\widehat{M} \subseteq \widehat{M}_\delta$ in the following let $\delta \in [\delta_e, 1)$ with δ_e from Equation (3.6). The main consequence of this choice of δ is that it guarantees the chain of inclusions

$$T^- \subseteq \widehat{M}^- \subseteq \widehat{M}_\delta^- \subseteq (D_\delta^k)^-, \quad (5.15)$$

where the last inclusion holds by Lemma 5.3.3.

The proof of the next result follows the lines of the proof for [62, Lemma 1], but it takes account of the difference in the computation of $\breve{v}^k$ compared to $\breve{v}^{ob}$.

Lemma 5.4.1. *Let $T^- \neq \varnothing$, let $(\hat{x}, \hat{y})$ denote any optimal point of CP, let $\delta \in [\delta_e, 1)$, and let $(\breve{x}^k, \breve{y}^k)$ be the current iterate of IPCP in some iteration $k \in \mathbb{N}_0$. Then its objective value $\breve{v}^k = c^\intercal \breve{x}^k + d^\intercal \breve{y}^k$ satisfies*

$$\breve{v}^k - v \leqslant \tfrac{1}{2}\|d\|_1 + \|(c,d)\|_2 \operatorname{dist}\left((\hat{x},\hat{y}), T^-\right).$$

Proof. Since T^- is a nonempty subset of the bounded set D^0, the Weierstrass theorem guarantees the existence of an optimal point of the orthogonal projection problem

$$Pr\left((\hat{x},\hat{y}), T^-\right): \quad \min_{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m} \left\| \begin{pmatrix} x-\hat{x} \\ y-\hat{y} \end{pmatrix} \right\|_2 \quad \text{s.t.} \quad (x,y) \in T^-$$

which we denote by (x^π, y^π). Furthermore, let $(\hat{x}^k, \hat{y}^k)$ be the minimal point of LP^k which is rounded to $(\breve{x}^k, \breve{y}^k)$ by the IPCP. Then, using (5.7) we may bound $\breve{v}^k - v$ above by

$$\begin{aligned}\breve{v}^k - v \leqslant \breve{v}^k - v_{CP} &= (c^\intercal \breve{x}^k + d^\intercal \breve{y}^k) - (c^\intercal \hat{x} + d^\intercal \hat{y}) \\ &= \Big((c^\intercal \breve{x}^k + d^\intercal \breve{y}^k) - (c^\intercal \hat{x}^k + d^\intercal \hat{y}^k)\Big) + \Big((c^\intercal \hat{x}^k + d^\intercal \hat{y}^k) - (c^\intercal x^\pi + d^\intercal y^\pi)\Big) \\ &\quad + \Big((c^\intercal x^\pi + d^\intercal y^\pi) - (c^\intercal \hat{x} + d^\intercal \hat{y})\Big). \end{aligned} \quad (5.16)$$

Due to $\breve{x}^k = \hat{x}^k$, Hölder's inequality and the definition of a rounding, for the first term in (5.16) we obtain

$$(c^\intercal \breve{x}^k + d^\intercal \breve{y}^k) - (c^\intercal \hat{x}^k + d^\intercal \hat{y}^k) \leqslant \|d\|_1 \, \|\breve{y}^k - \hat{y}^k\|_\infty \leqslant \tfrac{1}{2}\|d\|_1.$$

Regarding the second term in (5.16), by (5.15) the point $(x^\pi, y^\pi) \in T^-$ is feasible for LP^k so that the optimality of $(\hat{x}^k, \hat{y}^k)$ for LP^k yields

$$(c^\intercal \hat{x}^k + d^\intercal \hat{y}^k) - (c^\intercal x^\pi + d^\intercal y^\pi) \leqslant 0.$$

Finally, for the third term in (5.16) the Cauchy-Schwarz inequality and the fact that (x^π, y^π) is the orthogonal projection of $(\hat{x}, \hat{y})$ to T^- imply

$$(c^\intercal x^\pi + d^\intercal y^\pi) - (c^\intercal \hat{x} + d^\intercal \hat{y}) \leqslant \|(c,d)\|_2 \operatorname{dist}\left((\hat{x}, \hat{y}), T^-\right).$$

This shows the assertion. □

The combination of Lemma 5.4.1 with (5.13) yields the following result which verifies that the a-priori bound for $\breve{v}^{ob}$ from the setting of [62] also holds for $\breve{v}^k$.

Theorem 5.4.2. *Let T^- satisfy Slater's condition, let γ satisfy* (5.12)*, let $\delta \in [\delta_e, 1)$, and let $(\breve{x}^k, \breve{y}^k)$ be the current iterate of IPCP in some iteration $k \in \mathbb{N}_0$. Then its objective value $\breve{v}^k = c^\intercal \breve{x}^k + d^\intercal \breve{y}^k$ satisfies*

$$\breve{v}^k - v \leqslant \tfrac{1}{2}\left(\|d\|_1 + \|(c,d)\|_2 \, \gamma \max\{\|(\|\beta\|_1)\|_\infty, 1, \|L_\infty\|_\infty\}\right).$$

A similar analysis is possible based on the continuous linear problem $\overline{LP}^k$ from (5.9) in place of the continuous convex problem CP. This allows us to drop the restrictive assumption $T^- \neq \varnothing$ and offers some further insights into how inner parallel cuts affect the deviation $\breve{v}^k - v$.

In this setting, Lemma 5.4.1 can be shown with T^- replaced by $(D^k)^-$ and with $(\hat{x}, \hat{y})$ replaced by any optimal point $(\bar{x}^k, \bar{y}^k)$ of $\overline{LP}^k$. The corresponding global error bound then has to be formulated for the constraints describing $(D^k)^-$. It is of the form

$$\begin{aligned} &\operatorname{dist}(\,(x,y), (D^k)^-) \qquad (5.17)\\ &\leqslant \gamma^k \left\| \left(Ax + By - b + \tfrac{\|\beta\|_1}{2}\right)_+, \left(y^\ell + \tfrac{e}{2} - y\right)_+, \left(y - y^u + \tfrac{e}{2}\right)_+, (c^j(x,y))_+, j = 1, \ldots, k \right\|_\infty \end{aligned}$$

for all $(x,y) \in \mathbb{R}^n \times \mathbb{R}^m$, with the functions

$$c^j(x,y) := g_{\ell_j}(\breve{x}^j, \breve{y}^j) + \left\langle \nabla g_{\ell_j}(\breve{x}^j, \breve{y}^j), \begin{pmatrix} x - \breve{x}^j \\ y - \breve{y}^j \end{pmatrix} \right\rangle + \tfrac{1}{2}\|\nabla_y g_{\ell_j}(\breve{x}^j, \breve{y}^j)\|_1.$$

For any $(\bar{x}, \bar{y}) \in D^k$ it reduces to

$$\operatorname{dist}\left((\bar{x}, \bar{y}), (D^k)^-\right) \leqslant \tfrac{1}{2}\gamma^k \left\| \left(\|\beta\|_1, 1, \|\nabla_y g_{\ell_j}(\breve{x}^j, \breve{y}^j)\|_1, \; j = 1, \ldots, k\right) \right\|_\infty,$$

so that the following variation of Theorem 5.4.2 can be shown. Recall that in this polyhedral setting a Hoffman constant γ^k with (5.17) exists without further assumptions.

Theorem 5.4.3. *Let $\delta \in [\delta_e, 1)$, for some iteration $k \in \mathbb{N}_0$ let γ^k satisfy* (5.17)*, and let $(\breve{x}^j, \breve{y}^j)$, $j = 1, \ldots, k$, be the history of iterates of IPCP. Then the objective value $\breve{v}^k = c^\intercal \breve{x}^k + d^\intercal \breve{y}^k$ satisfies*

$$\breve{v}^k - v \leqslant \tfrac{1}{2}\left(\|d\|_1 + \|(c,d)\|_2 \, \gamma^k \max\{\|(\|\beta\|_1)\|_\infty, 1, \max_{j=1,\ldots,k} \|\nabla_y g_{\ell_j}(\breve{x}^j, \breve{y}^j)\|_1\}\right).$$

Note that the upper bound from Theorem 5.4.3 may increase in each iteration. However, this is in accordance with the fact that in the IPCP, as in any outer approximation method, one has to expect increasing objective values of the iterates.

In view of the mentioned effort to compute Hoffman constants γ^k, one may not expect to be able to compute the upper bound from Theorem 5.4.3 explicitly. However, a possible qualitative interpretation is that one may expect small deviations of the iterates' objective values from v if, firstly, the vectors c, d, and $\|\beta\|_1$ are sparse and possess small entries and, secondly, if one generates sparse cuts with small entries.

For a comparison of the two bounds from Theorem 5.4.2 and Theorem 5.4.3 observe that with the valid Lipschitz constants

$$L_\infty^{g_i} := \max_{(x,y)\in D^0} \|\nabla_y g_i(x,y)\|_1,\ i \in I,$$

and due to $(\breve{x}^k, \breve{y}^k) \in D_\delta^0$, $k \in \mathbb{N}_0$, we obtain

$$\left\|\nabla_y g_{\ell_k}(\breve{x}^k, \breve{y}^k)\right\|_1 \leqslant L_\infty^{g_{\ell_k}}$$

for each k. Under the considerably weaker assumptions of Theorem 5.4.3 this yields the upper bound

$$\breve{v}^k - v \leqslant \tfrac{1}{2}\left(\|d\|_1 + \|(c,d)\|_2\, \gamma^k \max\{\|(\|\beta\|_1)\|_\infty, 1, \left\|L_\infty^g\right\|_\infty\}\right)$$

which closely resembles the one from Theorem 5.4.2. The involved Hoffman constants γ^k depend, however, on the current iteration, so that in this sense it is not an a-priori bound. Moreover, examples show that the estimate $\gamma^k \leqslant \gamma$ with γ from (5.12) does not necessarily hold. Whenever it does hold, the upper bound from Theorem 5.4.3 is not only based on less restrictive assumptions, but also better then the one from Theorem 5.4.2.

5.5 Computational Study

With our computational study we wish to shed light on the potential of the IPCP for computing good feasible points as well as for enhancing polyhedral outer approximations for MICPs. To do so, we first intend to clarify if the method is able to compute feasible points for problems from the literature, or if most of them exhibit a non-granular structure. This is done in Section 5.5.1, where we present problems for which the method is able to compute a feasible point. For these problems, we evaluate the computational cost of the algorithm and the quality of the generated feasible points and cutting planes.

Subsequently, we study non-granular and inconsistent problems in Sections 5.5.2 and 5.5.3, respectively. The analysis of these sections will not only reveal non-granular problems from the library, but also evaluate the difficulty of the numerical certificate using the IPCP. In this numerical study we again set $\delta := 1 - 10^{-4}$ and consider a problem to be granular if $\widehat{M}_\delta \neq \varnothing$ holds.

Our computational study comprises 139 consistent and 6 inconsistent convex problems from the MINLPLib [17]. We have collected problems which contain at least one integer variable and where all inequality constraint functions are proven to be convex. As the main intention of this study is to establish the validity of the algorithm and to show its potential, we have excluded such problems from our test bed

and postpone a thorough numerical study on the influence of equality constraints on granularity to Chapter 6.

We have implemented the IPCP in Python 3.7. We used the Pyomo framework [36] for obtaining and modifying the (non)linear models and *Clp* and *Ipopt* from the Coin infrastructure [21] for solving the appearing linear and nonlinear subproblems.

We initially ran our experiments with a feasibility tolerance of $\epsilon = 10^{-6}$ but observed difficulties with a few ill-conditioned LPs, where *Clp* returned a point that was not ϵ-feasible for the sub-LP, which forced the IPCP to stall (cf. Remark 5.3.9). Setting ϵ to $2 \cdot 10^{-6}$ the method terminated for these problems and we decided to use this tolerance throughout our computational study. As we shall report shortly, almost all generated feasible points were nonetheless 10^{-6}-feasible, which is a more common feasibility tolerance for MICPs. Moreover, for some (potentially ill-conditioned) problems, when applying the postprocessing step, *Ipopt* returned points that were more than $2 \cdot 10^{-6}$ infeasible, evaluated in our Python environment. Whenever this was the case, we used the point obtained by the IPCP without postprocessing.

All tests were run on a computer with an Intel i7 processor with 8 cores with 3.60 GHz and 32 GB of RAM. The source code and the data of our experiments are publicly available under `https://github.com/ChristophNeumann/IPCP`.

5.5.1 Instances with Feasible Points

In this section we study problems where the IPCP yields a feasible point. The main focus will be answering the question if the method has the potential to successfully support (OA-based) solvers. In our analysis, we have three main questions in mind: (i) does the IPCP converge quickly for practical problems, (ii) are the feasible points helpful in the sense that they are not easily obtainable with OA-based methods, and (iii) can the feasible points and cutting planes help these methods to converge more quickly.

We use the publicly available OA-based methods *B-OA* and *B-Hyb* implemented in Bonmin 1.8.7 [13] for this evaluation as well as *SCIP* 7.0.1 [30], compiled with *Ipopt* and SOPLEX, which is also publicly available. All methods (including the IPCP) use *Ipopt 3.12.12* as NLP-Solver and are equipped with non-commercial LP-Solvers, which enables a balanced comparison.

In Section 5.3.4 we discussed that if the only source of nonlinearity stems from the objective function, we may just terminate at any iteration with a feasible point. In fact, this was the case for several problems from our test bed. For these problems we terminate if we do not make progress in the objective value of the iterates for 20 iterations and return the best iterate. We shall add some further remarks for these problems at the end of this section.

We report that the IPCP is able to compute feasible points for 67 out of 139 tested consistent problems from the MINLPLib. We list these problems in Table 5.1, where we also state the number of (integer) variables, the number of (nonlinear) constraints and the optimal objective value v which is listed on the MINLPLib website [54].

The subscript epi means that we applied the epigraph reformulation to the problem, including a valid lower bound for the epigraph variable that we obtained by solving the convex relaxation of $MICP$, before applying the IPCP. For each problem, we further report the objective value of the computed feasible point (obj), the lower bounds v^k_{MILP} and v^0_{MILP} (cf. (5.8)), the number of iterations (iter), the maximum constraint violation of the point computed by the IPCP (g_{max}), and the cumulative run

time (time, in seconds) of all auxiliary problems, which includes the postprocessing step from Remark 5.3.2.

Moreover, to provide context of the quality of the obtained feasible points and to answer questions (ii) and (iii), we list the time needed by *B-Hyb*, *B-OA* and *SCIP* to compute a feasible point of similar quality. We obtained these values by running all methods with default settings and a time limit of 30 minutes, and searching the log-files for the first time a feasible point with an objective value is reported that is at least as good as that of the IPCP. To let *SCIP* know that all constraint functions are convex, we added the recommended option *assumeconvex=TRUE*.

For the problems *ex4, nvs10, st_miqp2 and st_test4*, we observed inconsistencies either with *B-Hyb* or *B-OA*. In the epigraph reformulated version that we used when applying the IPCP, these models were correctly solved and we report the corresponding values in Table 5.1.

problem data			IPCP							time solvers		
	variables	constraints	v	obj	v^k_{MILP}	v^0_{MILP}	iter	g_{max} (10^{-6})	time	*B-Hyb*	*B-OA*	*SCIP*
cvxnonsep_normcon20	20(10)	1(1)	-21.7	-20.8	-23.3	-41.4	46	0.0	0.4	14.5	36.3	0.3
cvxnonsep_normcon20r	40(10)	21(20)	-21.7	-20.6	-41.4	-41.4	157	1.0	1.4	0.2	0.0	0.2
cvxnonsep_normcon30	30(15)	1(1)	-34.2	-33.0	-38.7	-82.3	82	0.0	0.8	>1800	>1800	1.7
cvxnonsep_normcon30r	60(15)	31(30)	-34.2	-31.5	-82.3	-82.3	238	1.0	2.1	0.1	0.0	0.2
cvxnonsep_normcon40	40(20)	1(1)	-32.6	-31.1	-37.0	-89.4	181	0.0	2.0	>1800	>1800	4.2
cvxnonsep_normcon40r	80(20)	41(40)	-32.6	-29.7	-67.6	-89.4	289	1.0	2.7	0.2	0.0	0.2
cvxnonsep_nsig20	20(10)	1(1)	80.9	82.2	77.0	10.9	201	0.0	2.0	16.2	0.7	6.7
cvxnonsep_nsig20r	40(10)	21(20)	80.9	82.6	10.9	10.9	146	0.0	1.3	0.1	0.0	0.0
cvxnonsep_nsig30	30(15)	1(1)	130.6	132.2	124.1	16.9	461	0.0	6.6	>1800	20.4	108.0
cvxnonsep_nsig30r	60(15)	31(30)	156.4	157.5	16.9	16.9	192	0.0	1.7	10.3	0.1	0.0
cvxnonsep_nsig40	40(20)	1(1)	134.0	134.9	127.4	18.0	801	0.0	18.3	319.8	13.6	>1800
cvxnonsep_nsig40r	80(20)	41(40)	134.0	134.5	18.0	18.0	263	0.0	2.4	327.8	0.0	0.1
cvxnonsep_pcon20	20(10)	1(1)	-21.5	-20.6	-23.6	-49.8	123	0.0	1.1	0.1	0.2	0.1
cvxnonsep_pcon20r	39(10)	20(19)	-21.5	-19.7	-49.8	-49.8	149	1.5	1.3	0.0	0.0	0.1
cvxnonsep_pcon30	30(15)	1(1)	-36.0	-34.0	-38.8	-77.5	207	0.0	2.2	1.3	4.0	1.4
cvxnonsep_pcon30r	59(15)	30(29)	-36.0	-33.6	-77.5	-77.5	239	1.8	2.2	0.1	0.0	0.1
cvxnonsep_pcon40	40(20)	1(1)	-46.6	-44.4	-49.7	-100.8	336	0.0	4.4	1337.0	2.2	2.3
cvxnonsep_pcon40r	79(20)	40(39)	-46.6	-43.6	-100.8	-100.8	307	2.3	2.9	0.1	0.0	0.0
cvxnonsep_psig20r	42(10)	22(21)	95.9	96.7	20.0	20.0	146	-0.0	1.3	0.1	0.0	0.0
cvxnonsep_psig30r	62(15)	32(31)	79.0	81.9	30.0	30.0	184	-0.0	1.6	0.1	0.0	0.1
cvxnonsep_psig40r	82(20)	42(41)	86.5	88.9	40.0	40.0	219	-0.2	2.0	0.1	0.0	0.1
du-opt$_{\text{epi}}$	21(13)	10(1)	3.6	4.7	3.3	3.3	26	0.0	0.2	10.1	0.0	0.1
ex1223a$_{\text{epi}}$	8(4)	10(5)	4.6	7.6	4.5	4.5	6	-120006.3	0.1	0.0	0.0	0.0
ex1223b$_{\text{epi}}$	8(4)	10(5)	4.6	8.0	3.9	3.9	9	-0.1	0.1	0.0	0.0	0.0
ex4$_{\text{epi}}$	37(25)	31(26)	-8.1	-4.4	-9.7	-16.7	24	0.0	0.2	0.2	0.2	0.0
gbd$_{\text{epi}}$	5(3)	5(1)	2.2	2.2	2.2	2.2	2	0.0	0.0	0.0	0.0	0.0
nvs03$_{\text{epi}}$	3(2)	3(2)	16.0	16.0	16.0	8.2	5	-400000.0	0.0	0.0	0.0	0.0
nvs10$_{\text{epi}}$	3(2)	3(3)	-310.8	-303.4	-313.1	-313.1	8	0.0	0.1	0.0	0.0	0.0

Continued on next page

problem data			IPCP							time solvers		
	variables	constraints	v	obj	v^k_{MILP}	v^0_{MILP}	iter	g_{max} (10^{-6})	time	*B-Hyb*	*B-OA*	*SCIP*
$nvs11_{epi}$	4(3)	4(4)	-431.0	-427.8	-432.8	-432.8	11	0.0	0.1	0.0	0.0	0.0
$nvs12_{epi}$	5(4)	5(5)	-481.2	-467.5	-483.2	-483.2	18	0.0	0.2	0.0	0.0	0.0
$nvs15_{epi}$	4(3)	2(1)	1.0	1.0	0.1	0.1	4	0.0	0.0	0.0	0.0	0.0
portfol_buyin	17(8)	19(2)	0.0	0.1	0.0	0.0	2	0.0	0.0	0.0	0.0	0.0
smallinvDAXr1b150-165	31(30)	4(1)	88.1	102.3	86.2	-inf	34	0.0	0.3	0.1	2.3	0.0
smallinvDAXr1b200-220	31(30)	4(1)	156.6	175.1	154.2	-inf	34	0.0	0.3	10.1	10.5	0.0
smallinvDAXr2b150-165	31(30)	4(1)	88.1	101.3	86.4	-inf	34	0.0	0.3	10.1	8.5	0.0
smallinvDAXr2b200-220	31(30)	4(1)	156.6	175.4	154.2	-inf	34	0.0	0.3	10.0	4.1	0.0
smallinvDAXr3b150-165	31(30)	4(1)	88.1	101.1	86.6	-inf	34	0.0	0.3	16.0	4.7	0.0
smallinvDAXr3b200-220	31(30)	4(1)	156.6	175.5	155.2	-inf	38	0.0	0.4	0.3	3.6	0.0
smallinvDAXr4b150-165	31(30)	4(1)	88.1	102.4	86.2	-inf	32	0.0	0.3	10.3	5.6	0.0
smallinvDAXr4b200-220	31(30)	4(1)	156.6	174.0	153.9	-inf	28	0.0	0.2	10.1	2.6	0.0
smallinvDAXr5b150-165	31(30)	4(1)	88.1	101.9	85.8	-inf	33	0.0	0.3	10.2	4.1	0.0
smallinvDAXr5b200-220	31(30)	4(1)	156.6	175.9	154.5	-inf	34	0.0	0.3	3.4	7.2	0.0
$squfl010\text{-}025_{epi}$	261(10)	276(1)	214.1	214.1	105.9	105.9	15	0.0	0.2	0.7	8.0	294.0
$squfl010\text{-}040_{epi}$	411(10)	441(1)	240.6	294.5	136.8	136.8	44	0.0	1.1	10.7	1.2	25.4
$squfl010\text{-}080_{epi}$	811(10)	881(1)	509.7	581.6	258.9	258.9	33	0.0	2.0	12.9	4.1	83.2
$squfl015\text{-}060_{epi}$	916(15)	961(1)	366.6	417.8	152.5	152.5	16	0.0	0.5	11.3	217.6	108.0
$squfl015\text{-}080_{epi}$	1216(15)	1281(1)	402.5	493.3	172.6	172.6	25	0.0	2.5	13.2	373.7	348.0
$squfl020\text{-}040_{epi}$	821(20)	841(1)	209.3	272.4	98.1	98.1	7	0.0	0.2	14.4	45.8	0.0
$squfl020\text{-}050_{epi}$	1021(20)	1051(1)	230.2	356.7	99.2	99.2	15	0.0	0.5	12.1	78.1	0.1
$squfl020\text{-}150_{epi}$	3021(20)	3151(1)	557.8	636.4	226.3	226.3	24	0.0	7.5	250.7	>1800	888.0
$squfl025\text{-}025_{epi}$	651(25)	651(1)	168.8	244.4	68.0	68.0	2	0.0	0.1	10.3	28.8	0.0
$squfl025\text{-}030_{epi}$	776(25)	781(1)	205.5	215.3	81.3	81.3	2	0.0	0.1	10.3	53.3	0.0
$squfl025\text{-}040_{epi}$	1026(25)	1041(1)	197.3	248.4	76.9	76.9	9	0.0	0.3	10.6	378.5	0.1
$squfl030\text{-}100_{epi}$	3031(30)	3101(1)	363.1	425.9	123.9	123.9	8	0.0	0.8	318.3	>1800	0.1
$squfl030\text{-}150_{epi}$	4531(30)	4651(1)	430.6	628.1	158.9	158.9	22	0.0	8.1	52.7	>1800	0.2
$squfl040\text{-}080_{epi}$	3241(40)	3281(1)	263.9	431.0	91.5	91.5	14	0.0	1.9	52.9	>1800	0.1

Continued on next page

problem data			IPCP							time solvers		
	variables	constraints	v	obj	v^k_{MILP}	v^0_{MILP}	iter	g_{max} (10^{-6})	time	*B-Hyb*	*B-OA*	*SCIP*
st_miqp1$_{epi}$	6(5)	2(1)	281.0	378.0	231.0	228.0	2	0.0	0.0	0.0	0.0	0.0
st_miqp2$_{epi}$	5(4)	4(1)	2.0	2.0	-1.0	-5.6	5	0.0	0.1	0.1	0.0	0.0
st_miqp3$_{epi}$	3(2)	2(1)	-6.0	0.0	-6.0	-6.0	4	0.0	0.0	0.0	0.0	0.0
st_miqp4$_{epi}$	7(3)	5(1)	-4574.0	-4574.0	-4574.0	-4885.9	4	0.0	0.0	10.0	0.0	0.0
st_test1$_{epi}$	6(5)	2(1)	0.0	0.0	-47.5	-63.3	5	0.0	0.1	0.0	0.0	0.0
st_test2$_{epi}$	7(6)	3(1)	-9.2	-7.2	-10.8	-25.1	2	0.0	0.0	10.0	0.0	0.0
st_test3$_{epi}$	14(13)	11(1)	-7.0	-5.0	-12.0	-12.9	3	0.0	0.0	10.0	0.0	0.0
st_test4$_{epi}$	7(6)	6(1)	-7.0	-5.0	-7.0	-12.5	3	0.0	0.0	0.0	0.0	0.0
st_testgr1$_{epi}$	11(10)	6(1)	-12.8	-12.7	-12.9	-13.4	16	0.0	0.1	0.1	0.0	0.0
st_testgr3$_{epi}$	21(20)	21(1)	-20.6	-20.2	-20.8	-22.1	4	0.0	0.0	10.0	0.0	0.0
st_testph4$_{epi}$	4(3)	11(1)	-80.5	-80.5	-80.5	-83.0	4	0.0	0.0	10.0	0.0	0.0

TABLE 5.1: Problems for which the IPCP computes a feasible point

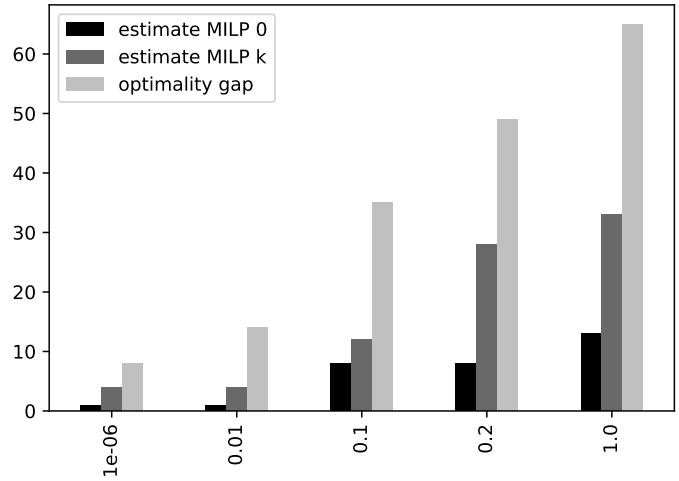


FIGURE 5.3: Cumulative histogram of (estimates of) the optimality gap (5.18) of the feasible points

Let us next summarize our findings. First, notice that the IPCP terminates in less than 20 seconds with a relatively small number of sub LPs for all tested problems. We remark that using the Pyomo interface, we were not able to exploit warm start capabilities of *Clp* but solved each LP from scratch. Hence we expect that the method could converge significantly faster when carefully integrated into a solver.

Secondly, even though we set ϵ to $2 \cdot 10^{-6}$, in all but 3 cases we obtained 10^{-6}-feasible points, either due to the postprocessing step of *Ipopt*, or due to the method itself. Hence, almost all computed points can be used by solvers with a usual feasibility tolerance. For the problem *cvxnonsep_pcon40r*, a linear constraint is violated by more than $2 \cdot 10^{-6}$ and hence this value is greater than the anticipated feasibility tolerance.

Moreover, the objective values are remarkably close to the optimal ones in almost all cases which indicates that they might provide valuable information. Indeed, the cumulative histogram in Figure 5.3 shows that the optimality gap

$$(\breve{v}^k - v) / \min\{|\breve{v}^k|, |v|\} \tag{5.18}$$

is less than 10% for more than half of the problems. This figure further illustrates that the estimated gap obtained by replacing v with v^k_{MILP}, which is available after termination of the algorithm by solving an MILP, is also very small in many instances. This gap is often significantly better compared to the one obtainable by $\overline{MILP}^0$. Hence, this analysis not only proves the good quality of the generated feasible points, but also indicates that the RIPCs could help to outer approximate these problems well and to thus potentially speed up the solution process of MICP-solvers.

Moreover, the biased performance profile in Figure 5.4 reveals that for several instances, the same objective value is not easily obtained by *B-OA*, *B-Hyb* and *SCIP*. These solvers are by default equipped with an arsenal of primal heuristics which are thus all included in this comparison. While (the collections of) these primal heuristics are able to compute feasible points of similar quality quickly for many problems from our test bed (especially those implemented in *SCIP*), for several problems they take orders of magnitude longer, or even entirely fail to compute such a point within 30 minutes.

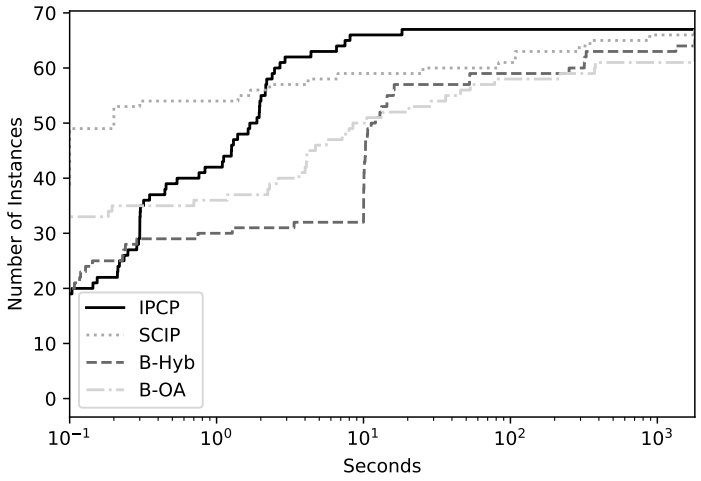


FIGURE 5.4: Biased performance profile: Time (log scale) needed for SCIP and the two Bonmin methods to compute a feasible point of similar quality as that of the IPCP

We wish to point out that the performance profile is necessarily *biased* in the sense that we "compare" methods with different purposes and that *B-OA*, *B-Hyb* and *SCIP* do much more than just computing a feasible point in that time. Yet, the point is *not* to state that the IPCP outperforms these methods, but rather to observe that in several cases, feasible points can provide useful information for providing upper bounds and cutting planes.

To address question (iii) more fully, we ran a second experiment, testing *B-Hyb*, *B-OA* and *SCIP* on modified models. We created the modified models according to Section 5.3.3, adding the last m reversed inner parallel cuts (RIPCs) along with the other proposed polyhedral constraints, and thus "prototyped" a straightforward integration of the IPCP into OA-based methods. We report results for solving times and lower bounds after (a possibly early) termination in Table A.3 in the appendix. Let us next give a quick summary.

Firstly, the reversed inner parallel cuts were able to significantly speed up all solvers for *some* instances. Secondly, the speedup was very problem- and solver-specific and sometimes the solution process was actually prolonged by the cuts so that a careful investigation is needed, when integrating the RIPCs into a solver. Generally, the potential is most apparent with *B-Hyb*, where we were able to correctly solve 14 additional problems using reversed inner parallel cuts. Yet we also observed some solver errors and false termination statements when using *B-Hyb* (with and without RICPs). Therefore, we would be cautious to place too much emphasis on these *B-Hyb* results. For *SCIP* and *B-OA* the results were generally stable; both methods solved the same problems independently of added RIPCs. Yet, in several cases the solving times differed significantly.

Some examples of problems where the RIPCs had a significant influence on the solving times are given in Table 5.2, where we list the run times of all methods for proving optimality for models with and without RICPs. While the times for running the IPCP are not included in this analysis, they are comparably small (cf. Table 5.1) and hence do not contribute to the effects we demonstrate. For instance the problem *cvxnonsep_nsig20* is solved much faster by *SCIP* and by *B-Hyb* after RICPs are added. The two *smallinvDAX* instances along with *cvxnonsep_normcon20* are examples of instances which are correctly solved by *B-Hyb* only if RICPs are added. All four cases

	Scip		*B-Hyb*		*B-OA*	
	RIPC	OM	RIPC	OM	RIPC	OM
cvxnonsep_normcon20	1.44	1.33	65.32	>1800	14.47	98.94
cvxnonsep_normcon40	601.36	681.81	>1800	>1800	>1800	>1800
cvxnonsep_nsig20	2.14	272.68	156.60	1261.60	14.80	10.06
du-opt	0.89	5.86	24.93	12.62	3.01	2.95
smallinvDAXr2b200-220	0.31	0.36	99.74	>1800	130.84	143.27
smallinvDAXr5b200-220	0.44	0.38	56.77	>1800	105.79	171.73
squfl010-025	1049.77	413.89	10.73	1.05	295.49	418.10

TABLE 5.2: Time (seconds) needed to solve an instance with and without RIPCs

	Scip		*B-Hyb*		*B-OA*	
	RIPC	OM	RIPC	OM	RIPC	OM
cvxnonsep_nsig30	128.07	121.17	130.48	130.48	130.60	130.61
cvxnonsep_nsig40	130.57	94.71	133.74	133.74	133.74	133.85
squfl020-150	226.34	0.00	297.58	286.17	229.32	227.04
squfl030-100	123.89	0.00	151.99	151.69	127.51	126.47

TABLE 5.3: Computed lower bounds after 1800 seconds run time with and without RIPCs

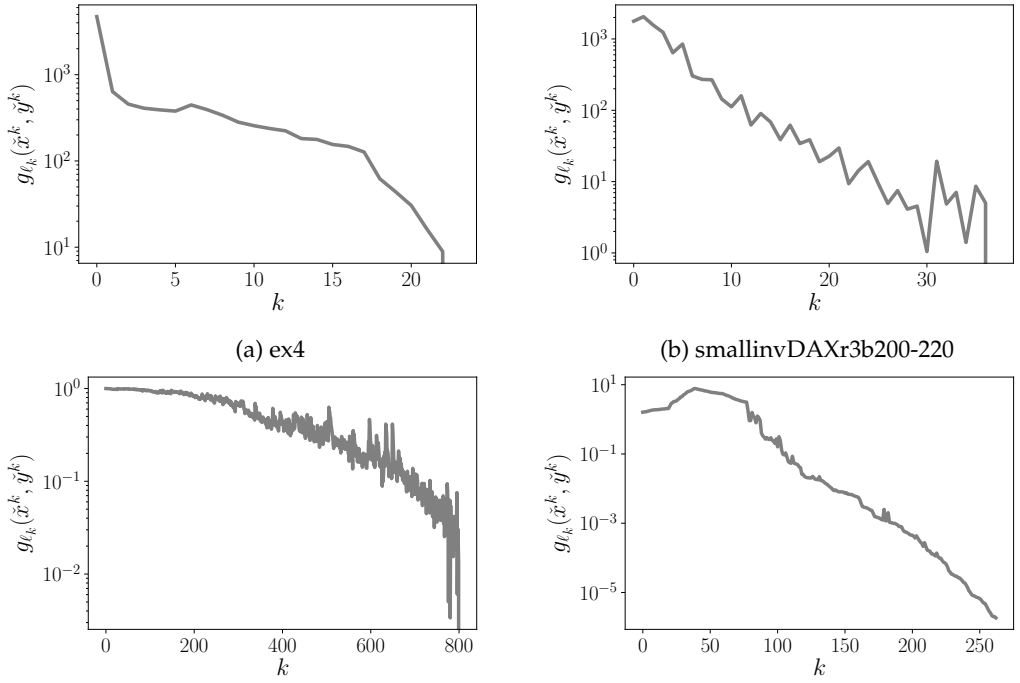
further illustrate that the usefulness of the RICPs strongly depends on the interplay between the solver and the specific instance. Finally, with the example *squfl010-025* we wish to point out that occasionally problems were solved significantly faster by some solvers when no RICPs were added. This further highlights the importance of carefully selecting those cuts that actually help the solver-specific solution process.

Table 5.3 complements this analysis by stating computed lower bounds for problems where all solvers failed to terminate within 30 minutes, stating the best bounds of the methods upon termination. In almost all cases, the bounds improve when RIPCs are added to the model.

Remarkably, for *cvxnonsep_nsig30* and *cvxnonsep_nsig40*, the lower bounds obtained by solving $\overline{MILP}^k$ are already better than the ones *SCIP* generates after 30 minutes (cf. Table 5.1). Hence, taking all results into account, we can at the very least state that the feasible points and cutting planes have the *potential* to speed up the solution process of OA-based methods.

Let us conclude this section with some remarks on the algorithm's behavior on several problems with distinct characteristics. Figure 5.5 shows (in log scale) the maximum constraint violation, $g_{\ell_k}(\tilde{x}^k, \tilde{y}^k)$, over the iterates k for the problems *ex4*, *smallinvDAXr3b200-220*, *cvxnonsep_nsig40*, and *cvxnonsep_nsig40r*.

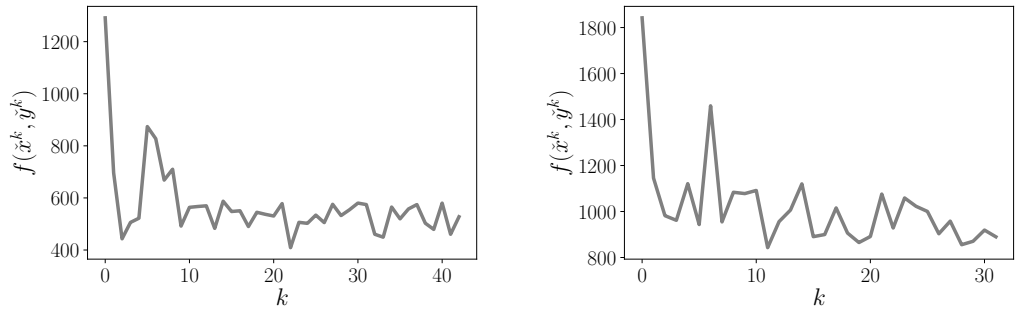
While the decrease shown in Figures 5.5a and 5.5b appears to be (almost) monotonic, Figure 5.5c shows large fluctuations from a trend of decreasing feasibility error. In Figure 5.5d the constraint violation even increases quite strongly within the first 40 iterations, eventually approaching zero smoothly after 100 iterations. Note that, even though the behavior is quite different for the distinct problems, Figure 5.5 reveals a monotonically decreasing pattern (with cyclic fluctuations), at least after a certain number of iterations. Hence, we indeed observe that the IPCP does not compute a feasible point "by accident", but rather by systematically reducing the feasibility error.



(a) ex4 (b) smallinvDAXr3b200-220

(c) cvxnonsep_nsig40 (d) cvxnonsep_nsig40r

FIGURE 5.5: Maximum constraint violation $g_{\ell_k}(\check{x}^k, \check{y}^k)$ over iterations



(a) squfl010-040 (b) squfl1010-080

FIGURE 5.6: Progress in the objective value before applying the post processing step for problems where the algorithm terminates early

Moreover, Figure 5.6 plots the progress in the objective value of the iterates for two problems where the only source of nonlinearity stems from the objective function. In contrast to the progress towards feasibility, we only occasionally observe an improvement after a certain amount of cutting planes have been added. Hence, stopping the algorithm early indeed appears to be reasonable for these problems.

5.5.2 Non-granular Consistent Instances

We will now turn towards a less fortunate case: to consistent problems where the method reports non-granularity. This will gather information about the computational costs of the numerical certificate for non-granularity which enables us to draw a more holistic picture of the potential benefits and the costs of the method. Furthermore, these results will reveal optimization problems for which we are unable to

instance	variables	constraints	iterations	time
$alan_{epi}$	9(4)	8(1)	1	0.02
ball_mk2_10	10(10)	1(1)	23	0.20
ball_mk2_30	30(30)	1(1)	103	1.03
$du\text{-}opt5_{epi}$	21(13)	10(1)	1	0.01
fo7	114(42)	211(14)	1	0.01
fo7_2	114(42)	211(14)	1	0.01
fo8	146(56)	273(16)	1	0.01
fo9	182(72)	343(18)	1	0.01
m3	26(6)	43(6)	1	0.01
m6	86(30)	157(12)	1	0.01
m7	114(42)	211(14)	1	0.01
$meanvarx_{epi}$	36(14)	45(1)	1	0.01
o7	114(42)	211(14)	1	0.01
o7_2	114(42)	211(14)	1	0.01
portfol_classical050_1	150(50)	103(1)	1	0.01
portfol_classical200_2	600(200)	403(1)	1	0.03
smallinvDAX*	31(30)	4(1)	1	0.01
smallinvSNP**	101(100)	4(1)	1	0.01
st_miqp5_{epi}	8(2)	14(1)	1	0.01
st_test5_{epi}	11(10)	12(1)	1	0.01
st_test6_{epi}	11(10)	6(1)	1	0.01
st_test8_{epi}	25(24)	21(1)	1	0.01
$synthes1_{epi}$	7(3)	7(3)	1	0.01
$unitcommit1_{epi}$	961(720)	5330(1)	1	0.06

TABLE 5.4: Computational cost of the non-granularity certificate for 24 consistent instances from the MINLPLib, where $\star$ and $\star\star$ are representative for 20 and 30 variations of problems, respectively

directly use the granularity concept for the computation of feasible points, at least without altering the model in some granularity-promoting way.

We obtain a certificate for non-granularity for 72 out of 139 consistent instances from our test bed. Among them are all 30 variations of the problem *smallinvSNP*, as well as 20 variations of the problem *smallinvDAX*. The number of iterations and the cumulative run times are quite similar for all these variations and hence we only list one representative instance each in Table 5.4, together with the remaining 22 non-granular instances.

Note that the IPCP detects non-granularity very quickly for all instances from our test bed. In fact, for almost all problems, already the initial outer approximation of the enlarged inner parallel set is empty and hence the algorithm terminates after only one iteration within a fraction of a second. This indicates that, especially when dealing with a non-granular problem, we do not need to wait long for the IPCP to terminate. Rather, we immediately obtain a certificate for the non-granularity and thus know that we have to use another method for computing a feasible point. This unfortunately also implies that we cannot create inner parallel cutting planes in most of these cases.

A slight exception are the two variations of the problem *ball_mk_2*. Here, the method needs 28 and 134 iterations respectively to prove non-granularity. We shall further elaborate the reason for this behavior in the next section by looking at the

structure of variations of this problem, which turns out to be particularly difficult for the application of the IPCP.

5.5.3 Inconsistent Instances

While it is trivial to see that inconsistent problems are not granular, the focus of this section will be comparing the cost of certifying non-granularity to the harder task of proving that a problem is inconsistent. We thus intend to reveal information about the performance of the method for unfavorable problems. Moreover, by examining these problems, we will also get a glimpse on the behavior of the IPCP when the initial linear outer approximation D becomes larger.

We tested the IPCP on several variations of the geometrical problem suggested in [38], where also the difficult consistent instances (*ball_mk_2*) from the previous section stem from. In the example given in [38] the feasible set is the intersection of (a subset of) $\mathbb{Z}^m$ with a ball with center $p = (1/2, \dots, 1/2)^\intercal$ and radius $r = \sqrt{m-1}/2$,

$$B(p,r) = \left\{ y \in \mathbb{Z}^m \mid \sum_{j=1}^{m} \left(y_j - \tfrac{1}{2}\right)^2 \leqslant \frac{m-1}{4},\ y^\ell \leqslant y \leqslant y^u \right\},$$

which is easily seen to be empty for any $y^\ell, y^u \in \mathbb{Z}^m$. Even for $y^\ell = 0, y^u = e$, any outer approximation method needs 2^m iterations to certify infeasibility of this set [38]. Note that in outer approximation algorithms for solving MICPs, each iteration involves the solution of an MILP and, hence, this problem is intractable for such methods even for a small dimension m.

Interestingly, this example is constructed in such a way that also non-granularity appears to be especially hard to certify due to the following reasons: first, the IPCP is closely linked to the extended cutting plane method and hence some variation of an outer approximation algorithm. Secondly, and even more importantly, the problem is *almost* granular. In fact, if we enlarged the radius to $r = \sqrt{m}/2$, the set $B(p,r)$ would not only become consistent but even granular, as the center $\frac{1}{2}e$ then also lies in the inner parallel set $(\widehat{B}(p,r))^-$. Certifying non-granularity for an almost granular problem appears to be one of the hardest tasks for the IPCP, so our sincere hope is that the following test sheds light on the algorithm's practical worst case behavior.

Table 5.5 lists results of the IPCP for six infeasible problems from the MINLPLib. Here, we try to minimize different linear objective functions over an empty feasible set which slightly varies from the above example. In particular, the ball is replaced by a general ellipsoid. The table also states the dimension m, as well as the common lower bound y_i^ℓ and the common upper bound y_i^u posed on all integer variables, $i = 1, \dots, m$.

Remarkably, the number of iterations needed to certify non-granularity is far from exponential. Moreover, we stress that for the last three problems the initial polyhedral outer approximation is quite coarse ($D = [-100, 100]^m$) compared to the first three problems ($D = [-1, 2]^m$). Although this clearly influences the number of iterations, the increase is not as significant as the increase of the size of D. Hence, the number of iterations of the IPCP appears to remain relatively small even for unfavorable problems where the initial polyhedral relaxation D is a bad approximation of the relaxed feasible set $\widehat{M}$.

Clearly, the task of certifying non-granularity is significantly easier and less informative compared to proving the inconsistency of a problem. However, one might have expected this to show up rather in the cumulative run time (as we need to only

	m	y_i^ℓ	y_i^u	iterations	time
ball_mk3_10	10	-1	2	31	0.24
ball_mk3_20	20	-1	2	77	0.63
ball_mk3_30	30	-1	2	142	1.37
ball_mk4_05	10	-100	100	168	1.37
ball_mk4_10	20	-100	100	524	6.71
ball_mk4_15	30	-100	100	958	21.29

TABLE 5.5: Performance for difficult non-granular (inconsistent) problems

solve LPs instead of MILPs) and not necessarily in the number of iterations. Therefore, our results demonstrate that even for problems with a relatively unfavorable structure, the IPCP runs quite quickly.

5.6 Conclusions

In this chapter we presented and analyzed an inner parallel cutting plane method for computing good feasible points along with valid cutting planes for mixed-integer convex optimization problems. The crucial advantage of this method is that it only needs to solve continuous linear subproblems. Compared to other methods from the literature, this results in significantly lower per iteration costs and, as our computational study reveals, also in a fast convergence for practical problems. The objective values of the generated feasible points are generally of good quality and often not easily obtainable by other methods and the generated cutting planes helped, in several instances, to significantly speed up convergence of outer approximation methods.

We remark that there exist consistent non-granular problems for which, instead of computing a feasible point, the method might only certify the non-granularity of the problem. In our computational study this was the case for roughly half of the problems. However, we emphasize that for these problems the method converges especially quickly and hence argue that the potential benefits of applying the method clearly exceed its costs.

Finally, we may adapt the IPCP in such a way that it mimics the outer approximation method it intends to support. In fact, instead of using Kelley's method as a basis, one could also investigate inner cutting plane methods that are inspired by other outer approximation methods. This vast subject is beyond the scope of this thesis and hence postponed to future research. The extension to the nonconvex case will also be the subject of future research.

Chapter 6

Equality Constraints and Inner Parallel Sets

The previous chapters assumed the absence of equality constraints that include integer variables in the formulation of the mixed-integer optimization problem. In this chapter we drop this assumption and investigate the possibilities of using inner parallel sets for obtaining feasible roundings under the occurrence of such equality constraints.

We might be tempted to intuitively argue that equality constraints on integer variables immediately imply an empty inner parallel set and thus render the concept useless for problems where such constraints occur. Section 6.1 demonstrates that this intuition is indeed true if we think of an equality constraint as two inequalities and limit our construction of an inner parallel set to the original variable space. However, it will turn out that there is the possibility of constructing a potentially nonempty inner parallel set in a reduced space.

This concept of reduction is further pursued and elaborated in Section 6.2 where we work out a general reduction scheme that is applicable to mixed-integer optimization problems with linear equality constraints. While the techniques for this approach are known (cf. e.g. [47, 65]), we did not find results in the literature that explicitly state this reduced MILP, and our hope is that this may also be helpful for other purposes.

In Section 6.3 we tailor these results to the application of feasible rounding approaches to MILPs with equality constraints on integer variables. Section 6.4 evaluates the elimination procedure and the application of feasible rounding approaches to equality constrained problems numerically and Section 6.5 concludes this chapter with some final remarks.

For simplicity and clarity of presentation, we focus our attention on MILPs. Yet we stress that the approach is equally applicable to all MI(NL)Ps that only contain linear equality constraints on integer variables, like it is the case for convex MINLPs.

6.1 Basic Ideas and Different Possibilities

In this chapter, we consider equality constrained optimization problems of the form

$$MILP: \quad \min_{(x,y)\in\mathbb{R}^n\times\mathbb{Z}^m} c^\intercal x + d^\intercal y \quad \text{s.t.} \quad \begin{aligned} Ax + By &\leqslant b \\ Cx + Dy &= \gamma \end{aligned}$$

with vectors $c \in \mathbb{R}^n$, $d \in \mathbb{R}^m$, $b \in \mathbb{R}^p$, $\gamma \in \mathbb{R}^q$, a (p,n)-matrix A, a (p,m)-matrix B, a (q,n)-matrix C and a (q,m)-matrix D, where C and D possess rational entries.

In view of the formulation of the enlarged inner parallel set, the granularity concept and the resulting possibility to obtain feasible points of an $MILP$ at first glance seem to be restricted to problems without equality constraints. Two obvious possibilities to incorporate equality constraints are their reformulation as pairs of inequality constraints, and their elimination by explicit computation of their solution space. In the present section we illustrate that the former approach is compatible with granularity only in special cases, while the latter approach bears a potential also for general equality constraints.

In fact, as mentioned earlier, equality constraints that are posed *only* on continuous variables may be remodeled as pairs of inequality constraints while maintaining the possibility of granularity. Recall the enlarged inner parallel set from (3.7) which, in the present $MILP$ setting, reads

$$\begin{aligned}\widehat{M}_\delta^- = \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m |\ Ax + By \leqslant \lfloor b \rfloor_\omega + \delta\omega - \tfrac{1}{2}\left\|\beta\right\|_1, \\ y^\ell + (\tfrac{1}{2} - \delta)e \leqslant y \leqslant y^u + (\delta - \tfrac{1}{2})e\}\end{aligned} \tag{6.1}$$

and assume that an equality constraint i posed only on continuous variables is remodeled as a pair of inequality constraints i_1 and i_2. Then, due to $\omega_{i_1} = \omega_{i_2} = 0$, for the resulting inequality constraints we have $\lfloor b_{i_1} \rfloor_{\omega_{i_1}} = b_i$, $\lfloor b_{i_2} \rfloor_{\omega_{i_2}} = -b_i$ and $\frac{1}{2}\left\|\beta_{i_1}\right\|_1 = \frac{1}{2}\left\|\beta_{i_2}\right\|_1 = 0$ and $\delta\omega_{i_1} = \delta\omega_{i_2} = 0$, and hence their right-hand sides remain unchanged in the transition to the enlarged inner parallel set. Therefore, this construction does not rule out that $\widehat{M}_\delta^-$ is nonempty for some $\delta \in [0,1)$.

For equality constraints including integer variables, this is usually not the case. Indeed, we will now introduce conditions under which this simple treatment of equality constraints immediately results in an empty set $\widehat{M}_\delta^-$ for any $\delta \in [0,1)$.

Proposition 6.1.1. *For some i_1 and i_2, let $\alpha_{i_1} = -\alpha_{i_2}$, $\beta_{i_1} = -\beta_{i_2}$ and $b_{i_1} = -b_{i_2}$ hold with $\beta_{i_1} \neq 0$ (that is, an equality constraint with integer variables is modeled as two inequality constraints). Moreover, let at least one of the following conditions hold:*

i) $\alpha_{i_1} \neq 0$,

ii) β_{i_1} contains at least two entries.

Then, the enlarged inner parallel set $\widehat{M}_\delta^-$ is empty for any $\delta \in [0,1)$.

Proof. In view of $\alpha_{i_1} = -\alpha_{i_2}$ and $\beta_{i_1} = -\beta_{i_2}$, adding the rows i_1 and i_2 in the constraint system describing $\widehat{M}_\delta^-$ yields

$$\begin{aligned}0 &\leqslant \lfloor b_{i_1} \rfloor_{\omega_{i_1}} + \lfloor b_{i_2} \rfloor_{\omega_{i_2}} - \tfrac{1}{2}(\left\|\beta_{i_1}\right\|_1 + \left\|\beta_{i_2}\right\|_1) + \delta(\omega_{i_1} + \omega_{i_2}) \\ &\leqslant b_{i_1} + b_{i_2} - \tfrac{1}{2}(\left\|\beta_{i_1}\right\|_1 + \left\|\beta_{i_2}\right\|_1) + \delta(\omega_{i_1} + \omega_{i_2}).\end{aligned}$$

By construction, this is the case if and only if

$$\left\|\beta_{i_1}\right\|_1 \leqslant 2\delta\omega_{i_1}$$

holds. With i) this leads to a contradiction due to $\omega_{i_1} = 0$ and $\left\|\beta_{i_1}\right\|_1 > 0$. If ii) holds, the contradiction follows from $\delta < 1$, along with the fact that β_{i_1} contains at least two entries each of which upper bounds ω_{i_1}. □

Proposition 6.1.1 shows the impossibility of remodeling equality constraints as two inequality constraints under the preservation of the potential for granularity, if at least two variables appear in the equation and at least one of them is integer. Let

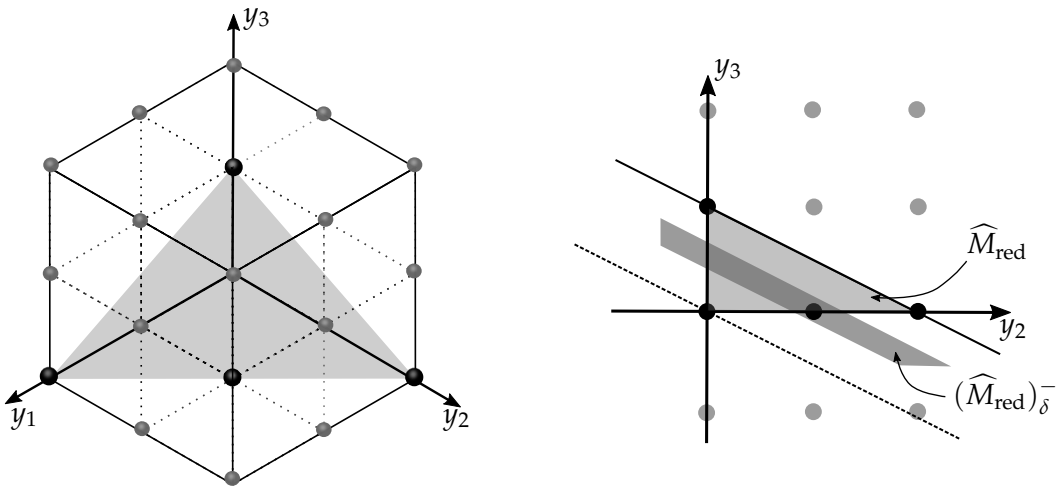


FIGURE 6.1: Eliminating the equality constraint from the original feasible set (left) results in a nonempty inner parallel set in the reduced space (right)

us briefly examine the case where exactly one integer variable j appears in equation i (and no continuous variables). Without loss of generality, let β_j be the j-th unit vector. Then, with $\alpha_i = 0$, y_j is fixed to the value b_i, and a reformulation as two inequality constraints is possible while maintaining the possibility for a nonempty inner parallel set.

Indeed, reformulating $y_j = b_i$ as two inequalities yields the constraints

$$-\lfloor -b_i \rfloor + \tfrac{1}{2} - \delta \leqslant y_j \leqslant \lfloor b_i \rfloor - \tfrac{1}{2} + \delta$$

for the enlarged inner parallel set. Due to

$$-\lfloor -b_i \rfloor = \begin{cases} \lfloor b_i \rfloor, & b_i \in \mathbb{Z} \\ \lfloor b_i \rfloor + 1, & \text{otherwise,} \end{cases}$$

for $\delta \geqslant \frac{1}{2}$ these constraints are attainable if and only if $b_i \in \mathbb{Z}$, exactly as it is the case for the original equality constraint. Of course, in applications and from an algorithmic perspective, it makes sense to eliminate such fixed variables from the problem formulation.

The above discussion shows in particular that conditions i) and ii) in Proposition 6.1.1 are actually necessary to prove that inconsistency of $\widehat{M}^-_\delta$ follows from the reformulation of an equality constraint as a pair of inequalities.

The next example illustrates how, rather than the reformulation as a pair of inequalities, also an elimination step may be possible for a general equality constraint containing integers, and how the granularity concept can benefit from it.

Example 6.1.2. *Consider the feasible set*

$$M = \{y \in \mathbb{Z}^3 \mid 0 \leqslant y \leqslant 2e,\ y_1 + y_2 + 2y_3 = 2\},$$

which is illustrated on the left-hand side of Figure 6.1. Proposition 6.1.1 implies that reformulating the equality constraint as two inequalities would rule out granularity. Notice, however, that we can actually eliminate y_1 using the substitution

$$y_1 = 2 - y_2 - 2y_3 \tag{6.2}$$

while guaranteeing $y_1 \in \mathbb{Z}$ if $y_2, y_3 \in \mathbb{Z}$. Also note that eliminating y_3 would not be possible in the same manner. This yields the reduced feasible set

$$M_{red} = \{y(:= (y_2, y_3)^\intercal) \in \mathbb{Z}^2 |\ 0 \leqslant y \leqslant 2e,\ y_2 + 2y_3 \leqslant 2,\ y_2 + 2y_3 \geqslant 0\}$$

and the corresponding enlarged inner parallel set

$$(\widehat{M}_{red})^-_\delta = \{y \in \mathbb{R}^2 |\ (\tfrac{1}{2} - \delta)e \leqslant y \leqslant (\tfrac{3}{2} + \delta)e,\ y_2 + 2y_3 \leqslant \tfrac{1}{2} + \delta,\ y_2 + 2y_3 \geqslant \tfrac{3}{2} - \delta\}.$$

Now, any rounding from $(\widehat{M}_{red})^-_\delta$ yields a feasible point for M_{red} which, together with (6.2)*, yields a point in M.*

The set $(\widehat{M}_{red})^-_\delta$ is illustrated on the right-hand side of Figure 6.1 with $\delta = 0.9$. Remarkably, now actually any point from M_{red} and thus any point in M can be obtained by rounding a point from $(\widehat{M}_{red})^-_\delta$.

Example 6.1.2 shows the potential of the granularity concept also for problems with equality constraints, if we eliminate variables while ensuring integrality conditions for eliminated integer variables.

Remark 6.1.3. *Notice that in the set M_{red} from Example 6.1.2, the constraint $y_2 + 2y_3 \geqslant 0$ is redundant and can be removed from the formulation. While this constraint does not impact the set $\widehat{M}_{red}$, its inner parallel set $(\widehat{M}_{red})^-_\delta$ would actually become larger by removing it. This again illustrates how granularity can benefit from the application of presolving techniques.*

6.2 Reduced Problems

Let us next extend Example 6.1.2 to general equality constrained mixed-integer linear optimization problems by examining the system $Cx + Dy = \gamma$, $y \in \mathbb{Z}^m$. If the latter has no solution then clearly M is empty and $MILP$ inconsistent (and thus also not granular). If, on the other hand, it is solvable, we may initially compute its solution space explicitly and incorporate it into the objective function $c^\intercal x + d^\intercal y$ and into the inequality constraints $Ax + By \leqslant b$ of $MILP$. This generates an equivalent optimization problem $MILP_{\text{red}}$ not only of smaller dimension, but in particular without equality constraints, which allows us to compute a potentially nonempty enlarged inner parallel set of the reduced problem.

The main challenge in the following will be to take care of the integrality conditions for the dependent integer variables in the explicit computation of the solution space of the equality constraints. In fact, to be able to take care of integrality conditions, dependent integer variables need to be separated from continuous variables, for which we propose the following concept.

Definition 6.2.1. *A $(q, n+m)$-matrix F is in partial reduced row echelon form (prref), if for some $t \leqslant \min\{q, n\}$ it has the form*

$$F = \begin{pmatrix} F_1 & F_2 \\ 0 & F_3 \end{pmatrix},$$

with a (t, n)-matrix F_1 of rank t in reduced row echelon form, a (t, m)-matrix F_2, and a $(q - t, m)$-matrix F_3.

In fact, applying Gauss-Jordan elimination to the matrix C yields a nonsingular (q, q)-matrix T which transforms the matrix (C, D) into prref with $t = \operatorname{rank}(C)$, that

is

$$\begin{pmatrix} \widetilde{C} & \widetilde{D}_1 \\ 0 & \widetilde{D}_2 \end{pmatrix} = T(C, D).$$

Note that the entries of $\widetilde{C}$, $\widetilde{D}_1$ and $\widetilde{D}_2$ are again rational and that the reduced row echelon form of C is $\begin{pmatrix} \widetilde{C} \\ 0 \end{pmatrix}$ with the (t, n)-matrix $\widetilde{C}$ and $t = \operatorname{rank}(C)$.

Finding all solutions to the system $Cx + By = \gamma$ is then equivalent to solving the two systems

$$\left(\widetilde{C}, \widetilde{D}_1\right) \begin{pmatrix} x \\ y \end{pmatrix} = \widetilde{\gamma}_1, \tag{6.3}$$

$$\widetilde{D}_2 y = \widetilde{\gamma}_2, \tag{6.4}$$

with $\widetilde{\gamma} = T\gamma$, and $\widetilde{\gamma}_1 = \widetilde{\gamma}_{[1:t]}$, $\widetilde{\gamma}_2 = \widetilde{\gamma}_{[t+1:q]}$. Here, for a vector v and an index set I, v_I denotes the subvector with the entries v_i, $i \in I$, of v. The notation $I = [\ell : u]$ with $\ell \leqslant u$ is shorthand for the index set $I = \{\ell, \dots, u\}$. Moreover, in the following for a matrix F and index sets I, J, $F_{I,J}$ will denote the submatrix of F with entries F_{ij}, $i \in I$, $j \in J$. The notation $F_{I,:}$ stands for the submatrix of F consisting of the rows with index $i \in I$, while $F_{:,J}$ denotes the submatrix of F consisting of the columns with index $j \in J$.

With respect to the reduced row echelon form of C, let $BV \subseteq \{1, \dots, n\}$ denote the index set of continuous *basic (dependent)* variables that correspond to the pivot elements of $\widetilde{C}$, and $NV \subseteq \{1, \dots, n\}$ the indices of continuous *nonbasic (independent)* variables. In the sequel, we will distinguish three cases:

1. All dependent variables are continuous ($t = q$).
2. The set of continuous dependent variables is empty ($t = 0$).
3. Some continuous and some integral variables are dependent ($0 < t < q$).

Let us initially discuss the first of the above cases, in which no integrality conditions on dependent variables appear. Although the corresponding elimination of equality constraints is well-known and straightforward, we give some details which will be analogous but not repeated in later proofs.

Lemma 6.2.2. *Let the $(q, n + m)$-matrix (C, D) have full row rank and let $T(C, D)$ be in partial reduced row echelon form (cf. Definition 6.2.1) with $t = q$, $F_1 := \widetilde{C}$, $F_2 := \widetilde{D}$, and an empty matrix F_3. Then, the problem MILP is equivalent to*

$$\begin{aligned} MILP_{\text{red}}: \quad & \min_{(x_{NV}, y) \in \mathbb{R}^{n-q} \times \mathbb{Z}^m} (c_{NV}^\intercal - c_{BV}^\intercal \widetilde{C}_{:,NV}) x_{NV} + (d^\intercal - c_{BV}^\intercal \widetilde{D}) y \\ s.t. \quad & (A_{:,NV} - A_{:,BV} \widetilde{C}_{:,NV}) x_{NV} + (B - A_{:,BV} \widetilde{D}) y \leqslant b - A_{:,BV} \widetilde{\gamma}. \end{aligned}$$

Proof. Due to $t = q$, $Cx + Dy = \gamma$ reduces to (6.3) with $\widetilde{D}_1 = \widetilde{D}$ and $\widetilde{\gamma}_1 = \widetilde{\gamma}$, which we may rearrange to

$$x_{BV} = \widetilde{C}_{:,BV}^{-1} (\widetilde{\gamma} - \widetilde{C}_{:,NV} x_{NV} - \widetilde{D} y) = \widetilde{\gamma} - \widetilde{C}_{:,NV} x_{NV} - \widetilde{D} y,$$

where the second equality stems from the fact that, in the reduced row echelon form, $\widetilde{C}_{:,BV}$ is the (q,q)-identity matrix. Moreover, we may rewrite the system of inequalities in the description of M as

$$\begin{aligned} b &\geqslant A_{:,BV}x_{BV} + A_{:,NV}x_{NV} + By \\ &= A_{:,BV}(\widetilde{\gamma} - \widetilde{C}_{:,NV}x_{NV} - \widetilde{D}y) + A_{:,NV}x_{NV} + By, \end{aligned}$$

and the objective function of $MILP$ as

$$c_{BV}^{\intercal}x_{BV} + c_{NV}^{\intercal}x_{NV} + d^{\intercal}y \;=\; c_{BV}^{\intercal}(\widetilde{\gamma} - \widetilde{C}_{:,NV}x_{NV} - \widetilde{D}y) + c_{NV}^{\intercal}x_{NV} + d^{\intercal}y$$

which overall yields the assertion.

Note that the optimal value of $MILP$ differs from the optimal value of $MILP_{\text{red}}$ by the constant $c_{BV}^{\intercal}\widetilde{\gamma}$.

Next, we investigate the second case ($t = 0$). As this only happens for $C = 0$ or $n = 0$, the original system $Cx + Dy = \gamma$ collapses to the system of linear Diophantine equations $Dy = \gamma$, which coincides with its partial reduced row echelon form. In particular, we will need to ensure integrality conditions on all dependent variables. To this end, we may use a series of *elementary (unimodular)* column operations to bring D into its Hermite normal form (HNF) [37]. A matrix is said to be in HNF if it has the form $(K, 0)$, with a nonsingular, lower triangular, nonnegative matrix K, in which each row has a unique maximum entry located on its main diagonal. Elementary column operations consist of

- exchanging two columns;
- multiplying a column by -1;
- adding an integral multiple of one column to another column.

Performing any of these operations is equivalent to post-multiplying D by a unimodular matrix (an integer matrix with a determinant of $+1$ or -1). The Hermite normal form theorem states that any rational matrix of full row rank can be brought into a unique HNF by a series of elementary column operations (cf. [37], or, e.g., [69] for a comprehensive introduction). In fact, one may find an unimodular (m,m)-matrix U such that

$$DU = (K, 0)$$

is the HNF of D. With the algorithm introduced in [43], this may even be done in polynomial time with respect to the length of the binary encoded input data.

With $U_1 := U_{:,[1:q]}$ it is straightforward to see that $\bar{y} := U_1K^{-1}\gamma$ is a particular solution of the system $Dy = \gamma$. Even better, $Dy = \gamma$ is solvable if and only if $K^{-1}\gamma$ is integral (cf, e.g., [69, Corollary 5.3 b.]). Therefore, if $K^{-1}\gamma$ is not integral, we immediately obtain $M = \varnothing$. Hence, in the following let $K^{-1}\gamma$ be integral. Then the set of all integral solutions of $Dy = \gamma$ may be determined as follows.

Lemma 6.2.3 ([69, Corollary 5.3 c.])**.** *Let $(K, 0)$ be the HNF of D with $DU = (K, 0)$, let $K^{-1}\gamma$ be integral, and define $U_1 := U_{:,[1:q]}$ as well as $U_2 := U_{:,[q+1:m]}$. Then the identity*

$$\{y \in \mathbb{Z}^m \,|\, Dy = \gamma\} \;=\; \left\{U_1K^{-1}\gamma + U_2\eta \,|\, \eta \in \mathbb{Z}^{m-q}\right\}$$

holds.

Substituting y in $MILP$ in accordance with Lemma 6.2.3 immediately yields the following result.

Proposition 6.2.4. *Let $C = 0$ or $n = 0$, let the (q, m)-matrix D have full row rank, let $(K, 0)$ be the HNF of D with $DU = (K, 0)$, let $K^{-1}\gamma$ be integral, and define $U_1 := U_{:,[1:q]}$ as well as $U_2 := U_{:,[q+1:m]}$. Then, the problem MILP is equivalent to*

$$\begin{aligned} MILP_{\text{red}}: \quad & \min_{(x,\eta)\in\mathbb{R}^n\times\mathbb{Z}^{m-q}} c^\intercal x + d^\intercal U_2\eta \\ & s.t. \quad Ax + BU_2\eta \leqslant b - BU_1K^{-1}\gamma. \end{aligned}$$

Note that the optimal value of $MILP$ differs from the optimal value of $MILP_{\text{red}}$ by the constant $d^\intercal U_1K^{-1}\gamma$. In case that D does not possess full row rank, again by Gaussian elimination redundant equations can be removed, so that Proposition 6.2.4 becomes applicable. An analogous remark applies to the subsequent Theorem 6.2.5.

Next we shall see how solving the third of the above cases ($0 < t < q$) may be accomplished by combining the first and second case (Lemma 6.2.2 and Proposition 6.2.4, resp.). Here, we initially determine the solution space to (6.4) and subsequently incorporate the latter in (6.3).

Theorem 6.2.5. *Let $T(C, D)$ be the partial reduced row echelon form of the $(q, n + m)$-matrix (C, D) with $0 < t = \operatorname{rank}(C) < q$, $F_1 := \widetilde{C}$, $F_2 := \widetilde{D}_1$, and $F_3 := \widetilde{D}_2$ (cf. Definition 6.2.1), where $\widetilde{D}_2$ has full row rank. Furthermore, let $(K, 0)$ be the HNF of $\widetilde{D}_2$ with $\widetilde{D}_2U = (K, 0)$, let $K^{-1}\widetilde{\gamma}_2$ be integral, and define $U_1 := U_{:,[1:q-t]}$ as well as $U_2 := U_{:,[q-t+1:m]}$. Then, the problem MILP is equivalent to*

$$\begin{aligned} MILP_{\text{red}}: \quad & \min_{(x_{NV},\eta)\in\mathbb{R}^{n-t}\times\mathbb{Z}^{m-q+t}} (c_{NV}^\intercal - c_{BV}^\intercal\widetilde{C}_{:,NV})x_{NV} + (d^\intercal - c_{BV}^\intercal\widetilde{D}_1)U_2\eta \\ & s.t. \quad (A_{:,NV} - A_{:,BV}\widetilde{C}_{:,NV})x_{NV} + (B - A_{:,BV}\widetilde{D}_1)U_2\eta \leqslant b - k, \end{aligned}$$

with $k = A_{:,BV}(\widetilde{\gamma}_1 - \widetilde{D}_1U_1K^{-1}\widetilde{\gamma}_2) + BU_1K^{-1}\widetilde{\gamma}_2$.

Proof. We may substitute y in accordance with Lemma 6.2.3 by initially determining the solution space for (6.4) as

$$y = U_1K^{-1}\widetilde{\gamma}_2 + U_2\eta, \tag{6.5}$$

with $\eta \in \mathbb{Z}^{m-q+t}$. Moreover, solving (6.3) for x_{BV} yields

$$\begin{aligned} x_{BV} &= \widetilde{\gamma}_1 - \widetilde{C}_{:,NV}x_{NV} - \widetilde{D}_1y \\ &= \widetilde{\gamma}_1 - \widetilde{C}_{:,NV}x_{NV} - \widetilde{D}_1(U_1K^{-1}\widetilde{\gamma}_2 + U_2\eta), \end{aligned} \tag{6.6}$$

with $x_{NV} \in \mathbb{R}^{n-t}$. The proof is completed along the lines of the one for Lemma 6.2.2. □

Note that the optimal value of $MILP$ differs from the optimal value of $MILP_{\text{red}}$ by the constant $c_{BV}^\intercal(\widetilde{\gamma}_1 - \widetilde{D}_1U_1K^{-1}\widetilde{\gamma}_2) + d^\intercal U_1K^{-1}\widetilde{\gamma}_2$. Moreover, recall that the violation of our assumption of integrality of $K^{-1}\widetilde{\gamma}_2$ in Theorem 6.2.5 (or $K^{-1}\gamma$ in Proposition 6.2.4) actually reveals inconsistency of $MILP$.

Remark 6.2.6. *As variables are usually box constrained, it is useful to explicitly write how these can be incorporated into the reduced model if not covered by the matrices A and B. Let $y^\ell \in (\mathbb{Z} \cup \{-\infty\})^m, y^u \in (\mathbb{Z} \cup \{\infty\})^m$ denote the vectors of lower and upper bounds on the integral variables and $x^\ell \in (\mathbb{R} \cup \{-\infty\})^n$, $x^u \in (\mathbb{R} \cup \{\infty\})^n$ correspondingly those of the continuous variables. In fact, with Lemma 6.2.3 it is straightforward to see that box*

constraints posed on integral variables, $y^\ell \leqslant y \leqslant y^u$, result in the constraints

$$y^\ell - U_1 K^{-1} \widetilde{\gamma}_2 \leqslant U_2 \eta \leqslant y^u - U_1 K^{-1} \widetilde{\gamma}_2$$

and analogously those on continuous basic variables yield the inequalities

$$\begin{aligned} \widetilde{C}_{:,NV} x_{NV} + \widetilde{D}_1 U_2 \eta &\leqslant -x^\ell_{BV} + \widetilde{\gamma}_1 - \widetilde{D}_1 U_1 K^{-1} \widetilde{\gamma}_2, \\ -\widetilde{C}_{:,NV} x_{NV} - \widetilde{D}_1 U_2 \eta &\leqslant x^u_{BV} - \widetilde{\gamma}_1 + \widetilde{D}_1 U_1 K^{-1} \widetilde{\gamma}_2, \end{aligned}$$

which potentially couple the variables x_{NV} and η. Also note that box constraints on nonbasic continuous variables x_{NV} can be incorporated without modification into the new model.

6.3 A Reduction Technique Tailored to Feasible Rounding Approaches

In the previous sections we have seen how equality constraints can be removed from the problem formulation while ensuring integrality conditions on integer variables and how this can, in principle, be useful for the application of feasible rounding approaches. Building on these results, we next develop a general reduction scheme specifically tuned to the application of feasible rounding approaches.

While equality constraints containing discrete variables *need* to be eliminated from the problem formulation to apply feasible rounding approaches (cf. Proposition 6.1.1), this is actually not the case for equality constraints only posed on continuous variables. Because LP-solvers can deal efficiently with equality constraints, it is beneficial to pass them to the solver instead of applying the effort to eliminate them.

This can certainly be achieved *after* the computation of the prref by selecting rows that contain only zero entries in $\widetilde{D}_1$. Yet, our aim is to also reduce the computational effort due to Gaussian elimination on a potentially large system $Cx + Dy = \gamma$ by excluding these variables *before* the computation of the prref. Note that only excluding rows from (C, D) where all entries of D are zero is often not feasible for this aim, because continuous variables that are coupled to integer variables via other constraints can still occur in such rows. The following running example of this section illustrates this.

Example 6.3.1. *Consider the case $n = 4$, $m = 2$ and the system $Cx + Dy = \gamma$, $y \in \mathbb{Z}^2$, with*

$$C = \begin{pmatrix} 1 & 3 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad \gamma = \begin{pmatrix} 4 \\ 4 \\ 2 \\ 3 \end{pmatrix}.$$

Here, only constraints 3 and 4 explicitly contain integer variables. Yet, even though $D_{2,:} = (0, 0)$ holds, constraint 2 implicitly contains integer variables because x_4 is coupled to y_1 via constraint 3. In contrast, no variables occurring in constraint 1 are coupled to integer variables, which enables a separate treatment for this constraint.

Thus, the challenge becomes to a-priori find continuous variables that are *not* coupled to integer variables. In the following, partly inspired by the discussion from [41], we show that this can indeed be achieved with a small computational effort.

For a matrix F, with $\operatorname{spm}(F)$ we denote its "sparsity pattern matrix", that is $(\operatorname{spm}(F))_{ij} = 0$ if $F_{ij} = 0$, and $(\operatorname{spm}(F))_{ij} = 1$, if $F_{ij} \neq 0$, and let $z = (x, y)^\intercal \in \mathbb{R}^{n+m}$.

The next lemma provides a direction how the matrix $H = \operatorname{spm}(C,D)^\intercal \operatorname{spm}(C,D)$ can be used to understand how the entries of z are linked via (C,D).

Lemma 6.3.2. *With $i,j \in \{1,\dots,m+n\}$, $i \neq j$, the variables z_i and z_j are directly coupled by some row $k \in \{1,\dots,q\}$ of (C,D) if and only if H_{ij} is nonzero.*

Proof. Writing

$$H_{ij} = \sum_{k=1}^{q} (\operatorname{spm}(C,D)^\intercal)_{ik} (\operatorname{spm}(C,D))_{kj} = \sum_{k=1}^{q} (\operatorname{spm}(C,D))_{ki} (\operatorname{spm}(C,D))_{kj}$$

reveals that H_{ij} counts the number of rows in which both variables i and j occur which proves the assertion.

□

Example 6.3.3. *For the data from Example 6.3.1 one easily computes*

$$H = \operatorname{spm}(C,D)^\intercal \operatorname{spm}(C,D) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Using Lemma 6.3.2, and with $\operatorname{diag}(H)$ denoting the diagonal matrix with diagonal entries from H, we can now view $\operatorname{spm}(H - \operatorname{diag}(H))$ as the adjacency matrix of an undirected graph Γ whose nodes represent the entries of z, and where we have edges between variables if they are directly coupled by some constraint.

This leads us to consider the *connected components* $\Gamma_1,\dots,\Gamma_\ell$ of Γ with $\ell \leqslant n+m$. Let the entry σ_j of the vector $\sigma \in \{1,\dots,\ell\}^{n+m}$ denote the index of the connected component which variable z_j is placed in. Thus the set $CC_I = \{\sigma_{n+1},\dots,\sigma_{n+m}\}$ collects all connected components that are related to integer variables, and a (continuous) variable x_j is only coupled to an integer variable by some constraint, if $\sigma_j \in CC_I$ holds.

Note that any row i of (C,D) connects all variables with nonzero entries, and that thus each constraint maps to exactly one connected component. We can therefore choose the first nonzero entry k of row i of (C,D) and set row i's component to $J_i = \sigma_k$.

Example 6.3.4. *For Example 6.3.1, and with H from Example 6.3.3, we find two connected components Γ_1 and Γ_2 of Γ with $\sigma = (1,1,2,2,2,2)^\intercal$ and $J_1 = 1$ as well as $J_2 = J_3 = J_4 = 2$. In view of $CC_I = \{\sigma_5,\sigma_6\} = \{2\}$ we obtain $J_1 \notin CC_I$, and $J_2, J_3, J_4 \in CC_I$.*

Example 6.3.4 indicates that the set $R := \{i |\ J_i \notin CC_I\}$ collects the indices of constraints that can be excluded from the elimination procedure. With $(C',D') := (C,D)_{R^c,:}$ denoting the submatrix of (C,D) with row indices outside of R, the next proposition shows this formally and further illustrates *how exactly* these constraints can be incorporated in the reduced model.

Proposition 6.3.5. *Let $\begin{pmatrix} \widetilde{C}' & \widetilde{D}'_1 \\ 0 & \widetilde{D}'_2 \end{pmatrix}$ be the prref of (C',D') and let BV', NV' denote the basic and nonbasic variables of $\widetilde{C}'$, respectively. Then the constraints from $Cx + Dy = \gamma$*

with indices in $R = \{i|\ J_i \notin CC_I\}$ can be incorporated into the reduced model separately by the system $C_{R,NV'}\, x_{NV'} = \gamma_R$.

Proof. Since CC_I collects connected components associated with *any* integer variable, let us initially note that $D_{R,:}$ is the zero matrix.

Furthermore, no variable can occur both in a row of (C, D) with index from R *and* in some row of (C', D') *with a nonzero coefficient*, because otherwise these rows would map to the same connected component of Γ. By definition all basic variables BV' must have some nonzero coefficient in some row from (C', D'), which implies that also $C_{R,BV'}$ is the zero matrix.

Thus, the constraints from $Cx + Dy = \gamma$ with indices in R can be written as

$$C_{R,NV'}\, x_{NV'} = \gamma_R,$$

which, because $x_{NV'}$ consists of variables from the reduced model, can be directly incorporated into the latter.

□

Connected components of a graph can be determined for example by breadth-first search in linear time with respect to the number of vertices and edges of the graph. This means that it is at most quadratic in $m + n$ which yields indeed a theoretical advantage over computing the reduced row echelon form, in particular for large values of q.

Overall, Proposition 6.3.5 leads to the reduced problem and to the feasible rounding approach in Algorithm 2. The main computational effort occurs in Steps 6, 7 and 11, at least with regard to worst-case complexity.

Clearly, the possibility of quickly computing a feasible point will not justify the application of the reduction technique for all sizes of (C, D). Therefore, one interesting question we shall come back to in our numerical study is to determine sizes of (C, D) that are reducible within a reasonable time after extracting constraints with indices in R from the system.

Example 6.3.6. *Let us illustrate the reduction scheme and the feasible rounding approach as presented in Algorithm 2 on the problem*

$$MILP: \quad \min_{x,y \in \mathbb{R}^4 \times \mathbb{Z}^2} e^\intercal x + e^\intercal y \quad s.t. \quad Cx + Dy = \gamma,\ 0 \leqslant x \leqslant 2e,\ 0 \leqslant y \leqslant 2e$$

with the all-ones vector e and C, D, γ from Example 6.3.1.

Steps 1 to 4 have already been performed in Example 6.3.3 and Example 6.3.4. Step 5 yields $R = \{1\}$ and, thus, $C_{R,:} = \begin{pmatrix} 1 & 3 & 0 & 0 \end{pmatrix}$ and $\gamma_R = 4$. Note that this already points to the fact that variables 1 and 2 need to be nonbasic variables. Indeed, computing the prref of $(C, D)_{R^c,:}$ in Step 6, we obtain

$$\widetilde{C} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},\ \widetilde{D}_1 = \begin{pmatrix} -0.5 & 0 \\ 1 & 0 \end{pmatrix},\ \widetilde{D}_2 = \begin{pmatrix} 1 & 1 \end{pmatrix},\ \widetilde{\gamma}_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix},\ \widetilde{\gamma}_2 = 3,$$

and computing the HNF of $\widetilde{D}_2$ yields $U = \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}$ and $K = 1$.

We can now use Remark 6.2.6 for the transformation of box constraints, where (after removing redundant constraints) the box constraints on y and on x_{BV} yield $\eta \in [1, 2]$.

Algorithm 2: Feasible Rounding Approach for equality constrained MILPs

Data: a problem $MILP$
Result: a feasible point $(\breve{x}^{ob}, \breve{y}^{ob})$, or "$MILP$ not granular", or "$MILP$ unbounded"

1 Compute $H = \operatorname{spm}(C,D)^\intercal \operatorname{spm}(C,D)$
2 Determine the connected components of the graph of $\operatorname{spm}(H - \operatorname{diag}(H))$, store them in $\sigma \in \mathbb{R}^{n+m}$
3 Set $CC_I = \{\sigma_{n+1}, \ldots, \sigma_{n+m}\}$
4 For $i = 1, \ldots, q$ compute $J_i = \sigma_k$, where k is chosen such that $(C,D)_{ik} \neq 0$ holds
5 Set $R = \{i |\ J_i \notin CC_I\}$
6 Compute the prref of $\begin{pmatrix} \widetilde{C} & \widetilde{D}_1 \\ 0 & \widetilde{D}_2 \end{pmatrix}$ of $(C,D)_{R^c,:}$ with basic variables BV and nonbasic variables NV
7 Determine $U = (U_1, U_2)$ and K by computing the HNF $\widetilde{D}_2 U = (K, 0)$
8 Set $\widetilde{A} = (A_{:,NV} - A_{:,BV}\widetilde{C}_{:,NV})$ and $\widetilde{B} = (B - A_{:,BV}\widetilde{D}_1)U_2$, and let $\widetilde{\alpha}_i^\intercal$ and $\widetilde{\beta}_i^\intercal$ denote the rows of $\widetilde{A}$ and $\widetilde{B}$, respectively
9 Compute the enlargement vector ω with

$$\omega_i = \begin{cases} \gcd(\widetilde{\beta}_i), & \text{if } \widetilde{\beta}_i \in \mathbb{Z}^{m-q+t} \text{ and } \widetilde{\alpha}_i = 0 \\ 0, & \text{otherwise} \end{cases}$$

10 Compute the shrinkage-vector $s = \frac{1}{2}(\left\|\widetilde{\beta}_1\right\|_1, \ldots, \left\|\widetilde{\beta}_p\right\|_1)^\intercal$
11 Try to compute an optimal point $(\widehat{x}_{NV}^{ob}, \widehat{\eta}^{ob})$ of the *objective based* LP

$$\begin{aligned} P_{\text{red}}^{ob}: \quad \min_{(x_{NV},\eta)\in\mathbb{R}^{n-t}\times\mathbb{R}^{m-q+t}} \ & (c_{NV}^\intercal - c_{BV}^\intercal \widetilde{C}_{:,NV})x_{NV} + (d^\intercal - c_{BV}^\intercal \widetilde{D}_1)U_2\eta \\ \text{s.t.} \ & \widetilde{\alpha}_i^\intercal x_{NV} + \widetilde{\beta}_i^\intercal \eta \leqslant \lfloor b_i - k_i \rfloor_{\omega_i} + \delta\omega_i - s_i, \ i = 1, \ldots, p, \\ & C_{R,NV}\, x_{NV} = \gamma_R, \end{aligned}$$

with $k = A_{:,BV}(\widetilde{\gamma}_1 - \widetilde{D}_1 U_1 K^{-1}\widetilde{\gamma}_2) + BU_1K^{-1}\widetilde{\gamma}_2$

12 **if** P_{red}^{ob} *is infeasible* **then**
13 return "$MILP$ not granular"
14 **else if** P_{red}^{ob} *is unbounded* **then**
15 return "$MILP$ is unbounded"
16 **else**
17 Compute a rounding $(\breve{x}_{NV}^{ob}, \breve{\eta}^{ob})$ of $(\widehat{x}_{NV}^{ob}, \widehat{\eta}^{ob})$ and the corresponding feasible point $(\breve{x}^{ob}, \breve{y}^{ob}) \in M$ using Equations (6.5) and (6.6)
18 return $(\breve{x}^{ob}, \breve{y}^{ob})$
19 **end**

Overall, we thus obtain the problem

$$P_{red}^{ob}: \quad \min_{(x,\eta)\in\mathbb{R}^2\times\mathbb{R}} \ x_1 + x_2 - \tfrac{1}{2}\eta \quad s.t. \quad x_1 + 3x_2 = 4,\ 0 \leqslant x \leqslant 2e,\ \tfrac{3}{2} - \delta \leqslant \eta \leqslant \tfrac{3}{2} + \delta$$

with optimal point $(\widehat{x}^{ob}, \widehat{\eta}^{ob}) = (0, \frac{4}{3}, \frac{3}{2} + \delta)^\intercal$ *and thus, for any* $\delta \in [0,1)$*, the feasible rounding* $(\breve{x}^{ob}, \breve{\eta}^{ob}) = (0, \frac{4}{3}, 2)^\intercal$ *for the reduced model. Using Equations* (6.5) *and* (6.6)*, we can transform this point back to the feasible point* $(\breve{x}^{ob}, \breve{y}^{ob}) = (0, \frac{4}{3}, 2, 0, 2, 1)^\intercal$ *of the*

original problem, which is the output of Algorithm 2. Coincidentally, this is also an optimal point of MILP.

Finally, as already suggested in Remark 5.3.2, we can potentially improve the obtained feasible point by applying a simple post processing step where we fix the integer variables to $\breve{y}^{ob}$, solve the continuous problem

$$P_{pp}(\breve{y}^{ob}): \quad \min_x c^\intercal x + d^\intercal y \quad \text{s.t.} \quad (x, \breve{y}^{ob}) \in M$$

and then update $(\breve{x}^{ob}, \breve{y}^{ob})$ to the optimal point of P_{pp}.

6.4 Computational Results

The purpose of our computational study is twofold. First, we wish to investigate if the reduction technique introduced in this chapter is computationally feasible for optimization problems from practice. Secondly, we wish to determine if the granularity concept can be used for computing good feasible points for relevant optimization problems which contain equality constraints.

We have implemented the reduction procedure and the feasible rounding approach outlined in Algorithm 2 in Matlab R2020a. For the computation of the connected components and the HNF we used standard Matlab functions. For determining the prref, we used the code from [4], which was (sometimes more than 5) orders of magnitudes faster than the Matlab's standard implementation. Moreover, we used Gurobi 8.1 to solve the appearing linear optimization problems and also as an evaluation for the cases where we obtained a feasible point. Our tests were run on a desktop computer with an Intel i7 processor with 4 cores à 4 GHz Turbo Boost and 16 GB RAM.

6.4.1 Practical Complexity of the Elimination Procedure

To obtain relevant test problems, we collected instances from the MIPLIB 2010 [46] and the MIPLIB 2017 [31]. We discarded duplicates and problems where no equality constraints on integer variables occur. Moreover, to avoid memory problems in computations, we restricted our study to problems smaller than 30 megabytes in the .mps format. This preselection yielded a test bed of 591 problems.

In Section 6.3, we already indicated that one limiting factor of the reduction scheme might be the computation of the prref and the HNF. In our study, we found that this was indeed the case. Due to the missing possibility to execute these Matlab functions with a preset time limit, we ran some initial experiments to obtain an impression of which problems are reducible within a reasonable time. We were looking for simple decision rules permitting to test the algorithm on a large data set without having to interfere manually.

In the computational study from Section 4.2, the optimization over the enlarged inner parallel set (of problems without equality constraints on integer variables) took less than one minute for all instances. We wanted the time for the computation of the HNF and the prref not to exceed this value by orders of magnitude, and decided that an estimated upper bound for these computations should not exceed 15 minutes.

For the computations on our machine we found the following two heuristic decision criteria useful:

(i) Compute the prref, if $q \leqslant 3 \cdot 10^4$ and $n \leqslant 3 \cdot 10^5$ holds.

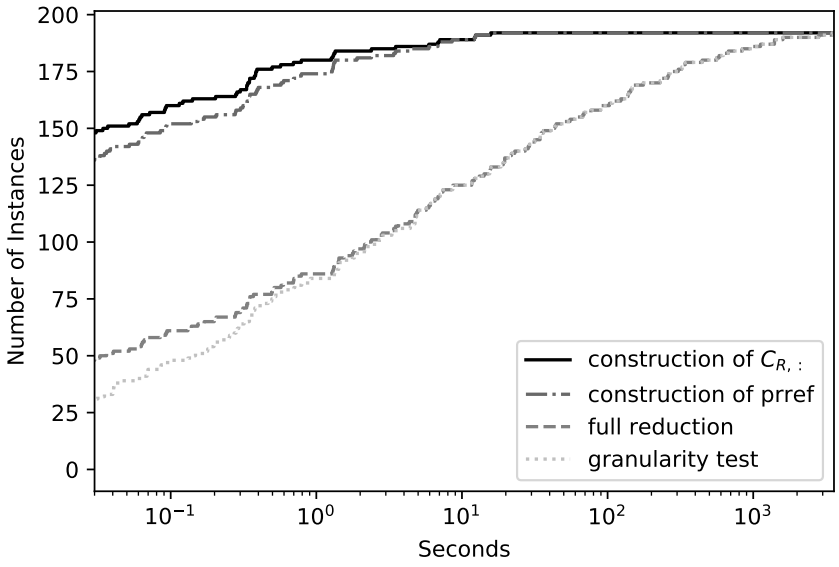


FIGURE 6.2: Cumulative share of problems for which the reduction scheme can be applied successfully over time

(ii) Compute the HNF, if the inequality $5.9(q-t)+0.79m \leqslant 1517$ is satisfied.

We remark that we left out important other factors, like the magnitude of the integer entries - which can generally play an important role in the computation of the HNF but was quite similar for all tested instances.

Using decision criterion (i), we were able to apply the prref for 583 problems. The computation of the prref needed at most one minute for all these instances, apart from the instance *uc-case11*, where the reduction of a $(19350, 29934)$-matrix did not finish after one hour. This yielded 582 remaining potential instances for reduction. Apart from 5 problems, the resulting matrix $\widetilde{D}_2$ was always integral for problems where decision criterion (ii) applied, which is the required input for Matlab's HNF function. In principle, scaling these (rational) matrices by the least common multiples of the denominators is possible but can result in (potentially significantly) larger values of $\widetilde{D}_2$ and thus in a longer run time in the computation of the HNF. We therefore excluded these 5 instances from our test bed.

From the remaining 577 problems, we were able to compute the HNF, and thus to fully reduce the model, for 192 instances using decision criterion (ii). Hence, a first main finding of our computational study is that the reduction technique is applicable to a relevant proportion of problems from practice.

Figure 6.2 summarizes the time needed for the reduction of these 192 instances by plotting the share of problems over time, for which different steps of Algorithm 2 could be performed. The individual curves include the construction of $C_{R,:}$ (Step 1 – Step 5), the computation of the prref (Step 6), that of the HNF (Step 7) and solving the problem P_δ^{ob} (Step 11). Detailed results for all 192 problems can be found in Table A.4 in the appendix.

For 83% of these instances, full reduction and a granularity test was possible within 100 seconds. Moreover, the construction of $C_{R,:}$ and of the prref was generally quite fast and only took a fraction of a second for most instances. Yet, the computation of the HNF took excessively long for several instances, even exceeding our (estimated) limit of 15 minutes in 11 cases.

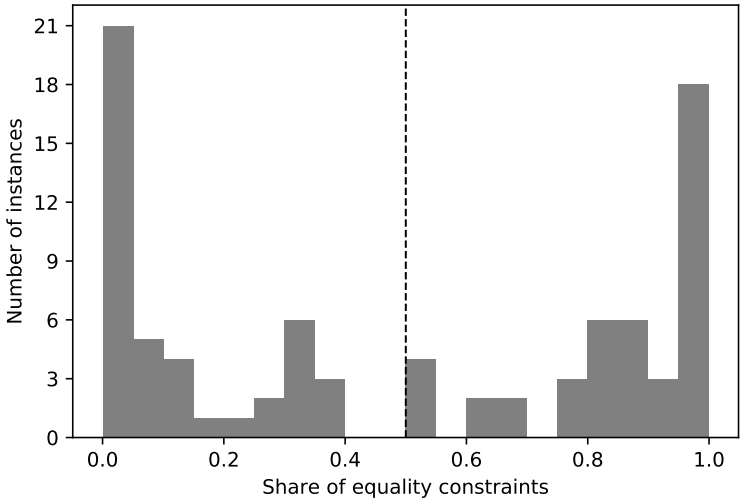


FIGURE 6.3: Share of extracted equality constraints for instances with separable continuous variables

Compared to all reduction steps combined, the optimization over the enlarged inner parallel set was very quick and took on average 0.08 and at most 4.89 seconds. This becomes visible in Figure 6.2 by the fact that the curves "full reduction" and "granularity test" are almost identical. This indicates that for the purpose of computing a feasible point by using the granularity concept, restricting oneself to smaller dimensions of systems of equations in the computation of the HNF might be more appropriate. We shall return to this observation in Section 6.4.2.

Due to its potential to reduce the time needed for the computation of the prref, separating equality constraints via the computation of connected components clearly can play a significant role in enhancing the chances of quick reducibility of a problem. Let us next shed some light on how this played out for the problems from our test bed.

We were able to extract a sometimes significant number of equality constraints for 129 (out of 591) instances. For these, Figure 6.3 shows a histogram of the share of equality constraints that could be incorporated separately, where the dotted line corresponds to the arithmetic mean. We remark that for 48 problems, this share was more than half, which indicates that this step was important for several problems. Moreover, Table 6.1 collects instances where more than 2000 constraints could be incorporated separately. Notice that these are partly problems with a very large number of variables and equality constraints and that for 3 of these instances, the computation of R lead to a quick reducibility.

6.4.2 Granularity in Equality Constrained Problems

Let us next investigate the possibility of applying the feasible rounding approach to equality constrained problems. From the 192 instances that remained from the reduction scheme, we obtained a nonempty enlarged inner parallel set for 30 instances. Hence, granularity is possible but generally less likely to occur for MILPs when equality constraints on integer variables are present. Indeed, while roughly half of the problems in the study in Section 4.2 were granular, here it is only one out of six.

	data					time		
	$\|R\|$	q	n	$q-t$	m	$C_{R,:}$	prref	HNF
mod011	4368	16	10879	0	96	0.06	0.00	0.00
neos-1061020	4227	6391	79	5349	13931	0.22	0.00	inf
neos-1423785	14964	960	19586	960	1920	0.01	0.00	inf
neos-3754224-navua	82485	2194	146654	2115	3574	0.09	0.04	inf
neos-4230265-orari	2370	5740	19590	2890	13390	0.04	4.04	inf
neos-4232544-orira	4740	11460	43080	5769	43980	0.07	29.14	inf
neos-4264598-oueme	4760	5740	41180	2880	13370	0.06	4.07	inf
neos-4292145-piako	2360	5740	19580	2880	13370	0.03	4.03	inf
neos-4300652-rahue	6006	96	12103	77	20900	0.05	0.00	inf
neos-5188808-nattai	5544	146	14256	146	288	0.00	0.00	29.97
neos-799711	22947	444	41088	444	910	1.09	0.00	inf
neos-799716	22943	444	29709	444	910	0.02	0.00	inf
ns1111636	4567	4400	347622	4400	13200	2.87	0.00	inf
ns2122698	40332	26932	138656	64	16447	0.26	0.03	inf
ns2124243	40332	26932	139636	64	16447	0.29	0.04	inf
shipsched	2001	6312	3045	6312	10549	0.00	0.00	inf
unitcal_7	2890	17	22899	0	2856	0.06	0.00	0.00

TABLE 6.1: Instances where the the computation of $C_{R,:}$ was particularly important. With q and $q-t$ we denote the number of rows of the matrices C and $\widetilde{D}_2$, respectively

Table 6.2 lists these granular problems along with the problem data and computing times for the important steps of the algorithm. To evaluate the quality of the generated feasible point, we furthermore list the run time Gurobi needs to compute a feasible point whose objective value is at least as good.

To compute this value, we passed the objective value of the feasible point obtained by our approach (including the post processing step) to Gurobi using the parameter "cutoff" and we set the parameter "heuristics" to the maximum value of one. The latter determines Gurobi's emphasis on finding good feasible points, where "larger values produce more and better feasible solutions, at a cost of slower progress in the best bound"[34].

	data				time				
instance	q	n	$q-t$	m	$C_{R,:}$	prref	HNF	P_{red}^{ob}	Gurobi
breastcancer-regularized.mps	4	9	4	706	0.13	0.00	1454.53	0.03	0.15
dc1l.mps	1638	1659	0	35638	3.56	0.01	0.00	1.39	26.08
ej.mps	1	0	1	3	0.00	0.00	0.04	0.00	0.00
enlight13.mps	169	0	169	338	0.00	0.00	491.58	0.01	0.01
enlight8.mps	64	0	64	128	0.01	0.00	32.71	0.00	0.00
enlight_hard.mps	100	0	100	200	0.00	0.00	102.61	0.00	0.00
go19.mps	80	0	80	441	0.00	0.00	21.06	0.04	0.49
markshare1.mps	6	12	0	50	0.01	0.01	0.00	0.00	0.00
markshare2.mps	7	14	0	60	0.00	0.00	0.00	0.00	0.01
markshare_4_0.mps	4	4	0	30	0.00	0.00	0.00	0.00	0.00
markshare_5_0.mps	5	5	0	40	0.00	0.01	0.00	0.00	0.00
neos-3116779-oban.mps	1	1	0	5140	0.00	0.02	0.00	0.00	0.00
neos-3118745-obra.mps	1	1	0	1130	0.00	0.00	0.00	0.01	0.03
neos-3352863-ancoa.mps	1	1	0	20045	0.00	0.00	0.00	3.05	3.10
neos-3610040-iskar.mps	1	345	0	85	0.00	0.00	0.00	0.00	0.03
neos-3610051-istra.mps	1	729	0	76	0.00	0.00	0.00	0.02	0.05
neos-3610173-itata.mps	1	767	0	77	0.00	0.00	0.00	0.01	0.03
neos-3611447-jijia.mps	1	387	0	85	0.00	0.00	0.00	0.00	0.02
neos-3611689-kaihu.mps	1	333	0	88	0.00	0.00	0.00	0.00	0.03
neos-935234.mps	139	2779	0	7530	0.01	0.00	0.00	0.27	29.96
neos-935627.mps	139	2779	0	7522	0.01	0.00	0.00	0.21	29.37
neos-935769.mps	139	2779	0	7020	0.01	0.00	0.00	0.22	12.15
neos-937511.mps	160	2770	0	8562	0.01	0.00	0.00	0.30	25.65
neos-937815.mps	160	2770	0	8876	0.02	0.00	0.00	0.34	28.80
neos-941262.mps	160	2770	0	6710	0.01	0.00	0.00	0.32	19.90

Continued on next page

	data				time				
instance	q	n	$q-t$	m	$C_{R,:}$	prref	HNF	P_{red}^{ob}	Gurobi
neos-948126.mps	156	2586	0	6965	0.01	0.00	0.00	0.29	33.54
neos-983171.mps	158	2408	0	6557	0.01	0.00	0.00	0.30	37.42
neos-984165.mps	155	2405	0	6478	0.01	0.00	0.00	0.32	46.45
pk1.mps	15	31	0	55	0.00	0.00	0.00	0.00	0.02
supportcase35.mps	2880	12365	0	576	0.01	0.17	0.00	0.53	0.04

TABLE 6.2: Comparison with Gurobi for granular problems

For 19 instances it took Gurobi more time to compute a point of similar quality than the reduction steps and solving the problem P^{ob}_{red} combined. This suggests that points computed by the feasible rounding approach could provide valuable information for equality constrained MILPs.

Moreover, Table 6.2 indicates that it is not likely that an application of the feasible rounding approach is helpful, if the run time of the reduction procedure exceeds 30 seconds. Interestingly, if one only takes problems into account where $q - t$ is at most one, the share of granular problems is 25 out of 99 and the chances of obtaining a granular problem after reduction are increased to roughly one out of four.

This also indicates that granularity is generally less likely to occur when discrete variables are eliminated using the HNF. From a practical point of view, one may hence restrict the application of the feasible rounding approaches to problems where $q - t$ or m is very small.

Notice that if we limit the application of the feasible rounding approach to such problems, the *neos*-instances in Table 6.2 are still included. The approach seems to be especially useful for these instances and also runs quite fast for all other remaining instances. Hence, we conclude that it may indeed be helpful for the computation of good feasible points when applied only to problems with a relatively small number of remaining equality constraints on integer variables (after the computation of the prref).

6.5 Conclusions and Further Investigations

In this chapter we have introduced an algorithmic framework that extends the applicability of the granularity concept to equality constrained MILPs. While naive approaches of dealing with equality constraints on integer variables prevent finding a nonempty inner parallel set, the tailored reduction scheme introduced in this chapter eliminates such constraints and thus indeed promotes the possibility for this set to be nonempty.

Our computational study indicates that the reduction scheme is possible for a relevant share of problems from standard libraries. Our study further shows that the granularity concept can be useful for the computation of good feasible points for equality constrained MILPs, if the number of equality constraints on integer variables is relatively small. When it is necessary to eliminate many integer variables from the model (i.e. when the matrix $\widetilde{D}_2$ has more than a few rows and m is large), the chances of granularity are significantly reduced and using other techniques for the computation of feasible points is likely to be more appropriate.

One reason for the decreasing chances of granularity under the occurrence of equality constraints might be the sparsity of the reduced problem compared to the original formulation. While MILPs often reveal a sparse pattern in the coefficient matrix of the inequality constraints, this is unlikely to occur after the reduction step. Because a relatively dense coefficient matrix for integer variables implies large values of the shrinkage-vector (cf. Line 10 of Algorithm 2), it is precisely this attribute that may prevent granularity for these problems.

This issue may be circumvented by developing alternative formulations for these problems, either before, or after the reduction step. An investigation of the possibility of (automatically) applying remodeling ideas on equality constrained MILPs in granularity promoting ways is thus an interesting field of investigation for future research. Finally, we leave the evaluation of applying the present approach to convex MINLPs with equality constraints to future research.

Chapter 7

Inner Parallel Sets in Search Trees

In the previous chapters, feasible rounding ideas were introduced and successfully tested as standalone concepts for mixed-integer linear and nonlinear optimization problems. Yet, so far it is untested how these approaches might function when integrated in branch-and-bound methods. In particular, it has not been studied how inner parallel sets behave when we move down a search tree.

In this chapter we intend to close this gap. Additionally, based on these results, we develop a novel method that combines feasible rounding approaches and diving ideas. This chapter is structured as follows.

In Section 7.1 we recall the general setting and briefly introduce some additional concepts. We then provide a theoretical analysis of the behavior of inner parallel sets when variables are fixed in Section 7.2. 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 new diving ideas which can improve standalone feasible rounding approaches. This is the content of Section 7.3. To arrive at a specific algorithm, we formulate a method for MILPs. Finally, in Section 7.4, we conduct a computational study on a standard test library of MILPs [31] that sheds a light on the effectiveness of these diving strategies and also on the potential benefit of integrating feasible rounding approaches into a solver framework. Section 7.5 concludes the chapter and offers directions for further research.

7.1 Preliminaries

In this section, we initially rehash some constructions from the previous chapters and fit them into the present setting.

In this chapter, we study general mixed-integer nonlinear optimization problems of the form

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

that is, we use the general setting from Section 2.1 but without loss of generality assume a linear objective function. We will also be interested in the special case of MILPs ($I = \varnothing$) for which we will develop a novel diving heuristic in Section 7.3.

Recall that we call $(\breve{x}, \breve{y})$ rounding of a point $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$, if

$$\breve{x} = x,\ \breve{y} \in \mathbb{Z}^m,\ |\breve{y}_j - y_j| \leqslant \tfrac{1}{2},\ j = 1, \dots m \tag{7.1}$$

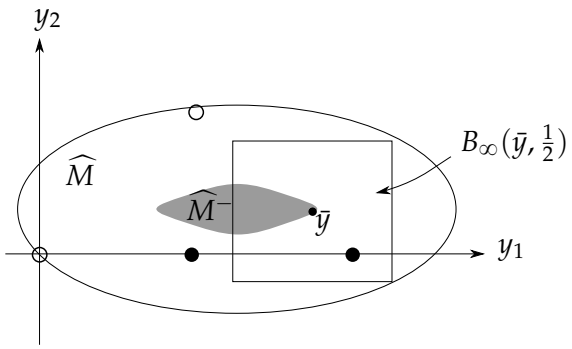


FIGURE 7.1: Construction of the inner parallel set $\widehat{M}^-$. The filled points are obtainable as roundings from $\widehat{M}^-$ and thus form the set R.

hold. Then, for a set $S \subseteq \mathbb{R}^n \times \mathbb{R}^m$ let us define the set of roundings obtainable from S as

$$R(S) := \{(\check{x}, \check{y}) \in \mathbb{R}^n \times \mathbb{Z}^m |\ (x, y) \in S \text{ and } (7.1)\}$$

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

Figure 7.1 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 $\widehat{M}^-$, that is, the relation $R = \{(0,1)^\intercal, (0,2)^\intercal\}$ holds.

Recall from Chapter 3 that in the linear case, with an enlargement parameter $\delta \in [0,1)$, the vector ω of greatest common divisors, $\widetilde{b} := \lfloor b \rfloor_\omega + \delta\omega$ and the all-ones vector e of dimension m, an explicitly computable enlarged relaxed feasible set is

$$\widetilde{M} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m |\ Ax + By \leqslant \widetilde{b},\ y^\ell - \delta e \leqslant y \leqslant y^u + \delta e\}. \tag{7.2}$$

For ease of notation, in this chapter we omit the δ-dependency of $\widetilde{M}$. Correspondingly, the enlarged inner parallel set can be written as

$$\widetilde{M}^- = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m |\ Ax + By \leqslant \widetilde{b} - \tfrac{1}{2}\|\beta\|_1,\ y^\ell + (\tfrac{1}{2} - \delta)e \leqslant y \leqslant y^u - (\tfrac{1}{2} - \delta)e\}. \tag{7.3}$$

We next illustrate the computation of the enlarged inner parallel set for a binary knapsack example which we shall also revisit in Section 7.2 to demonstrate the usefulness of fixing binary variables. Here, we use the abbreviation $\widetilde{R} := R(\widetilde{M}^-)$.

Example 7.1.1. *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 (7.3) *with* $\omega = (1,1)^\intercal$ *and* $\|\beta\|_1 = (3,3)^\intercal$ *the enlarged inner parallel set is*

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

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

7.2 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 binary variables 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 we noted in the computational study in Chapter 4 and is further substantiated by the theoretical bounds derived in [62]. Therefore, the case of fixing binary variables is of particular interest for our present analysis.

In the following, we make no further distinction between different nodes of a branch-and-bound tree, and 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 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

$$\widehat{M}_{(i)}(\ell) = \{(x, \widetilde{y}) \in \mathbb{R}^n \times \mathbb{R}^{m-1} |\ (x, (\widetilde{y}_1, \ldots, \widetilde{y}_{i-1}, \ell, \widetilde{y}_i, \ldots, \widetilde{y}_{m-1})) \in \widehat{M}\}. \tag{7.4}$$

Moreover, with

$$\widehat{M}_{(i)}(\ell)^- = \{(x, \widetilde{y}) \in \mathbb{R}^n \times \mathbb{R}^{m-1} |\ \{x\} \times B_\infty(\widetilde{y}, \tfrac{1}{2}) \subseteq \widehat{M}_{(i)}(\ell)\} \tag{7.5}$$

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 7.2.1. *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

$$y^{-i} := (y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_m)^\mathsf{T} \in \mathbb{R}^{m-1}, \tag{7.6}$$

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

$$y^{+i}(\ell) := (y_1, \ldots, y_{i-1}, \ell, y_i, \ldots, y_{m-1})^\mathsf{T} \in \mathbb{R}^m \tag{7.7}$$

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

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

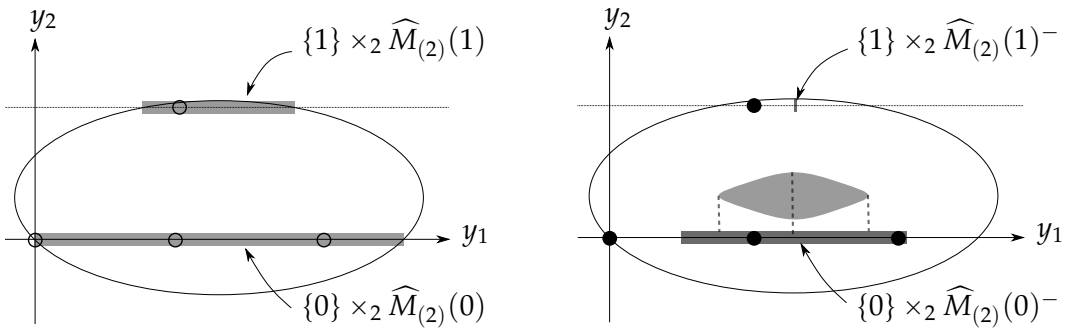


FIGURE 7.2: Construction of the i-ℓ-relaxed feasible set (left) and the i-ℓ-fixed inner parallel sets (right) with $i = 2$ and $\ell \in \{0, 1\}$.

The construction of i-ℓ-fixed (inner parallel) sets is illustrated in Figure 7.2 for $i = 2$ and $\ell \in \{0, 1\}$. Remarkably, fixing y_2 results in

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

and thus allows us to obtain all points in M as roundings from i-ℓ-fixed inner parallel sets. Recall from Figure 7.1 that we were only able to obtain the two points $(0,1)^\intercal, (0,2)^\intercal$ 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 Figure 7.2: 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 7.2.2. *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. □

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

Theorem 7.2.3. *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 7.2.2 implies $(x, y^{-i}) \in \widehat{M}_{(i)}(\breve{y}_i)^-$. Note that, although the rounding $(\breve{x}, (\widecheck{y^{-i}}))$ of (x, y^{-i}) is in general not unique, it can be chosen such that $(\widecheck{y^{-i}}) = (\breve{y})^{-i}$ 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 7.2.3 together with our considerations from Figures 7.1 and 7.2 immediately yields the following corollary.

Corollary 7.2.4. *The set of feasible points obtainable by the feasible rounding approaches is* nondecreasing *and* potentially increases *with increasing depth of the search tree.*

Let us next revisit Example 7.1.1 to illustrate the explicit construction of i-ℓ-fixed enlarged inner parallel sets for MILPs.

Example 7.2.5. *Let us consider the feasible set M from Example 7.1.1 and fix y_3. Again, with $\omega = (1,1)^\intercal$, this results in the two 3-ℓ-fixed enlarged sets*

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

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

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

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 7.2.5 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 7.2.1 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 $\beta_i^\intercal y \leqslant b_i$ with $\beta_i = (1,3,3)^\intercal$ and $b_i = 3$. Then, when fixing y_1 and using the enlargement techniques from (7.2), 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.

7.3 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 develop a method for mixed-integer linear optimization problems $MILP$ and use the notation introduced in Section 7.1 for $MINLP$ with $I = \emptyset$. In particular, we employ the construction of the enlarged inner parallel set from (7.3).

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 will be 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.

7.3.1 A Diving Step for a Nonempty Enlarged Inner Parallel Set

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

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

Due to our assumption $\widetilde{M}^- \neq \emptyset$, 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 $\widetilde{M}^- \neq \emptyset$, 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 $\breve{v}^{ob} = c^\intercal \breve{x}^{ob} + d^\intercal \breve{y}^{ob}$.

One crucial observation from Lemma 7.2.2 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 $i \in \{1, \ldots, m\}$ and $\ell = \breve{y}_i$.

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

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

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 (7.3).

We abbreviate

$$(x^{ob}, \widetilde{y}^{ob})_{(i)}(\ell) := (x^{ob}, (\widetilde{y}^{ob})^{+i}(\ell)) \tag{7.8}$$

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 $(\breve{x}^{ob}, \breve{y}^{ob})_{(i)}(\ell)$. Moreover, 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 $\ell = \breve{y}_i^{ob}$. The next example elaborates this idea more fully and shows that, even though the fixing value for variable i is given by $\breve{y}_i^{ob}$, different orders of selecting variables can yield different feasible points.

Example 7.3.1. *Consider the optimization problem*

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

By using Equation (7.3) *with* $\delta = 0.9$*, we can formulate the objective based problem*

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

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

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

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

with optimal point $\widetilde{y}^{ob} = (1.12, 0.64)^\intercal$ *and thus the IP-feasible point* $\breve{y}^{ob}_{(2)}(0) = (1, 0, 1)^\intercal$ *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* $y_3 = 0$*. Note that both fixings rule out the possibility to obtain the feasible point* $(1, 0, 1)^\intercal$ *on a path in the search tree and that this point is hence only obtainable if we initially fix* $y_2 = 0$*.*

Example 7.3.1 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 7.3.3.

Remark 7.3.2. *The main reason for our choice of fixing variable* i *to* $\breve{y}_i^{ob}$ *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 7.2.2 we can fix any* i *of these* k *variables to values* $\ell \in \{\breve{y}_i^1, \breve{y}_i^2\}$*. In our linear setting, the inner parallel set is convex and hence even all values from the interval* $[\min\{\breve{y}_i^1, \breve{y}_i^2\}, \max\{\breve{y}_i^1, \breve{y}_i^2\}]$ *are possible.*

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

7.3.2 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 $\widehat{M}$,

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

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 Chapter 4, 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 that 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 7.3.3. *Let (x^f,y^f,z^f) be an optimal point of P^f. Then for any $i\in\{1,\dots,m\}$ 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 7.2.2. It implies $v^f_{(i)}(\breve{y}_i)\leqslant z^f=v^f$ and thus part b of the assertion. □

Proposition 7.3.3b 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 7.3.3. 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),

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

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

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

this constraint can be written as

$$\alpha_j^\intercal x^f + (\beta_j^{-i})^\intercal (y^f)^{-i} - z^f \leqslant \widetilde{b}_j - \tfrac{1}{2}(\left\|\beta_j^{-i}\right\|_1 + |B_{ji}|) - B_{ji} y_i^f. \tag{7.9}$$

Proposition 7.3.3a implies that when we evaluate the corresponding constraint j of $(\widetilde{M}_L)_{(i)}(\breve{y}_i^f)^-$ at $(x^f, (y^f)^{-i}, z^f)$, we obtain the valid inequality

$$\alpha_j^\intercal x^f + (\beta_j^{-i})^\intercal (y^f)^{-i} - z^f \leqslant \widetilde{b}_j - \tfrac{1}{2}\left\|\beta_j^{-i}\right\|_1 - B_{ji}\breve{y}_i^f. \tag{7.10}$$

Moreover, because the left-hand sides of inequalities (7.9) and (7.10) coincide, we can now compare their right-hand sides to see if constraint j is relaxed in the transition from P^f to $P^f_{(i)}(\ell)$. Subtracting the right-hand side of (7.9) from the right-hand side of (7.10) yields the *degree of freedom*

$$f_{ji} = \tfrac{1}{2}|B_{ji}| + B_{ji}(y_i^f - \breve{y}_i^f) \tag{7.11}$$

that becomes available in constraint j of the problem $P^f_{(i)}(\breve{y}_i^f)$ due to fixing variable i. Note that $f_{ji} \in [0, |B_{ji}|]$ not only confirms Proposition 7.3.3a, 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}. Hence, if a variable appears in a constraint, usually also a strictly positive degree of freedom is possible, in particular because $|y_i^f - \breve{y}_i^f| = \frac{1}{2}$ implies ambiguity of the rounding $\breve{y}_i^f$, so that one might be able to choose $\breve{y}_i^f$ such that $f_{ji} = |B_{ji}|$ holds.

In the following, let $J_A \subseteq \{1, \ldots, p\}$ 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 7.3.4. *With an optimal point (x^f, y^f, z^f) of P^f and $v^f > 0$, for some $i \in \{1, \ldots, m\}$ let $f_i > 0$. Then we have $v_{(i)}(\breve{y}_i^f) < v^f$.*

Proof. By Proposition 7.3.3a, the point $(x^f, (y^f)^{-i}, z^f)$ is feasible for $P^f_{(i)}(\breve{y}_i)$. As its objective value coincides with v^f, it suffices to show that it is not optimal for $P^f_{(i)}(\breve{y}_i)$.

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 (7.10) holds with equality for some $j \in \{1, \ldots, p\}$.

For $j \in J_A$ this is ruled out by our assumption $f_{ji} > 0$. Moreover, for $j \in \{1, \ldots, p\} \backslash J_A$ inequality (7.9) is strictly satisfied. This, with $f_{ji} \geqslant 0$, implies that also

inequality (7.10) 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 7.3.5. *Consider a linear optimization problem with 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

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

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 $(y^f, z^f) = (\frac{1}{2} - \delta, \frac{1}{2} + \delta, \frac{1}{4}, \frac{3}{2} - \delta)^\intercal$ 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 $\breve{y}^f = (0, 1, 0)^\intercal$. 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 $(\frac{1}{2}, \frac{1}{2} + \delta)^\intercal$.

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

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

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 7.3.4 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 $\bar{I} = \{i_1, \dots, i_k\} \subseteq \{1, \dots, m\}$ and a set of corresponding integer values $\bar{L} = \{\ell_{i_1}, \dots, \ell_{i_k}\}$, in the following let the $\bar{I}$-$\bar{L}$-fixed enlarged inner parallel set $\widetilde{M}_{(I)}(\bar{L})^-$ be defined analogously to Equations (7.4) and (7.5) 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 $\bar{L} = \{\breve{y}^f_{i_1}, \dots, \breve{y}^f_{i_k}\}$. Then, a repeated application of Lemma 7.2.2 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 (7.9) - (7.11), 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 freedoms of these variables, that is,

$$f_{j\bar{I}} = \sum_{i \in \bar{I}} f_{ji} = \sum_{i \in \bar{I}} \left(\tfrac{1}{2}|B_{ji}| + B_{ji}(y^f_i - \breve{y}^f_i) \right). \tag{7.12}$$

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 7.3.4 to the following proposition.

Proposition 7.3.6. *With an optimal point (x^f, y^f, z^f) of P^f, for some $\bar{I} \subseteq \{1, \dots, m\}$ 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 $J_U = \bigcup_{i \in \{1,\dots,m\}} J_i$ there exists some index set $\bar{I}$ with $f_{\bar{I}} > 0$, if and only if $J_U = J_A$ holds. Therefore, by Proposition 7.3.6, $J_U = J_A$ 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., [56]), where J_U is the universe and $\{J_i |\ i \in \{1, \dots, m\}\}$ 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 [20, 42] 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.

$$i_k = \operatorname*{arg\,max}_{i \in \{1,\dots,m\}} |\{j \in J_i |\ j \in J_U \backslash (\bigcup_{\tilde{i} \in \bar{I}^k} J_{\tilde{i}})\}|. \tag{7.13}$$

Algorithm 3: feasibility-IPS-diving

Data: a 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 $(\breve{x}^{fd}, \breve{y}^{fd})$

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

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

  with merged rounding $(\breve{x}^f, \breve{y}^f)_{(\bar{I}^k)}(\bar{L}^k)$ and non-gran. measure $v^f_{(\bar{I}^k)}(\bar{L}^k)$
  set $v^{fd} \leftarrow v^f_{(\bar{I}^k)}(\bar{L}^k)$
  if $(\breve{x}^f, \breve{y}^f)_{(\bar{I}^k)}(\bar{L}^k) \in M$ **then**
    $(\breve{x}^{fd}, \breve{y}^{fd}) \leftarrow (\breve{x}^f, \breve{y}^f)_{(\bar{I}^k)}(\bar{L}^k)$
  end
  choose a set of indices $I^k \subseteq \{1, \dots, m\} \backslash \bar{I}^k$
  set $\bar{I}^{k+1} \leftarrow \bar{I}^k \cup I^k$, $\bar{L}^{k+1} \leftarrow \bar{L}^k \cup \{\breve{y}^k_{i_k} \mid i_k \in I^k\}$, $k \leftarrow k+1$
end
set $\bar{I} \leftarrow \bar{I}^{k-1}$, $\bar{L} \leftarrow \bar{L}^{k-1}$

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.

7.3.3 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.

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 7.3.2, until we possibly obtain a nonempty enlarged inner parallel set of some child node. The detailed procedure, feasibility-**I**nner**P**arallel**S**et-diving, is outlined in Algorithm 3 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 $v^f_{(\bar{I}^k)}(\bar{L}^k) > 0$

Algorithm 4: objective-IPS-diving

Data: a bounded 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* $\bar{I}^k \subsetneq \{1, \dots, m\}$ **do**

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

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

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 $I^k \subseteq \{1, \dots, m\} \backslash \bar{I}^k$

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**

holds (cf. Example 7.3.5). 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 4, 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 4 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 $MILP$.

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 7.2.2 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 $\breve{v}^{ob}_{(\bar{I}^k)}(\bar{L}^k) = c^\intercal \breve{x}^k + d^\intercal \breve{y}^k + \sum_{i\in\bar{I}^k} d_i\ell_i$ 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.

When integrating such diving approaches into a solver framework it makes sense

to apply a probing step after fixing variables $\bar{I}^k$ to $\bar{L}^k$ which potentially allows to fix additional variables and thus to speed up the diving process.

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 7.3.6, 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 (7.9) and (7.10) 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 (7.11). 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 $\breve{v}^{ob}_{(\bar{I})}(\bar{L})$. 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 however more intricate and not easy to predict. Still, choosing indices in accordance with equation (7.13) offers new flexibility in the constraints and is thus likely to enable the possibility of obtaining a different rounding, which might be beneficial for obtaining new (and hopefully improved) values $v^f_{(\bar{I})}(\bar{L})$.

7.4 Computational Study

The main intention of our computational study is to see if feasible rounding approaches can benefit from applying diving steps as outlined in Algorithms 3 and 4. In particular, we wish to determine if the granularity concept can be extended to encompass more problems by using feasibility-IPS-diving, and if the objective values of the feasible points generated in the root node can be improved by applying objective-IPS-diving steps.

Corollary 7.2.4 states that the number of roundings is non-decreasing and *potentially increasing* with increasing depth of the search tree. Our analysis will thus additionally offer an intuition of whether we can actually expect to obtain an *increasing* number of feasible roundings via inner parallel sets in branch-and-bound trees for problems from practice.

A second intention of our study is to examine the influence of different choices of fixing variables and in particular to evaluate the introduced greedy strategy for feasibility-IPS-diving. Finally, we address the important question whether the generated points can add value to the arsenal of primal heuristics within the solver framework SCIP.

The test bed of our computational study stems from the collection set of the MIPLIB 2017 [31]. We collected problems in standard form of size less than 30 megabyte which contain no equality constraints on integer variables, as the treatment of such constraints when using feasible rounding approaches needs special attention (cf. Chapter 6). Additionally discarding all infeasible problems results in a test bed containing 244 instances.

We have implemented the feasible rounding approaches with diving strategies outlined in Algorithms 3 and 4 in Matlab R2020a and in Pyscipopt [52]. All tests are run on a 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.

7.4.1 Selection of Indices

Recall that the flexibility of our diving method introduced in Section 7.3.3 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 (7.13). To avoid collecting variable constraint pairs (i, j) with trivial degrees of freedom, we set $J_i := \{j \in J_A |\ f_{ji} > 10^{-4}|B_{ji}|\}$.

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. Secondly, 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

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

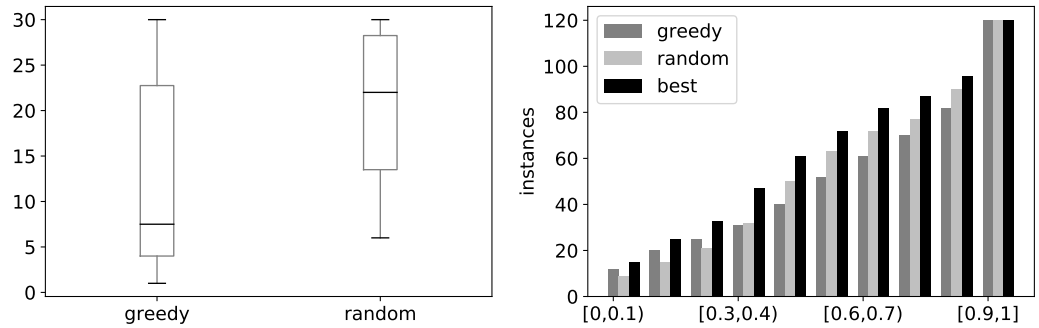
Note that, while $f_{j\bar{I}} \in \mathbb{R}$ denotes the impact on constraint j of fixing all variables from $\bar{I}$, the value $f_{J_A i} \in \mathbb{R}$ stands for the added up impact on all active constraints of fixing variable i.

7.4.2 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. Secondly, we examine if applying objective-IPS-diving steps yields improved feasible points and assess the significance of this (potential) improvement.

Out of the 244 instances from our test bed, we find that 121 are granular in the root node. Using feasibility-IPS-diving with both diving strategies, we are able to find granular nodes for 148 problems so that the share of problems for which we may compute granularity based feasible points increases from 49.6% (root node only) to 60.7% (using feasibility-IPS-diving). Thus our first finding is that using diving steps significantly increases the applicability of the granularity concept. We report detailed results for these 148 problems in Table A.5 in the appendix.

As a comparison of the two methods for selecting fixing indices (random and greedy), we can state that for 20 non-granular instances both diving methods are able to find granular nodes. The random strategy finds a granular node in 4 additional cases, and in 3 cases only the greedy strategy yields a granular node.



(a) Number of diving steps to a granular node

(b) Remaining optimality gap (7.14)

FIGURE 7.3: A comparison of diving methods among each other and with the root 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 indices. Yet, as the boxplot of the number of iterations of both methods shown in Figure 7.3a reveals, the greedy method usually finds granular nodes much earlier in the search tree. Indeed, for the 20 instances where both methods yield a granular node, the median number of iterations is 7.5 for the greedy and 22 for the random method and also the 25th and 75th percentiles differ significantly. Additionally, a direct comparison of the number of iterations for each problem individually shows that the greedy method needs (often significantly) less iterations in 13 cases and more iterations in only 2 cases. Hence, if we are interested in quickly finding granular nodes, the greedy strategy seems to be the appropriate choice.

For the 121 (root-node) granular problems, we can compute and evaluate the improvement yielded by objective-IPS-diving. In this regard, with v denoting the optimal (or best known) value obtained from the MIPLIB website [55], for each problem with $\breve{v}^{ob} \neq v$ we compute the value

$$gap_{closed} = (\breve{v}^{ob} - \breve{v}^{obd}) / (\breve{v}^{ob} - v) \tag{7.14}$$

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 one 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 120 problems.

Figure 7.3b 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 bounds closed by both strategies individually, as well as a third option *best*, which is the bound 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 61 of the 120 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 50 and 40 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 points to the fact that both methods might be complementary to each other and at least demonstrates that it is beneficial to apply different fixing strategies from the root node.

Overall we may 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 indices.

7.4.3 Possibilities of Integrating Feasible Rounding Approaches and Diving Ideas Into a Solver Framework

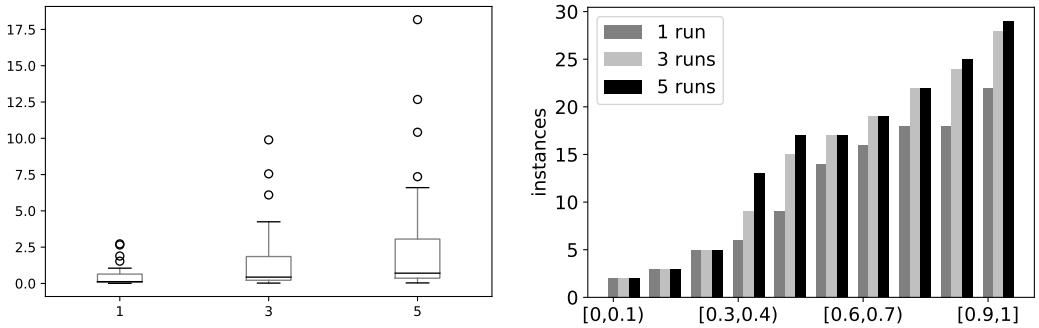
In a second experiment, we study the potential benefit of integrating feasible rounding approaches with and without diving steps into a solver framework. We use SCIP for this purpose and initially evaluate the quality of the generated feasible points compared to the best solution SCIP obtains with its various heuristics after solving the root node. To this end, we test the method in the Pyscipopt framework, executing it after the processing of a node is finished.

We focus our analysis on instances where the root node is granular and apply up to 5 runs of diving using the random strategy with different seeds. As described in Section 7.4.1 in each run we need to solve at most 30 linear optimization problems (of decreasing size). One advantage of SCIP is that after fixing variables, probing steps can be applied so that the number of diving steps needed can often be significantly reduced.

After SCIP's preprocessing steps, we obtain a test set of 107 granular problems. In 13 cases, we report that the feasible rounding obtained by solving the objective-based problem in the root node is able to improve upon those previously found by SCIP. By applying 1, 3 and 5 random diving runs, this number is increased to 22, 28 and 29, respectively. Detailed results for the 29 instances where 5 diving runs yield best incumbent solutions can be found in Table A.6 in the appendix. The significant increase in best incumbent solutions again highlights the potential of applying diving steps when using feasible roundings approaches. Moreover, the number of best incumbent solutions increases significantly when 3 diving runs are applied (compared to 1). With 5 random diving runs, we only obtain 1 additional best incumbent solution (compared to 3 runs) which suggests that 3 runs might be a good compromise between effort and benefit of the method.

This is further substantiated in Figure 7.4. In Figure 7.4a we compare the cumulative time of solving all appearing LPs t_{diving} with the time SCIP used previous to applying the diving methods t_{scip}. Here, we show a boxplot of the ratio t_{diving}/t_{scip}. For 3 diving runs, this ratio is between 0.022 and 9.89, with median 0.43 and 75th percentile of 1.85 which seems to be within a reasonable range.

Figure 7.4b gives an impression of the relative improvement of the best incumbent solutions in a cumulative histogram. Once more we display the remaining optimality gap $1 - gap_{closed}$, where the $\check{v}^{ob}$ in Equation (7.14) is replaced with the best incumbent solution of SCIP. For 15 problems, more than half of the optimality gap is closed by the points obtained within 3 rounds of diving. For one diving round, this



(a) Box plot of the time ratio t_{diving}/t_{scip}

(b) Remaining optimality gap

FIGURE 7.4: Computational effort and quality of the generated feasible point compared to SCIPs incumbent solution in the root node

is the case only for 9 problems which again shows that running objective-IPS-diving more than one time can be beneficial.

To give a fuller 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 28 problems where 3 rounds of diving yield best incumbent solutions.

We report that in 25 cases, SCIP needs additional time to compute a feasible point of similar quality. To give an impression of the potential benefit of the method, we list the 12 instances in Table 7.1 where SCIP needs more than 30 seconds additional time to compute a feasible point of similar quality. Here we report the objective value obtained in the root node and after 3 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.

	objective			time	
name	root	diving	optimal	diving	SCIP
b1c1s1	72555.0	69071.5	24544.2	0.8	35.2
b2c1s1	73676.5	68701.5	25687.9	1.5	45.5
dg012142	25623489.0	14373382.6	2300867.0	6.8	41.0
gsvm2rl11	42635.3	39792.8	18121.6	15.9	64.8
gsvm2rl12	34.4	34.4	22.1	29.4	>1800.0
gsvm2rl9	16382.8	13611.9	7438.2	3.9	368.0
mushroom-best	3613.9	2072.9	0.1	11.4	80.0
neos-983171	50987.0	8747.0	2360.0	84.7	230.0
opm2-z10-s4	-1489.0	-22681.0	-33269.0	18.2	299.0
opm2-z8-s0	-2220.0	-11328.0	-15775.0	6.7	50.7
sorrell7	-45.0	-160.0	-196.0	622.1	1770.0
sorrell8	-168.0	-324.0	-350.0	11.8	560.0

TABLE 7.1: Instances where SCIP needs significantly more time to compute a feasible point with similar quality

In most instances from Table 7.1, the additional time SCIP needs to compute a point of similar quality is quite significant. 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 3 rounds of diving take no longer than 30 seconds. For both listed instances of the problem *sorell*, the time difference is also quite remarkable and, in contrast to the problem *gsvm2rl12*, here the diving steps are crucial.

These examples demonstrate that in some cases, feasible rounding approaches combined with the introduced diving ideas can be very 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.

7.5 Conclusion and Outlook

In this chapter we investigated the effects of feasible rounding approaches when combined with branch-and-bound methods. We showed that the number of roundings obtainable within a search tree is non-decreasing and potentially increasing with increasing depth of the search tree.

Moreover, 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. Secondly, 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 instances is not granular in the root node but becomes granular in some child node explored by our diving strategies. Secondly, for granular nodes, 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 effects 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.

Finally, we 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 leave for future research.

Chapter 8

Conclusion and Directions for Future Research

An apparent conclusion from the preceding chapters is that the concept of inner parallel sets is widely applicable within mixed-integer optimization. Indeed, after the application of various enlargement and approximation techniques presented in this thesis, inner parallel sets of many problems from practice turned out to be nonempty. This enabled resorting to feasible rounding approaches based on this concept, which proved to be able to compute good feasible points for mixed-integer linear, convex and nonconvex optimization problems.

The possibility of a closed-form expression of inner parallel sets for mixed-integer linear problems allows a straight-forward application of algorithmic ideas in this context. This is also one of the main advantages of computing polyhedral outer approximations of the inner parallel set via inner parallel cutting planes in the case of mixed-integer convex optimization problems.

When dealing with nonconvex nonlinear constraint functions, the computation of an (inner) approximation of the inner parallel set based on Lipschitz constants is generally more intricate. To broaden the range of applications, in these cases we extended the geometrically intuitive idea of granularity (which is based on inner parallel sets) to the algorithmically more attractive concept of pseudo-granularity. This enabled the successful application of feasible rounding approaches to a significant number of mixed-integer quadratically constrained quadratic optimization problems from practice.

Generally, the presented approaches work best under the absence of equality constraints on integer variables in the problem formulation. We showed that if such constraints are present, inner parallel sets can be nonempty in a reduced variable space. Yet, this is less likely to happen compared to problems without equality constraints on integer variables in the original variable space.

Computationally, the inner parallel cutting plane method for mixed-integer convex optimization problems seems to be particularly promising, as well as diving strategies based on inner parallel sets. So far, we introduced and tested such diving strategies only on mixed-integer linear optimization problems; a combination of cutting planes and diving ideas is hence not part of this thesis but an interesting avenue for future research.

An integration of our diving ideas could also significantly improve the presented feasible rounding approach for mixed-integer nonlinear problems that uses the concept of pseudo-granularity. Here, the drawback that in the computation of Lipschitz constants we use global information to locally approximate the inner parallel set becomes less important when the sizes of considered boxes decrease.

Enlargement ideas of the relaxed feasible set are crucial for the applicability of feasible rounding approaches. The ideas presented in this thesis focused on additive relaxations of the defining functions of the feasible set. We briefly indicated that there is also the possibility of using structurally different functions. The general idea is to "reverse" the tightening ideas usually employed in mixed-integer optimization. This idea might have, in particular, the potential to make the introduced reduction scheme for problems where equality constraints on integer variables appear more widely applicable. Indeed, there might be ways to circumvent the dense structure of (the constraint matrices of) the reduced problem, which we suspect to be the main reason for empty inner parallel sets of many applications. This investigation of beneficial structural reformulations is beyond the scope of this thesis and also a promising field for future research.

Appendix A

Complementary Material for the Computational Studies

A.1 Computational Results for Plain Feasible Rounding Approaches

	FRA-SOR		FRA-SLOR		Gurobi	
name	time	objective	time	objective	time	objective
30_70_4.5_0.95_100	2.49	8885.00	0.30	19611.00	0.00	1144.00
50v-10	0.02	199236.75	0.00	1.420549e+07	0.00	31966.50
a1c1s1	0.02	23143.23	0.09	35639.32	0.09	19747.08
b2c1s1	0.05	76071.01	0.08	117226.42	0.19	62796.52
bg512142	0.07	3.984866e+06	0.00	1.851402e+08	0.00	9.399349e+07
buildingenergy	14.10	44205.67	2.92	54541.38	0.06	1.624119e+07
cov1075	0.08	77.00	0.00	120.00	0.00	56.00
dfn-gwin-UUM	0.00	227208.00	0.00	478056.00	0.00	174600.00
ex1010-pi	1.41	8599.00	0.03	25200.00	0.00	641.00
fast0507	1.08	55653.00	0.09	122425.00	0.02	351.00
fixnet6	0.00	92716.00	0.00	94378.00	0.00	20415.00
g200x740i	0.02	194475.00	0.00	197794.00	0.00	53393.00
ger50_17_trans	0.39	555975.25	0.03	657982.99	0.02	30068.24
germany50-DBM	1.11	606750.00	0.02	1.290160e+06	0.00	2.823860e+06
iis-100-0-cov	0.03	100.00	0.02	100.00	0.00	35.00
iis-bupa-cov	0.19	107.00	0.00	345.00	0.00	48.00
iis-pima-cov	0.39	153.00	0.02	768.00	0.00	44.00
janos-us-DDM	0.00	1.508461e+06	0.00	3.506885e+06	0.00	6.029231e+06
k16x240	0.00	177473.00	0.00	185233.00	0.00	24175.00
m100n500k4r1	0.02	-11.00	0.00	0.00	0.00	-18.00
macrophage	0.06	1582.00	0.02	1582.00	0.00	609.00
manna81	0.03	-12869.00	0.00	0.00	0.00	-6954.00
mas74	0.00	736774.15	0.00	1.000000e+12	0.00	157344.61
mas76	0.00	782652.58	0.00	1.000000e+12	0.00	157344.61
mc11	0.02	128960.00	0.00	128960.00	0.11	13509.00
methanosarcina	1.14	7302.00	0.09	7302.00	0.00	5046.00
mik.250-1-100.1	0.02	0.00	0.00	2.000000e+07	0.00	446229.00
modglob	0.00	2.153798e+07	0.00	6.688030e+08	0.00	6.896386e+08
n15-3	6.54	88891.00	9.73	1.776685e+08	426.81	66291.00
n3-3	0.22	40030.00	0.22	7.320524e+07	0.85	22030.00
n3700	0.02	8.057113e+07	0.03	8.065223e+07	0.00	3.034453e+06
n3705	0.02	7.980784e+07	0.05	7.987833e+07	0.00	2.756298e+06
n370a	0.02	8.075517e+07	0.05	8.080840e+07	0.00	3.144278e+06
n4-3	0.05	19275.00	0.03	3.480254e+07	0.11	12175.00
n9-3	0.13	28825.00	0.11	5.040473e+07	0.66	19025.00
neos-1112782	0.00	2.477444e+13	0.00	2.477444e+13	0.00	2.248990e+12
neos-1112787	0.00	2.178675e+13	0.00	2.178675e+13	0.00	1.702341e+12
neos-1225589	0.00	9.819371e+10	0.00	9.819371e+10	0.00	3.866793e+09
neos-1616732	0.00	200.00	0.00	200.00	0.00	173.00

TABLE A.1: A comparison of the feasible rounding approaches and Gurobi with regard to time (seconds) and objective value on unaltered models (I)

	FRA-SOR		FRA-SLOR		Gurobi	
name	time	objective	time	objective	time	objective
neos-932816	0.45	5.487565e+06	0.16	6.727970e+06	0.00	513000.00
neos-933638	2.52	5.074448e+06	0.08	6.077942e+06	0.00	484000.00
neos-933966	2.11	5.074471e+06	0.06	6.078073e+06	0.00	484000.00
neos-934278	1.86	4.618871e+06	0.05	5.576040e+06	0.00	468000.00
neos15	0.00	144796.91	0.00	3.610584e+08	0.00	2.790968e+07
npmv07	3.57	1.049283e+11	1.46	5.994301e+11	4.22	1.048114e+11
ns4-pr3	0.13	38220.00	0.02	607015.00	0.00	132965.00
ns4-pr9	0.08	36375.00	0.02	360295.00	0.00	105945.00
opm2-z10-s2	14.50	-1118.00	8.21	0.00	0.02	-8104.00
opm2-z11-s8	24.75	-1611.00	27.83	-2193.00	0.03	-9433.00
opm2-z12-s14	45.84	-1306.00	42.27	0.00	0.03	-11994.00
opm2-z12-s7	48.09	-1653.00	43.07	0.00	0.02	-12375.00
opm2-z7-s2	1.08	-1359.00	0.61	-1156.00	0.00	-3515.00
p100x588b	0.00	448087.00	0.00	454481.00	0.00	94028.00
p6b	0.00	0.00	0.00	0.00	0.00	-52.00
p80x400b	0.00	311096.00	0.00	316421.00	0.00	139650.00
pp08a	0.00	18100.00	0.02	27958.79	0.00	13560.00
pp08aCUTS	0.00	20030.46	0.00	23739.61	0.00	13560.00
qiu	0.03	3059.55	0.02	4127.36	0.08	1120.97
queens-30	1.37	0.00	0.02	0.00	0.02	-29.00
r80x800	0.00	26891.00	0.00	40827.00	0.00	23943.00
ramos3	9.13	1077.00	0.02	2187.00	0.00	542.00
ran14x18.disj-8	0.02	42607.74	0.00	43141.00	0.00	7657.00
ran14x18	0.00	42659.02	0.00	43246.99	0.00	8870.00
ran16x16	0.00	42914.02	0.02	43787.05	0.00	6587.00
set1ch	0.00	170115.59	0.02	244167.96	0.00	139113.00
set3-10	0.03	1.992090e+06	0.02	4.187314e+06	0.14	1.756477e+06
set3-15	0.06	1.925868e+06	0.02	3.955883e+06	0.20	1.610398e+06
set3-20	0.05	1.916274e+06	0.02	3.896786e+06	0.16	1.712935e+06
seymour.disj-10	0.46	677.00	0.02	1209.00	0.00	366.00
seymour	0.30	773.00	0.02	1372.00	0.00	502.00
stockholm	1.36	962.00	2.45	962.00	20.81	151.00
sts405	0.06	405.00	0.03	405.00	0.00	357.00
sts729	0.43	729.00	0.11	729.00	0.02	665.00
tanglegram1	9.42	34171.00	0.13	34171.00	0.00	7843.00
tanglegram2	1.14	4490.00	0.02	4490.00	0.02	2160.00
toll-like	0.23	2204.00	0.02	2204.00	0.00	1155.00
zib54-UUE	0.23	2.404391e+07	0.02	2.404391e+07	0.56	1.888427e+07

TABLE A.2: A comparison of the feasible rounding approaches and Gurobi with regard to time (seconds) and objective value on unaltered models (II)

A.2 Computational Results for Inner Parallel Cuts

	SCIP		*B-Hyb*		*B-OA*		*SCIP*		*B-Hyb*		*B-OA*	
	t(RICP)	t(OM)	t(RICP)	t(OM)	t(RICP)	t(OM)	lb(RICP)	lb(OM)	lb(RICP)	lb(OM)	lb(RICP)	lb(OM)
cvxnonsep_normcon20	1.44	1.33	65.32	>1800	14.47	98.94	-21.75	-21.75	-21.75	-21.76	-21.75	-21.75
cvxnonsep_normcon20r	0.17	0.20	0.33	0.27	0.07	0.06	-21.75	-21.75	-21.75	-21.75	-21.75	-21.75
cvxnonsep_normcon30	15.53	15.77	>1800	>1800	>1800	>1800	-34.24	-34.24	-34.43	-34.43	-34.40	-34.43
cvxnonsep_normcon30r	0.28	0.25	1.05	0.34	0.11	0.08	-34.24	-34.24	-34.24	-34.24	-34.24	-34.24
cvxnonsep_normcon40	601.36	681.81	>1800	>1800	>1800	>1800	-32.63	-32.63	-32.89	-32.89	-32.89	-32.89
cvxnonsep_normcon40r	0.31	0.34	1.02	0.38	0.16	0.04	-32.63	-32.63	-32.63	-32.63	-32.63	-32.63
cvxnonsep_nsig20	2.14	272.68	156.60	1261.60	14.80	10.06	80.95	80.95	80.95	80.95	80.95	80.95
cvxnonsep_nsig20r	0.10	0.08	1.23	0.09	0.07	0.05	80.95	80.95	80.95	80.95	80.95	80.95
cvxnonsep_nsig30	>1800	>1800	>1800	>1800	>1800	>1800	128.07	121.17	130.48	130.48	130.60	130.61
cvxnonsep_nsig30r	0.28	0.16	307.29	>1800	0.14	0.11	156.43	156.43	156.43	156.37	156.43	156.43
cvxnonsep_nsig40	>1800	>1800	>1800	>1800	>1800	>1800	130.57	94.71	133.74	133.74	133.74	133.85
cvxnonsep_nsig40r	0.25	0.21	>1800	>1800	0.27	0.09	133.96	133.96	133.83	133.80	133.96	133.96
cvxnonsep_pcon20	0.99	0.64	7.78	0.77	1.49	1.55	-21.51	-21.51	-21.51	-21.51	-21.51	-21.51
cvxnonsep_pcon20r	0.15	0.11	0.30	0.06	0.15	0.08	-21.51	-21.51	-21.51	-21.51	-21.51	-21.51
cvxnonsep_pcon30	6.99	6.68	300.90	>1800	110.25	80.47	-35.99	-35.99	-35.99	-36.16	-35.99	-35.99
cvxnonsep_pcon30r	0.07	0.33	0.47	0.22	0.36	0.43	-35.99	-35.99	-35.99	-35.99	-35.99	-35.99
cvxnonsep_pcon40	19.96	43.72	1652.65	>1800	829.18	450.75	-46.60	-46.60	-46.60	-46.78	-46.60	-46.60
cvxnonsep_pcon40r	0.40	0.53	1.05	0.26	0.31	0.12	-46.60	-46.60	-46.60	-46.47	-46.60	-46.60
cvxnonsep_psig20r	0.19	0.11	>1800	0.42	0.11	0.06	95.90	95.90	95.81	95.90	95.90	95.90
cvxnonsep_psig30r	0.19	0.16	>1800	0.20	0.18	0.07	79.00	79.00	78.88	79.00	79.00	79.00
cvxnonsep_psig40r	0.22	0.28	>1800	0.23	0.19	0.07	86.55	86.55	86.43	86.55	86.55	86.55
du-opt	0.89	5.86	24.93	12.62	3.01	2.95	3.56	3.56	3.56	3.56	3.56	3.56
ex1223a	0.02	0.02	0.03	0.03	0.00	0.00	4.58	4.58	4.58	4.58	4.58	4.58
ex1223b	0.03	0.11	0.05	0.04	0.04	0.02	4.58	4.58	4.58	5.58	4.58	4.58
ex4	1.19	1.60	0.52	0.66	0.33	0.30	-8.06	-8.06	-8.06	48233.05	-8.06	-8.06
gbd	0.00	0.00	0.01	0.01	0.00	0.00	2.20	2.20	2.20	2.20	2.20	2.20
nvs03	0.00	0.01	0.02	0.01	0.00	0.00	16.00	16.00	16.00	16.00	16.00	16.00
nvs10	0.00	0.00	0.03	0.01	0.00	0.00	-310.80	-310.80	-310.80	-296.80	-310.80	-310.80
nvs11	0.01	0.03	0.02	0.02	0.01	0.03	-431.00	-431.00	-431.00	-431.00	-431.00	-431.00
nvs12	0.02	0.04	0.19	0.03	0.02	0.03	-481.20	-481.20	-481.20	-481.20	-481.20	-481.20
nvs15	0.00	0.02	0.02	0.03	0.01	0.01	1.00	1.00	1.00	1.00	1.00	1.00
portfol_buyin	0.09	0.18	0.03	0.08	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
smallinvDAXr1b150-165	0.43	0.37	38.07	>1800	136.00	101.16	88.10	88.10	88.10	88.08	88.10	88.10
smallinvDAXr1b200-220	0.32	0.33	49.88	>1800	135.84	159.22	156.60	156.60	156.60	156.59	156.60	156.60

Continued on next page

	SCIP		*B-Hyb*		*B-OA*		*SCIP*		*B-Hyb*		*B-OA*	
	t(RICP)	t(OM)	t(RICP)	t(OM)	t(RICP)	t(OM)	lb(RICP)	lb(OM)	lb(RICP)	lb(OM)	lb(RICP)	lb(OM)
smallinvDAXr2b150-165	0.49	0.40	116.96	>1800	96.22	80.76	88.10	88.10	88.10	88.08	88.10	88.10
smallinvDAXr2b200-220	0.31	0.36	99.74	>1800	130.84	143.27	156.60	156.60	156.60	156.59	156.60	156.60
smallinvDAXr3b150-165	0.44	0.39	47.91	>1800	100.45	107.92	88.10	88.10	88.10	88.08	88.10	88.10
smallinvDAXr3b200-220	0.40	0.38	86.55	1405.79	87.81	96.35	156.60	156.60	156.60	156.60	156.60	156.60
smallinvDAXr4b150-165	0.50	0.39	34.55	>1800	70.89	153.74	88.10	88.10	88.10	88.08	88.10	88.10
smallinvDAXr4b200-220	0.28	0.38	37.50	>1800	103.11	150.65	156.60	156.60	156.60	156.59	156.60	156.60
smallinvDAXr5b150-165	0.50	0.39	24.78	>1800	78.57	90.96	88.10	88.10	88.10	88.08	88.10	88.10
smallinvDAXr5b200-220	0.44	0.38	56.77	>1800	105.79	171.73	156.60	156.60	156.60	156.59	156.60	156.60
squfl010-025	1049.77	413.89	10.73	1.05	295.49	418.10	214.11	214.11	214.11	214.11	214.11	214.11
squfl010-040	>1800	>1800	100.62	12.45	1328.76	1232.22	136.84	184.14	255.75	240.60	240.60	240.60
squfl010-080	>1800	>1800	208.62	40.19	>1800	>1800	258.90	240.54	571.55	509.71	397.03	365.99
squfl015-060	>1800	>1800	65.80	45.97	>1800	>1800	152.47	207.82	396.81	366.62	231.55	205.36
squfl015-080	>1800	>1800	NaN	648.13	>1800	>1800	172.58	155.88	NaN	402.49	207.31	197.74
squfl020-040	>1800	>1800	354.36	96.77	>1800	>1800	98.14	128.38	224.70	209.25	147.38	127.90
squfl020-050	>1800	>1800	116.05	961.13	>1800	>1800	99.24	122.33	235.74	230.20	116.24	113.64
squfl020-150	>1800	>1800	>1800	>1800	>1800	>1800	226.34	0.00	297.58	286.17	229.32	227.04
squfl025-025	>1800	>1800	17.62	11.21	>1800	>1800	80.00	141.20	182.53	178.63	107.65	100.69
squfl025-030	>1800	>1800	24.75	12.97	>1800	>1800	81.33	139.84	214.73	205.50	138.37	117.86
squfl025-040	>1800	>1800	989.93	22.57	>1800	>1800	76.87	67.31	212.99	199.16	92.00	88.37
squfl030-100	>1800	>1800	>1800	>1800	>1800	>1800	123.89	0.00	151.99	151.69	127.51	126.47
squfl030-150	>1800	>1800	>1800	>1800	>1800	>1800	158.93	0.00	159.05	177.94	158.93	158.93
squfl040-080	>1800	>1800	NaN	>1800	>1800	>1800	91.49	0.00	NaN	105.04	92.04	91.91
st_miqp1	0.00	0.00	0.02	0.02	0.00	0.00	281.00	281.00	281.00	281.00	281.00	281.00
st_miqp2	0.00	0.00	0.04	NaN	0.01	NaN	2.00	2.00	2.00	NaN	2.00	NaN
st_miqp3	0.00	0.00	0.07	10.00	0.00	0.00	-6.00	-6.00	-6.00	-6.00	-6.00	-6.00
st_miqp4	0.00	0.01	0.13	10.01	0.01	0.00	-4574.00	-4574.00	-4574.00	-4574.00	-4574.00	-4574.00
st_test1	0.00	0.00	0.03	10.00	0.01	0.01	0.00	0.00	-0.00	0.00	-0.00	0.00
st_test2	0.00	0.00	0.07	10.02	0.00	0.00	-9.25	-9.25	-9.25	-9.25	-9.25	-9.25
st_test3	0.00	0.00	0.08	10.01	0.00	0.00	-7.00	-7.00	-7.00	-7.00	-7.00	-7.00
st_test4	0.00	0.00	0.08	NaN	0.00	NaN	-7.00	-7.00	-7.00	NaN	-7.00	NaN
st_testgr1	0.02	0.03	0.36	0.10	0.14	0.23	-12.81	-12.81	-12.81	-12.81	-12.81	-12.81
st_testgr3	0.03	0.03	0.06	10.06	0.14	0.15	-20.59	-20.59	-20.59	-20.59	-20.59	-20.59
st_testph4	0.00	0.00	0.02	10.00	0.00	0.00	-80.50	-80.50	-80.50	-80.50	-80.50	-80.50

TABLE A.3: Solver times (t) and lower bounds (lb) for original models (OM) and models with reversed inner parallel cuts (RICPs). NaN means that a solver error occurred

A.3 Computational Results for Equality Constrained Problems

	data					time			
instance	$\|R\|$	q	n	$q-t$	m	$C_{R,:}$	prref	HNF	P^s_{red}
22433	0	198	198	0.00	231.00	0.01	0.01	0.00	0.00
23588	0	137	137	0.00	231.00	0.01	0.00	0.00	0.00
aflow30a	29	29	421	29.00	421.00	0.00	0.00	68.93	0.00
aflow40b	39	39	1364	39.00	1364.00	0.01	0.00	1598.39	0.00
arki001	0	20	850	0.00	538.00	0.06	0.00	0.00	0.01
assign1-10-4	0	62	52	61.00	520.00	0.00	0.00	305.79	0.01
assign1-5-8	0	31	26	30.00	130.00	0.00	0.00	7.16	0.00
b-ball	0	19	12	8.00	88.00	0.00	0.00	2.70	0.00
bc	784	1	1268	0.00	483.00	0.29	0.03	0.00	0.11
bc1	784	1	1499	0.00	252.00	0.30	0.00	0.00	0.10
biella1	0	1197	1218	0.00	6110.00	0.09	0.00	0.00	0.05
bienst1	112	16	477	4.00	28.00	0.00	0.00	0.13	0.00
bienst2	112	16	470	5.00	35.00	0.00	0.00	0.20	0.00
binkar10_1	0	1016	2128	0.00	170.00	0.00	0.08	0.00	0.00
blend2	0	89	89	0.00	264.00	0.00	0.00	0.00	0.00
blp-ar98	0	215	215	0.00	15806.00	15.76	0.00	0.00	0.04
blp-ic97	0	92	92	0.00	9753.00	5.92	0.00	0.00	0.02
blp-ic98	0	90	90	0.00	13550.00	12.17	0.00	0.00	0.04
blp-ir98	0	66	66	0.00	6031.00	2.39	0.00	0.00	0.01
bppc4-08	0	20	2	20.00	1454.00	0.00	0.00	560.28	0.00
bppc8-02	0	20	2	20.00	230.00	0.00	0.00	6.28	0.00
bppc8-09	0	18	2	18.00	429.00	0.00	0.00	23.06	0.00
breastcancer-regularized	0	4	9	4.00	706.00	0.01	0.00	1401.02	0.04
cdma	0	60	3656	0.00	4235.00	0.14	0.01	0.00	0.02
control20-5-10-5	410	20	1210	20.00	100.00	0.01	0.00	1.84	0.01
control30-3-2-3	122	30	242	30.00	90.00	0.00	0.00	1.32	0.00
control30-5-10-4	610	30	1810	30.00	150.00	0.00	0.00	3.30	0.00
csched007	0	301	301	50.00	1457.00	0.01	0.01	582.97	0.01
csched008	0	301	252	50.00	1284.00	0.00	0.01	341.57	0.01
csched010	0	301	301	50.00	1457.00	0.00	0.00	623.88	0.01
dano3_3	1176	48	13804	3.00	69.00	0.38	0.02	1.16	0.01
dano3_5	1176	48	13758	5.00	115.00	0.38	0.02	2.23	0.01
dano3mip	1176	48	13321	47.00	552.00	0.38	0.00	839.58	0.01
danoint	128	16	465	15.00	56.00	0.00	0.00	1.42	0.00
dc1c	0	1638	1659	0.00	8380.00	0.28	0.00	0.00	0.08
dc1l	0	1638	1659	0.00	35638.00	3.50	0.00	0.00	1.27
dcmulti	75	3	473	3.00	75.00	0.00	0.00	0.49	0.00
dell	0	31	521	0.00	105.00	0.01	0.00	0.00	0.00
dolom1	0	1693	1892	0.00	9720.00	0.35	0.00	0.00	0.12
dsbmip	403	40	1745	40.00	192.00	0.00	0.00	4.77	0.00
ej	0	1	0	1.00	3.00	0.03	0.01	0.33	0.01
enlight11	0	121	0	121.00	242.00	0.00	0.00	176.04	0.00
enlight4	0	16	0	16.00	32.00	0.00	0.00	0.70	0.00
enlight8	0	64	0	64.00	128.00	0.00	0.00	28.47	0.00
enlight9	0	81	0	81.00	162.00	0.00	0.00	55.22	0.00
enlight_hard	0	100	0	100.00	200.00	0.00	0.00	102.83	0.00
eva1aprime5x5opt	112	192	1312	192.00	400.00	0.00	0.00	76.78	0.13
fastxgemm-n2r6s0t2	0	144	736	0.00	48.00	0.00	0.00	0.00	0.01
fastxgemm-n2r7s4t1	0	168	848	0.00	56.00	0.00	0.00	0.00	0.01
gen	0	150	720	6.00	150.00	0.00	0.00	6.07	0.00
germanrr	0	239	239	0.00	10574.00	6.84	0.00	0.00	0.03
glass4	0	36	20	36.00	302.00	0.01	0.00	19.64	0.00
gmu-35-40	0	5	5	0.00	1200.00	0.00	0.03	0.00	0.00
gmu-35-50	0	5	5	0.00	1914.00	0.01	0.00	0.00	0.01
gmut-76-40	0	6	6	0.00	24332.00	1.32	0.00	0.00	0.11
graphdraw-domain	0	45	54	45.00	200.00	0.00	0.00	5.69	0.02
graphdraw-gemcutter	0	28	38	28.00	128.00	0.00	0.00	2.13	0.01
ic97_tension	0	319	523	0.00	180.00	0.00	0.07	0.00	0.00
icir97_tension	1	1202	1659	0.00	835.00	0.01	3.39	0.00	0.00
mad	0	30	20	20.00	200.00	0.00	0.00	11.73	0.00
markshare1	0	6	12	0.00	50.00	0.02	0.01	0.00	0.01
markshare2	0	7	14	0.00	60.00	0.00	0.00	0.00	0.00

Continued on next page

	data					time			
instance	$\|R\|$	q	n	$q-t$	m	$C_{R,:}$	prref	HNF	P^s_{red}
markshare_4_0	0	4	4	0.00	30.00	0.00	0.01	0.00	0.00
markshare_5_0	0	5	5	0.00	40.00	0.00	0.00	0.00	0.00
milo-v12-6-r1-58-1	1140	300	2940	0.00	1500.00	0.00	0.00	0.00	0.08
milo-v12-6-r1-75-1	1463	385	3773	0.00	1925.00	0.00	0.00	0.00	0.18
milo-v12-6-r2-40-1	798	210	1848	0.00	840.00	0.00	0.00	0.00	0.04
milo-v13-4-3d-3-0	216	120	396	60.00	120.00	0.00	0.00	2.84	0.00
milo-v13-4-3d-4-0	288	160	528	80.00	160.00	0.00	0.00	4.87	0.01
misc04inf	0	311	4867	0.00	30.00	0.04	0.00	0.00	0.00
misc05inf	0	29	62	15.00	74.00	0.00	0.00	2.14	0.00
misc07	0	35	1	31.00	259.00	0.00	0.00	67.54	0.00
mkc	0	2	2	0.00	5323.00	0.33	0.00	0.00	0.01
mkc1	0	2	2238	0.00	3087.00	0.34	0.01	0.00	0.00
mod011	4368	16	10879	0.00	96.00	0.06	0.00	0.00	0.00
nag	1	1374	1499	35.00	1385.00	0.00	0.00	108.28	0.01
neos-1396125	396	21	1032	21.00	129.00	0.00	0.00	7.41	0.00
neos-1420790	0	2280	4386	90.00	540.00	0.00	0.00	62.01	0.03
neos-1425699	0	26	20	6.00	85.00	0.00	0.00	1.33	0.00
neos-1445532	0	923	12407	0.00	1999.00	0.02	0.00	0.00	0.01
neos-2629914-sudost	0	32	240	31.00	256.00	0.00	0.00	34.37	0.06
neos-2978193-inde	324	8	20736	8.00	64.00	0.09	0.00	1.71	0.02
neos-2978205-isar	324	8	103680	8.00	320.00	1.35	0.00	88.32	0.06
neos-3046601-motu	0	272	19	136.00	289.00	0.00	0.00	29.05	0.00
neos-3046615-murg	0	240	18	120.00	256.00	0.00	0.00	15.52	0.00
neos-3065804-namu	0	1095	2551	0.00	2190.00	0.03	0.71	0.00	2.07
neos-3116779-oban	0	1	1	0.00	5140.00	0.00	0.03	0.00	0.01
neos-3118745-obra	0	1	1	0.00	1130.00	0.00	0.00	0.00	0.02
neos-3135526-osun	0	32	22	21.00	170.00	0.00	0.00	13.14	0.00
neos-3209462-rhin	0	176	56070	43.00	383.00	0.00	0.01	36.24	0.09
neos-3352863-ancoa	0	1	1	0.00	20045.00	0.00	0.00	0.00	4.89
neos-3372571-onahau	0	3601	12901	66.00	185.00	12.37	0.05	3.22	0.16
neos-3421095-cinca	1	24	737	24.00	159.00	0.00	0.00	3.67	0.01
neos-3610040-iskar	0	1	345	0.00	85.00	0.00	0.00	0.00	0.00
neos-3610051-istra	0	1	729	0.00	76.00	0.00	0.00	0.00	0.02
neos-3610173-itata	0	1	767	0.00	77.00	0.00	0.00	0.00	0.02
neos-3611447-jijia	0	1	387	0.00	85.00	0.00	0.00	0.00	0.01
neos-3611689-kaihu	0	1	333	0.00	88.00	0.00	0.00	0.00	0.00
neos-3627168-kasai	463	2	927	0.00	535.00	0.02	0.00	0.00	0.00
neos-3660371-kurow	0	1974	4578	24.00	144.00	0.00	1.27	3.43	0.00
neos-3699377-maori	740	180	15377	0.00	1856.00	0.12	0.02	0.00	0.04
neos-3703351-marne	584	140	7445	0.00	1382.00	0.03	0.00	0.00	0.01
neos-3761878-oglio	584	140	5177	0.00	1312.00	0.01	0.01	0.00	0.01
neos-3762025-ognon	428	100	3725	0.00	948.00	0.01	0.00	0.00	0.01
neos-4333596-skien	199	26	545	25.00	460.00	0.00	0.00	43.40	0.00
neos-480878	24	12	345	12.00	189.00	0.00	0.00	2.19	0.00
neos-5041756-cobark	0	300	301	0.00	60000.00	0.50	0.00	0.00	0.03
neos-5045105-creuse	0	48	68	0.00	3780.00	0.05	0.00	0.00	0.09
neos-5051588-culgoa	0	72	138	0.00	3780.00	0.06	0.00	0.00	0.01
neos-5075914-elvire	1118	118	2365	0.00	2638.00	0.00	0.00	0.00	0.00
neos-5140963-mincio	0	28	13	27.00	183.00	0.00	0.00	34.47	0.00
neos-5182409-nasivi	0	264	405	20.00	1600.00	0.03	0.00	1121.98	0.01
neos-5188808-nattai	5544	146	14256	146.00	288.00	0.00	0.00	29.97	0.02
neos-574665	0	456	492	0.00	248.00	0.00	0.01	0.00	0.02
neos-585467	150	1116	1270	116.00	846.00	0.07	1.83	321.14	0.02
neos-595904	132	20	3360	20.00	1148.00	0.02	0.00	229.27	0.00
neos-807639	0	850	950	0.00	80.00	0.00	0.28	0.00	0.01
neos-860300	0	20	1	18.00	1384.00	0.02	0.00	2813.93	0.02
neos-911970	0	35	48	35.00	840.00	0.00	0.00	138.06	0.01
neos-935234	0	139	2779	0.00	7530.00	0.01	0.00	0.00	0.24
neos-935769	0	139	2779	0.00	7020.00	0.01	0.00	0.00	0.21
neos-983171	0	158	2408	0.00	6557.00	0.01	0.00	0.00	0.28
neos16	0	10	0	10.00	377.00	0.00	0.00	8.66	0.00
neos2	30	13	1061	13.00	1040.00	0.01	0.00	233.50	0.00
neos4	453	684	5712	0.00	17172.00	0.02	0.00	0.00	0.03
neos6	0	223	446	0.00	8340.00	0.79	0.00	0.00	0.11
neos859080	4	39	0	39.00	160.00	0.00	0.00	21.64	0.00
newdano	112	16	449	15.00	56.00	0.00	0.00	1.43	0.00
nh97_tension	0	737	958	0.00	618.00	0.00	0.59	0.00	0.00
noswot	0	2	28	0.00	100.00	0.00	0.00	0.00	0.00

Continued on next page

	data					time			
instance	$\|R\|$	q	n	$q-t$	m	$C_{R,:}$	prref	HNF	P^s_{red}
nsa	0	37	352	0.00	36.00	0.00	0.00	0.00	0.00
nsr8k	0	6195	6316	0.00	32040.00	0.57	0.00	0.00	0.34
opt1217	0	48	1	48.00	768.00	0.02	0.00	269.05	0.00
p2m2p1m1p0n100	0	1	1	0.00	100.00	0.00	0.00	0.00	0.00
pg	0	100	2600	0.00	100.00	0.00	0.00	0.00	0.00
pigeon-08	0	48	72	40.00	272.00	0.00	0.00	6.53	0.00
pigeon-10	0	60	90	50.00	400.00	0.00	0.00	14.18	0.00
pigeon-13	0	78	117	65.00	637.00	0.00	0.00	35.00	0.00
pigeon-16	0	96	144	80.00	928.00	0.00	0.00	90.37	0.00
pk1	0	15	31	0.00	55.00	0.00	0.00	0.00	0.00
ponderthis0517-inf	0	26	0	26.00	975.00	0.00	0.00	262.45	0.00
prod1	1	7	101	7.00	149.00	0.00	0.00	1.77	0.00
prod2	1	10	101	10.00	200.00	0.00	0.00	4.76	0.00
pw-myciel4	0	45	0	45.00	1059.00	0.00	0.00	259.04	0.01
r4l4-02-tree-bounds-50	0	4768	6700	0.00	4768.00	0.00	0.00	0.00	0.00
rmatr100-p10	0	1	7259	1.00	100.00	0.00	0.00	21.86	0.01
rmatr100-p5	0	1	8684	1.00	100.00	0.00	0.00	18.51	0.01
rmatr200-p10	0	1	35054	1.00	200.00	0.00	0.00	146.73	0.08
rmatr200-p20	0	1	29405	1.00	200.00	0.00	0.00	148.00	0.06
rmatr200-p5	0	1	37616	1.00	200.00	0.00	0.00	143.96	0.12
roll3000	0	174	428	0.00	738.00	0.02	0.00	0.00	0.01
rout	0	31	241	15.00	315.00	0.01	0.00	44.35	0.00
sct1	0	1290	12574	0.00	10312.00	1.30	0.00	0.00	0.03
sct2	0	360	3013	0.00	2872.00	0.09	0.00	0.00	0.01
sct31	0	553	2995	0.00	5520.00	0.35	0.00	0.00	0.01
sct32	0	553	2039	0.00	7728.00	0.70	0.00	0.00	0.01
sct5	0	1279	14261	0.00	23004.00	7.04	0.00	0.00	0.03
siena1	0	1808	1966	0.00	11775.00	0.33	0.00	0.00	0.19
supportcase30	0	4	0	4.00	1024.00	0.01	0.00	3485.22	0.00
supportcase35	0	2880	12365	0.00	576.00	0.01	0.16	0.00	0.62
timtab1	0	171	226	0.00	171.00	0.00	0.02	0.00	0.00
timtab1CUTS	0	171	226	0.00	171.00	0.00	0.01	0.00	0.00
trento1	0	1248	1272	0.00	6415.00	0.20	0.00	0.00	0.07
tw-myciel4	0	71	0	71.00	760.00	0.00	0.00	49.68	0.44
uccase8	0	12381	27837	0.00	9576.00	0.11	4.24	0.00	0.07
uct-subprob	20	881	1877	0.00	379.00	0.00	0.00	0.00	0.00
unitcal_7	2890	17	22899	0.00	2856.00	0.06	0.00	0.00	0.02
xmas10-2	2	4645	7200	26.00	900.00	0.00	1.31	30.26	0.03
enlight13	0	169	0	169.00	338.00	0.00	0.00	445.97	0.01
enlight14	0	196	0	196.00	392.00	0.00	0.00	1000.28	0.00
gmut-77-40	0	6	6	0.00	24332.00	1.27	0.02	0.00	0.11
go19	0	80	0	80.00	441.00	0.00	0.00	19.64	0.04
lrsa120	0	119	3600	0.00	239.00	0.01	0.00	0.00	0.01
neos-1620770	0	35	0	35.00	792.00	0.00	0.00	139.77	0.01
neos-506422	0	7	2464	7.00	63.00	0.00	0.00	0.51	0.00
neos-820146	0	15	0	15.00	600.00	0.00	0.00	40.93	0.00
neos-820157	0	120	0	75.00	1200.00	0.00	0.00	325.69	0.00
neos-847302	0	141	8	141.00	729.00	0.00	0.00	1398.33	0.00
neos-911880	0	35	48	35.00	840.00	0.00	0.00	136.07	0.00
neos-935627	0	139	2779	0.00	7522.00	0.01	0.00	0.00	0.19
neos-937511	0	160	2770	0.00	8562.00	0.01	0.00	0.00	0.24
neos-937815	0	160	2770	0.00	8876.00	0.01	0.00	0.00	0.31
neos-941262	0	160	2770	0.00	6710.00	0.01	0.00	0.00	0.30
neos-948126	0	156	2586	0.00	6965.00	0.01	0.00	0.00	0.35
neos-984165	0	155	2405	0.00	6478.00	0.01	0.00	0.00	0.27
neos858960	4	47	0	47.00	160.00	0.00	0.00	123.25	0.00
ns1702808	6	82	138	76.00	666.00	0.00	0.00	674.91	0.00
ns1766074	0	20	10	19.00	90.00	0.00	0.00	4.97	0.00
ns2081729	0	30	61	30.00	600.00	0.00	0.00	11.66	0.01
pigeon-11	0	66	99	55.00	473.00	0.00	0.00	19.21	0.00
pigeon-12	0	72	108	60.00	552.00	0.00	0.00	27.16	0.00
uc-case3	0	13725	26493	0.00	11256.00	0.39	8.17	0.00	0.08

TABLE A.4: Run times of each reduction step for reducible problems

A.4 Computational Results for Diving Methods

	granular			objective				iterations	
	root	greedy	random	root	greedy	random	best known	greedy	random
30_70_45_05_100	True	True	True	9473.00	3840.00	4488.00	9.00	0	0
30_70_45_095_100	True	True	True	8778.00	3310.00	4119.00	3.00	0	0
30_70_45_095_98	True	True	True	9195.00	4344.00	4175.00	12.00	0	0
50v-10	True	True	True	199236.75	30232.08	97076.88	3311.18	0	0
a1c1s1	True	True	True	21033.23	21029.39	21029.39	11503.44	0	0
a2c1s1	True	True	True	20866.08	20865.33	20865.33	10889.14	0	0
ab51-40-100	True	True	True	-1024393739.00	-5583762710.00	-2814631027.00	-10420305975.00	0	0
ab67-40-100	True	True	True	-1278992345.00	-4600773056.00	-4396351421.00	-11186253442.00	0	0
ab69-40-100	True	True	True	-1172879898.00	-3498179839.00	-2658682925.00	-11186281442.00	0	0
ab71-20-100	True	True	True	-1740358396.00	-6891846490.00	-4870493333.00	-10420305975.00	0	0
ab72-40-100	True	True	True	-1298844792.00	-3687915681.00	-3608237166.00	-11186620442.00	0	0
app3	False	False	True	inf	inf	6449642.38	5751714.33	25	25
australia-abs-cta	False	True	True	inf	4613.65	3593.32	106.90	30	30
b1c1s1	True	True	True	69466.45	69376.94	69336.06	24544.25	0	0
b2c1s1	True	True	True	69351.01	67975.52	66085.52	25687.90	0	0
beasleyC1	True	True	True	102.00	102.00	102.00	85.00	0	0
beasleyC2	True	True	True	232.00	232.00	217.00	144.00	0	0
beasleyC3	False	True	True	6844.00	964.00	4710.00	754.00	1	20
berlin	True	True	True	1921.00	1921.00	1321.00	1044.00	0	0
berlin_5_8_0	False	True	True	inf	95.00	92.00	62.00	4	15
bg512142	True	True	True	3968845.86	293461.73	267769.75	184202.75	0	0
bmoipr2	False	True	True	383315500000.00	108118487164.11	164298702888.88	-46416168.30	20	30
brasil	True	True	True	32720.00	32720.00	19702.00	13655.00	0	0
cbs-cta	False	True	False	inf	110276062.82	inf	0.00	30	30
cdc7-4-3-2	True	True	True	0.00	-133.00	-136.00	-289.00	0	0
cod105	True	True	True	0.00	-2.00	-4.00	-12.00	0	0
core2536-691	False	False	True	inf	inf	7059.00	689.00	30	9
cost266-UUE	True	True	True	42188320.70	42008592.16	42014519.80	25148940.56	0	0
cvs08r139-94	True	True	True	-86.00	-86.00	-86.00	-116.00	0	0
cvs16r106-72	True	True	True	-33.00	-51.00	-39.00	-81.00	0	0
cvs16r128-89	True	True	True	-79.00	-79.00	-79.00	-97.00	0	0
cvs16r70-62	True	True	True	-15.00	-18.00	-15.00	-42.00	0	0
cvs16r89-60	True	True	True	-17.00	-32.00	-32.00	-65.00	0	0
dale-cta	False	True	True	inf	584.22	584.22	0.00	30	30

Continued on next page

	granular			objective				iterations	
	root	greedy	random	root	greedy	random	best known	greedy	random
dg012142	True	True	True	77158148.08	12353298.33	12347007.75	2300867.00	0	0
ex1010-pi	True	True	True	8595.00	2813.00	4305.00	235.00	0	0
fast0507	True	True	True	55643.00	2520.00	28290.00	174.00	0	0
g200x740	True	True	True	45614.00	45558.00	45614.00	44316.00	0	0
gen-ip002	True	True	True	-3543.64	-4573.41	-4550.52	-4783.73	0	0
gen-ip016	True	True	True	-7700.61	-8783.06	-9052.59	-9476.16	0	0
gen-ip021	True	True	True	3014.67	2738.20	2478.29	2361.45	0	0
gen-ip036	True	True	True	-3827.20	-4552.37	-4500.01	-4606.68	0	0
gen-ip054	True	True	True	11138.24	7593.88	7923.48	6840.97	0	0
ger50-17-ptp-pop-3t	True	True	True	14477.03	13687.20	9191.84	5231.11	0	0
ger50-17-ptp-pop-6t	True	True	True	17651.44	16456.02	11865.07	8942.63	0	0
ger50-17-trans-dfn-3t	True	True	True	553623.65	492635.07	256747.00	3969.43	0	0
ger50-17-trans-pop-3t	True	True	True	553803.46	495392.25	263898.10	4038.44	0	0
ger50_17_trans	True	True	True	555975.25	503750.31	265018.82	7393.26	0	0
germany50-UUM	True	True	True	751380.00	688720.00	666830.00	628490.00	0	0
glass-sc	True	True	True	74.00	47.00	51.00	23.00	0	0
gr4x6	True	True	True	284.05	252.15	222.15	202.35	0	0
gsvm2rl11	True	True	True	43040.05	42693.90	41964.44	18121.64	0	0
gsvm2rl12	True	True	True	50.00	42.17	36.74	22.12	0	0
gsvm2rl3	True	True	True	0.60	0.60	0.60	0.34	0	0
gsvm2rl5	True	True	True	10.00	10.00	10.00	5.42	0	0
gsvm2rl9	True	True	True	15597.47	15597.47	14896.93	7438.18	0	0
iis-glass-cov	True	True	True	73.00	46.00	50.00	21.00	0	0
iis-hc-cov	True	True	True	78.00	49.00	50.00	17.00	0	0
istanbul-no-cutoff	False	True	True	330.32	330.32	322.56	204.08	25	30
k16x240b	True	True	True	12874.00	12874.00	12779.00	11393.00	0	0
khb05250	True	True	True	126786075.50	120511827.00	122443058.00	106940226.00	0	0
manna81	True	True	True	-12867.00	-13162.00	-13162.00	-13164.00	0	0
mas74	True	True	True	736774.15	50264.53	28886.86	11801.19	0	0
mas76	True	True	True	782652.58	69745.40	64247.85	40005.05	0	0
mc11	True	True	True	13548.00	13548.00	13167.00	11689.00	0	0
mc7	True	True	True	5884.00	5875.00	4740.00	3417.00	0	0
mc8	True	True	True	1994.00	1977.00	1865.00	1566.00	0	0
mik-250-20-75-1	True	True	True	0.00	0.00	0.00	-49716.00	0	0
mik-250-20-75-2	True	True	True	0.00	0.00	0.00	-50768.00	0	0
mik-250-20-75-3	True	True	True	0.00	0.00	0.00	-52242.00	0	0
mik-250-20-75-4	True	True	True	0.00	0.00	0.00	-52301.00	0	0

Continued on next page

	granular			objective				iterations	
	root	greedy	random	root	greedy	random	best known	greedy	random
mik-250-20-75-5	True	True	True	0.00	0.00	0.00	-51532.00	0	0
n13-3	True	True	True	20570.00	17275.00	17325.00	13385.00	0	0
n3700	True	True	True	1831715.07	1654139.00	1426174.05	1227629.00	0	0
n3705	True	True	True	1847346.12	1597719.00	1442446.00	1225465.00	0	0
n3707	True	True	True	1788849.26	1625368.00	1354690.01	1186691.00	0	0
n3709	True	True	True	1811682.71	1676838.00	1405693.00	1207965.00	0	0
n370b	True	True	True	1911867.06	1669199.00	1442808.00	1236963.00	0	0
n5-3	True	True	True	16325.00	12725.00	14285.00	8105.00	0	0
n6-3	True	True	True	25100.00	19400.00	21550.00	15175.00	0	0
n7-3	True	True	True	22010.00	17890.00	19325.00	15426.00	0	0
n9-3	True	True	True	28825.00	21995.00	26030.00	14409.00	0	0
neos-1112782	True	True	True	225000000000000.00	225000000000000.00	1065578894123.88	571844066711.00	0	0
neos-1112787	True	True	True	200000000000000.00	200000000000000.00	593525796221.26	564772773667.00	0	0
neos-1171737	False	False	True	inf	inf	-34.00	-195.00	30	30
neos-1367061	True	True	True	31856051.54	31780764.13	31781395.61	31320456.26	0	0
neos-1430701	False	True	True	0.00	-42.00	-18.00	-77.00	12	25
neos-1442119	False	True	True	0.00	-98.00	-54.00	-181.00	13	27
neos-1603965	True	True	True	865504980.43	627172725.19	859650980.43	619244367.66	0	0
neos-2987310-joes	False	True	False	inf	-222862.32	inf	-607702988.30	30	30
neos-3072252-nete	False	True	True	24183360.00	23371673.00	13723328.00	11807698.00	27	27
neos-4290317-perth	False	True	True	inf	3257293.75	3912559.02	3017324.03	1	18
neos-4954672-berkel	False	True	True	18517796.00	9962581.00	8110833.00	2612710.00	2	28
neos-5076235-embley	False	True	True	inf	3150.00	3224.00	2362.00	22	23
neos-5079731-flyers	False	False	True	inf	inf	3650.00	2440.00	25	23
neos-5192052-neckar	False	True	True	-1100000.00	-9030000.00	-11280000.00	-11670000.00	2	6
neos-787933	True	True	True	1764.00	1764.00	1748.00	30.00	0	0
neos-848198	True	True	True	170974.00	72169.00	66620.00	51837.00	0	0
neos-872648	True	True	True	52.57	52.57	48.77	48.61	0	0
neos-873061	True	True	True	152.43	152.43	145.98	113.66	0	0
neos-933638	True	True	True	5074461.10	133369.00	141463.70	276.00	0	0
neos-933966	True	True	True	5074480.10	111392.00	149438.90	318.00	0	0
neos17	True	True	True	0.49	0.36	0.40	0.15	0	0
neos22	False	True	True	inf	939684.38	1188643.75	779715.00	29	29
neos5	True	True	True	38.50	17.00	22.00	15.00	0	0
nexp-150-20-1-5	True	True	True	102.00	89.00	82.00	66.00	0	0
nexp-150-20-8-5	True	True	True	17880.00	4288.00	15864.00	231.00	0	0
nexp-50-20-1-1	False	True	True	inf	68.00	58.00	29.00	8	6

Continued on next page

	granular			objective				iterations	
	root	greedy	random	root	greedy	random	best known	greedy	random
nexp-50-20-4-2	False	True	True	inf	340.00	739.00	71.00	5	9
ns4-pr6	True	True	True	29550.00	29453.00	29452.00	29314.00	0	0
opm2-z10-s4	True	True	True	-1489.00	-3122.00	-2983.00	-33269.00	0	0
opm2-z6-s1	True	True	True	-1076.00	-2174.00	-1510.00	-6202.00	0	0
opm2-z7-s8	True	True	True	-1654.00	-4881.00	-2503.00	-11242.00	0	0
opm2-z8-s0	True	True	True	-2220.00	-5443.00	-3066.00	-15775.00	0	0
osorio-cta	False	True	True	inf	0.04	0.04	0.03	7	7
p200x1188c	True	True	True	20962.00	20962.00	17373.00	15078.00	0	0
p500x2988	True	True	True	72594.00	72561.00	72542.00	71836.00	0	0
p500x2988c	True	True	True	17538.39	17538.00	17538.00	15215.00	0	0
p500x2988d	True	True	True	6.00	6.00	6.00	6.00	0	0
qiu	True	True	True	1805.18	744.15	314.45	-132.87	0	0
queens-30	True	True	True	0.00	-3.00	-15.00	-40.00	0	0
r50x360	True	True	True	2016.00	2016.00	2016.00	1653.00	0	0
rail507	False	True	False	inf	1759.00	inf	174.00	23	30
railway_8_1_0	False	True	True	inf	482.00	486.00	400.00	5	20
ramos3	True	True	True	1095.00	543.00	642.00	192.00	0	0
ran12x21	True	True	True	5550.05	4751.00	4537.00	3664.00	0	0
ran13x13	True	True	True	4439.02	4122.00	3940.00	3252.00	0	0
ran14x18-disj-8	True	True	True	9675.74	5423.00	4758.00	3712.00	0	0
set3-09	True	True	True	1985526.93	1759784.82	1759789.87	176497.15	0	0
set3-10	True	True	True	1992089.60	1756477.36	1756482.40	185179.04	0	0
set3-15	True	True	True	1925867.59	1701104.14	1701109.19	124886.00	0	0
set3-16	True	True	True	1919621.67	1701416.05	1701421.27	134040.41	0	0
set3-20	True	True	True	1916274.13	1712935.20	1712940.25	159462.57	0	0
seymour	True	True	True	743.00	501.00	559.00	423.00	0	0
seymour1	True	True	True	476.57	418.34	438.53	410.76	0	0
sorrell7	True	True	True	0.00	-140.00	-65.00	-196.00	0	0
sorrell8	True	True	True	0.00	-298.00	-171.00	-350.00	0	0
sp150x300d	True	True	True	70.00	69.00	70.00	69.00	0	0
stockholm	True	True	True	156.00	152.00	146.00	125.00	0	0
supportcase39	True	True	True	-1078779.33	-1081369.98	-1082077.66	-1085069.60	0	0
supportcase42	True	True	True	10.45	9.00	9.00	7.76	0	0
ta1-UUM	True	True	True	510951307.90	289145959.41	293656954.80	7518328.20	0	0
tanglegram4	True	True	True	55202.00	24714.00	27548.00	10696.00	0	0
tanglegram6	True	True	True	8856.00	3425.00	4865.00	1224.00	0	0
toll-like	True	True	True	2204.00	729.00	1345.00	610.00	0	0

Continued on next page

	granular			objective				iterations	
	root	greedy	random	root	greedy	random	best known	greedy	random
tr12-30	False	True	True	inf	196614.00	185722.86	130596.00	7	6
usAbbrv-8-25_70	False	True	True	inf	200.00	193.00	120.00	4	21
v150d30-2hopcds	True	True	True	117.00	55.00	68.00	41.00	0	0

TABLE A.5: Instances with some granular node, corresponding objective values and number of feasibility diving iterations

	root node	1 dive	3 dives	5 dives	SCIP	best known
set3-09	1122029.59	1042062.59	1030350.59	938802.21	1759784.82	176497.15
set3-20	1663579.45	1141125.67	863520.06	771482.81	1712935.20	159462.57
neos-983171	50987.00	9431.00	8747.00	8747.00	9272.00	2360.00
b1c1s1	72555.00	71900.43	69071.53	61920.13	69333.52	24544.25
neos-1445743	-3187.00	-12951.00	-15684.00	-15684.00	-11109.00	-17905.00
opm2-z6-s1	-1040.00	-3986.00	-4494.00	-4494.00	-3808.00	-6202.00
set3-10	1679684.07	991124.54	961363.41	750274.15	1756477.36	185179.04
gen-ip054	10928.19	7875.28	7620.17	7207.21	7235.30	6840.97
neos-1367061	31780762.62	31780762.62	31780762.62	31780762.62	33300456.26	31320456.26
bg512142	302107.00	275544.50	273390.00	273390.00	6301928.50	184202.75
neos-1445765	-2468.00	-13730.00	-13823.00	-14019.00	-11513.00	-17783.00
gsvm2rl9	16382.76	13611.89	13611.89	13611.89	31802.40	7438.18
opm2-z10-s4	-1489.00	-18032.00	-22681.00	-22871.00	-20344.00	-33269.00
sorrell7	-45.00	-153.00	-160.00	-160.00	-152.00	-196.00
b2c1s1	73676.52	69221.52	68701.52	68701.52	71120.52	25687.90
opm2-z8-s0	-2220.00	-10005.00	-11328.00	-11328.00	-9833.00	-15775.00
sorrell8	-168.00	-324.00	-324.00	-329.00	-301.00	-350.00
gsvm2rl12	34.35	34.35	34.35	34.35	50.00	22.12
seymour1	437.31	425.43	421.72	421.45	438.08	410.76
set3-16	1209249.73	913569.70	877326.71	661927.58	1701416.05	134040.41
mushroom-best	3613.90	2285.86	2072.90	2063.90	4208.00	0.06
gr4x6	332.35	236.50	218.60	218.60	219.35	202.35
qiu	3173.59	1919.21	603.68	603.68	1805.18	-132.87
gsvm2rl11	42635.34	41080.59	39792.84	39792.84	83555.95	18121.64
gsvm2rl5	10.00	10.00	9.07	8.81	10.00	5.42
neos-1112787	20002990000000.00	1095537656401.14	1070535971510.15	590877191280.26	21786753400000.00	564772773667.00
dg012142	25623489.00	14981424.74	14373382.60	14373382.60	33433439.00	2300867.00
opm2-z7-s8	-1654.00	-7243.00	-7638.00	-7681.00	-5599.00	-11242.00
haprp	4604106.31	3813762.73	3792385.43	3792385.43	4557402.61	3673280.68

TABLE A.6: A comparison of objective values for instances where feasible rounding approaches yield best incumbent solutions

Bibliography

[1] T. Achterberg and T. Berthold. "Improving the feasibility pump". In: *Discrete Optimization* 4.1 (2007), pp. 77–86. DOI: 10.1016/j.disopt.2006.10.004.

[2] T. Achterberg, R. E. Bixby, Z. Gu, E. Rothberg, and D. Weninger. "Presolve Reductions in Mixed Integer Programming". In: *INFORMS Journal on Computing* 32.2 (2019), pp. 473–506. DOI: 10.1287/ijoc.2018.0857.

[3] T. Achterberg, T. Koch, and A. Martin. "MIPLIB 2003". In: *Operations Research Letters* 34.4 (2006), pp. 361–372. DOI: 10.1016/j.orl.2005.07.009.

[4] A. Ataei. *Fast Reduced Row Echelon Form*. MATLAB Central File Exchange. URL: https://www.mathworks.com/matlabcentral/fileexchange/21583-fast-reduced-row-echelon-form (visited on 09/02/2020).

[5] D. Azé. "A survey on error bounds for lower semicontinuous functions". In: *ESAIM: Proceedings* 13 (2003), pp. 1–17. DOI: 10.1051/proc:2003004.

[6] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. "Mixed-integer nonlinear optimization". In: *Acta Numerica* 22 (2013), pp. 1–131. DOI: 10.1017/S0962492913000032.

[7] T. Berthold. *Heuristic algorithms in global MINLP solvers: Zugl.: Berlin, Techn. Univ., Diss., 2014*. 1. Aufl. München: Verl. Dr. Hut, 2015.

[8] T. Berthold. "RENS". In: *Mathematical Programming Computation* 6.1 (2014), pp. 33–54. DOI: 10.1007/s12532-013-0060-9.

[9] T. Berthold and A. M. Gleixner. "Undercover: a primal MINLP heuristic exploring a largest sub-MIP". In: *Mathematical Programming* 144.1-2 (2014), pp. 315–346. DOI: 10.1007/s10107-013-0635-2.

[10] L. T. Biegler and I. E. Grossmann. "Retrospective on optimization". In: *Computers & Chemical Engineering* 28.8 (2004), pp. 1169–1192. DOI: 10.1016/j.compchemeng.2003.11.003.

[11] R. E. Bixby. "A brief history of linear and mixed-integer programming computation". In: *Documenta Mathematica* (2012), pp. 107–121.

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

[13] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. "An algorithmic framework for convex mixed integer nonlinear programs". In: *Discrete Optimization* 5.2 (2008), pp. 186–204. DOI: 10.1016/j.disopt.2006.10.011.

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

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

[16] P. Bonami and J. Lee. *Bonmin users' manual: Version 1.8*. 2013. URL: https://projects.coin-or.org/Bonmin/browser/stable/1.8/Bonmin/doc/BONMIN_UsersManual.pdf?format=raw (visited on 03/17/2020).

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

[18] G. C. Calafiore and L. El Ghaoui, eds. *Optimization Models*. Cambridge University Press, 2018. DOI: 10.1017/CBO9781107279667.

[19] J. Carlson, A. M. Jaffe, and A. Wiles. *The Millennium Prize problems*. Cambridge, Mass. and Providence, RI: Clay Mathematics Inst and American Mathematical Soc, 2006.

[20] V. Chvatal. "A Greedy Heuristic for the Set-Covering Problem". In: *Mathematics of Operations Research* 4.3 (1979), pp. 233–235. DOI: 10.1287/moor.4.3.233.

[21] *COIN-OR*. URL: https://www.coin-or.org/ (visited on 03/16/2020).

[22] M. Conforti, G. Cornuéjols, and G. Zambelli. *Integer Programming*. Vol. 271. Cham: Springer International Publishing, 2014. DOI: 10.1007/978-3-319-11008-0.

[23] W. Cook. "Computing in Combinatorial Optimization". In: *Computing and Software Science*. Ed. by B. Steffen and G. Woeginger. Vol. 10000. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 27–47. DOI: 10.1007/978-3-319-91908-9_3.

[24] G. Dantzig, D. R. Fulkerson, and S. Johnson. "Solution of a Large-Scale Traveling-Salesman Problem". In: *Journal of the Operations Research Society of America* 2.4 (1954), pp. 393–410. DOI: 10.1287/opre.2.4.393.

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

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

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

[28] R. Fletcher and S. Leyffer. "Solving mixed integer nonlinear programs by outer approximation". In: *Mathematical Programming* 66.1-3 (1994), pp. 327–349. DOI: 10.1007/BF01581153.

[29] M. Fukushima and J.-S. Pang. "Some Feasibility Issues in Mathematical Programs with Equilibrium Constraints". In: *SIAM Journal on Optimization* 8.3 (1998), pp. 673–681. DOI: 10.1137/S105262349731577X.

[30] G. Gamrath, D. Anderson, K. Bestuzheva, W.-K. Chen, L. Eifler, M. Gasse, P. Gemander, A. Gleixner, L. Gottwald, K. Halbig, G. Hendel, C. Hojny, T. Koch, P. Le Bodic, S. Maher, F. Matter, M. Miltenberger, E. Mühmer, B. Müller, M. Pfetsch, F. Schlösser, F. Serrano, Y. Shinano, C. Tawfik, S. Vigerske, F. Wegscheider, D. Weninger, and J. Witzig. *The SCIP Optimization Suite 7.0*. ZIB, Takustr. 7, 14195 Berlin, 2020.

[31] A. Gleixner, G. Hendel, G. Gamrath, T. Achterberg, M. Bastubbe, T. Berthold, P. Christophel, K. Jarck, T. Koch, J. Linderoth, M. Lübbecke, H. D. Mittelmann, D. Ozyurt, T. K. Ralphs, D. Salvagnin, and Y. Shinano. "MIPLIB 2017: data-driven compilation of the 6th mixed-integer programming library". In: *Mathematical Programming Computation* 34.4 (2021), p. 361. DOI: 10.1007/s12532-020-00194-3.

[32] R. E. Gomory. "Outline of an Algorithm for Integer Solutions to Linear Programs and An Algorithm for the Mixed Integer Problem". In: *50 Years of Integer Programming 1958-2008*. Ed. by M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 77–103. DOI: 10.1007/978-3-540-68279-0_4.

[33] F. J. Grunewald and D. Segal. "How to solve a quadratic equation in integers". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 89.1 (1981), pp. 1–5. DOI: 10.1017/S030500410005787X.

[34] L. L. Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2020. URL: https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/8.1/refman.pdf (visited on 05/04/2021).

[35] E. R. Hansen and G. W. Walster. *Global optimization using interval analysis*. 2. ed., rev. and expanded. Vol. 264. Monographs and textbooks in pure and applied mathematics. New York: Dekker, 2004. DOI: 10.1201/9780203026922.

[36] W. E. Hart, J.-P. Watson, and D. L. Woodruff. "Pyomo: modeling and solving mathematical programs in Python". In: *Mathematical Programming Computation* 3.3 (2011), pp. 219–260. DOI: 10.1007/s12532-011-0026-8.

[37] C. Hermite. "Sur l'introduction des variables continues dans la théorie des nombres." In: *Journal für die reine und angewandte Mathematik* 41 (1851), pp. 191–216.

[38] H. Hijazi, P. Bonami, and A. Ouorou. "An Outer-Inner Approximation for Separable Mixed-Integer Nonlinear Programs". In: *INFORMS Journal on Computing* 26.1 (2014), pp. 31–44. DOI: 10.1287/ijoc.1120.0545.

[39] F. S. Hillier. "Efficient Heuristic Procedures for Integer Linear Programming with an Interior". In: *Operations Research* 17.4 (1969), pp. 600–637. DOI: 10.1287/opre.17.4.600.

[40] A. J. Hoffman. "On Approximate Solutions of Systems of Linear Inequalities". In: *Journal of Research of the National Bureau of Standards* 1952.49 (1952), pp. 263–265.

[41] J. Hughes. *Extract independent sub-systems from a bigger linear Eq. System*. Mathematics Stack Exchange. URL: https://math.stackexchange.com/q/2340486 (visited on 08/26/2020).

[42] D. S. Johnson. "Approximation algorithms for combinatorial problems". In: *Journal of Computer and System Sciences* 9.3 (1974), pp. 256–278. DOI: 10.1016/S0022-0000(74)80044-9.

[43] R. Kannan and A. Bachem. "Polynomial Algorithms for Computing the Smith and Hermite Normal Forms of an Integer Matrix". In: *SIAM Journal on Computing* 8.4 (1979), pp. 499–507. DOI: 10.1137/0208040.

[44] J. E. Kelley JR. "The cutting-plane method for solving convex programs". In: *Journal of the society for Industrial and Applied Mathematics* 8.4 (1960), pp. 703–712.

[45] D. Klatte and G. Thiere. "Error bounds for solutions of linear equations and inequalities". In: *ZOR Mathematical Methods of Operations Research* 41.2 (1995), pp. 191–214. DOI: 10.1007/BF01432655.

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

[47] M. Köppe and R. Weismantel. "Cutting planes from a mixed integer Farkas lemma". In: *Operations Research Letters* 32.3 (2004), pp. 207–211. DOI: 10.1016/j.orl.2003.08.003.

[48] J. Kronqvist, A. Lundell, and T. Westerlund. "The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming". In: *Journal of Global Optimization* 64.2 (2016), pp. 249–272. DOI: 10.1007/s10898-015-0322-3.

[49] W. Li. "Sharp Lipschitz Constants for Basic Optimal Solutions and Basic Feasible Solutions of Linear Programs". In: *SIAM Journal on Control and Optimization* 32.1 (1994), pp. 140–153. DOI: 10.1137/S036301299222723X.

[50] W. Li. "The sharp Lipschitz constants for feasible and optimal solutions of a perturbed linear program". In: *Linear Algebra and its Applications* 187 (1993), pp. 15–40. DOI: 10.1016/0024-3795(93)90125-8.

[51] M. Lubin, E. Yamangil, R. Bent, and J. P. Vielma. "Polyhedral approximation in mixed-integer convex optimization". In: *Mathematical Programming* 172.1-2 (2018), pp. 139–168. DOI: 10.1007/s10107-017-1191-y.

[52] S. Maher, M. Miltenberger, J. P. Pedroso, D. Rehfeldt, R. Schwarz, and F. Serrano. "PySCIPOpt: Mathematical Programming in Python with the SCIP Optimization Suite". In: *Mathematical Software - ICMS 2016*. Ed. by G.-M. Greuel, T. Koch, P. Paule, and A. Sommese. Lecture Notes in Computer Science. Cham and Heidelberg: Springer, 2016, pp. 301–307. DOI: 10.1007/978-3-319-42432-3_37.

[53] O. L. Mangasarian and T.-H. Shiau. "Lipschitz Continuity of Solutions of Linear Inequalities, Programs and Complementarity Problems". In: *SIAM Journal on Control and Optimization* 25.3 (1987), pp. 583–595. DOI: 10.1137/0325033.

[54] *MINLPLib*. URL: http://www.minlplib.org/instances.html (visited on 03/16/2020).

[55] *MIPLIB 2017 website*. URL: https://miplib.zib.de/tag_collection.html (visited on 03/02/2021).

[56] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience series in discrete mathematics and optimization. New York: Wiley, 2010. DOI: 10.1002/9781118627372.

[57] A. Neumaier. *Interval methods for systems of equations*. Vol. volume 37. Encyclopedia of mathematics and its applications. Cambridge: Cambridge University Press, 1990. DOI: 10.1017/CBO9780511526473.

[58] C. Neumann. "A Feasible Rounding Approach for Mixed-Integer Optimization Problems". Unpublished Master Thesis, Karlsruhe, 2017.

[59] C. Neumann and O. Stein. "Feasible rounding approaches for equality constrained mixed-integer optimization problems". In: *Optimization Online, Preprint ID 2020-10-8045* (2020).

[60] C. Neumann and O. Stein. "Generating feasible points for mixed-integer convex optimization problems by inner parallel cuts". In: *Optimization Online, Preprint ID 2018-11-6947* (2020).

[61] C. Neumann, O. Stein, and N. Sudermann-Merx. "A feasible rounding approach for mixed-integer optimization problems". In: *Computational Optimization and Applications* 72.2 (2019), pp. 309–337. DOI: 10.1007/s10589-018-0042-y.

[62] C. Neumann, O. Stein, and N. Sudermann-Merx. "Bounds on the Objective Value of Feasible Roundings". In: *Vietnam Journal of Mathematics* 48.2 (2020), pp. 299–313. DOI: 10.1007/s10013-020-00393-4.

[63] C. Neumann, O. Stein, and N. Sudermann-Merx. "Granularity in Nonlinear Mixed-Integer Optimization". In: *Journal of Optimization Theory and Applications* 184.2 (2020), pp. 433–465. DOI: 10.1007/s10957-019-01591-y.

[64] J.-S. Pang. "Error bounds in mathematical programming". In: *Mathematical Programming* 79.1-3 (1997), pp. 299–332. DOI: 10.1007/BF02614322.

[65] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Books on Computer Science. Newburyport: Dover Publications, 2013.

[66] S. M. Robinson. "An Application of Error Bounds for Convex Programming in a Linear Space". In: *SIAM Journal on Control* 13.2 (1975), pp. 271–273. DOI: 10.1137/0313015.

[67] R. T. Rockafellar. *Convex analysis*. Princeton paperbacks. Princeton, N.J: Princeton University Press, 1997. DOI: 10.2307/j.ctt14bs1ff.

[68] S. Sahni. "Computationally Related Problems". In: *SIAM Journal on Computing* 3.4 (1974), pp. 262–279. DOI: 10.1137/0203021.

[69] A. Schrijver. *Theory of linear and integer programming*. Reprinted. A Wiley-Interscience publication. Chichester: Wiley, 2000.

[70] C. L. Siegel. "Über Die Analytische Theorie Der Quadratischen Formen". In: *The Annals of Mathematics* 36.3 (1935), p. 527. DOI: 10.2307/1968644.

[71] O. Stein. "Error bounds for mixed integer linear optimization problems". In: *Mathematical Programming* 156.1-2 (2016), pp. 101–123. DOI: 10.1007/s10107-015-0872-7.

[72] O. Stein. "Error bounds for mixed integer nonlinear optimization problems". In: *Optimization Letters* 10.6 (2016), pp. 1153–1168. DOI: 10.1007/s11590-016-1011-y.

[73] A. Wächter and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* 106.1 (2006), pp. 25–57. DOI: 10.1007/s10107-004-0559-y.

[74] T. Westerlund and F. Pettersson. "An extended cutting plane method for solving convex MINLP problems". In: *Computers & Chemical Engineering* 19 (1995), pp. 131–136. DOI: 10.1016/0098-1354(95)87027-X.

[75] C. Zualinescu. “Sharp Estimates for Hoffman’s Constant for Systems of Linear Inequalities and Equalities”. In: *SIAM Journal on Optimization* 14.2 (2003), pp. 517–533. DOI: 10.1137/S1052623402403505.

Index

Additive relaxation, 14, 114

Basic variables, 79
Big-M reformulation, 11, 38
Binary variables
  difficulties with, 7
  example with, 14, 94
  in practical applications, 30, 35, 44
Bonmin, comparison with, 40–44, 63–71

Computation of
  connected components, 84
  enlargements, *see* Enlargement(s)
  feasible points, *see* Feasible point(s)
  Lipschitz constants, *see* Lipschitz constants
  pseudo-granularity parameters, 27
Computational results
  for (integer) equality constrained MILPs, 86–92
  for connected components, 88
  for inconsistent instances, 73–74
  for IPS-diving, 107–111
  for MICPs, 62–74
  for MILPs, 29–38, 107–111
  for MIQCQPs, 38–45
  for optimality gap closed by diving, 109
  for plain feasible rounding approaches, 29–45, 107–111
  for reversed inner parallel cuts, 69
  for the connected components, 88
  for the HNF, 87
  for the IPCP, 62–74
  for the prref, 87
  of the elimination procedure, 86–88
Connected components, 83, 84, *see also* Computational results
Convergence of the IPCP, 53–55
Coupled variables, 82–84

Cutting plane method
  extended, 48
  inner parallel, *see also* Inner parallel cuts, 50
  Kelley, 48, 50, 52
Cutting planes, *see also* Inner parallel cuts, 47, 50, 53

Degree of freedom, 101–103, 107
Diving
  heuristic for MILPs, 98–106
  steps for empty inner parallel sets, 100–104
  steps for nonempty inner parallel sets, 98–99

Elementary column operations, 80
Enlargement(s)
  computation of, 30, 38, 45
  effects on Lipschitz constants of, 21
  example of, 14, 16, 18, 19, 21, 22, 25
  for nonlinear constraints, 19
  for polyhedral constraints, 15, 23
  main idea of, 8, 13
  of fixed sets, *see* Fixing variables
  trade off with, 22, 38, 45
Epigraph reformulation, 39, 40, 48, 63
  and granularity, 56
Equality constraints
  elimination of, 78–82
  example for the elimination of, 77
  preservation of the potential for granularity with, 76
  treatment of, 30, 76
Error bounds, 2, 3, 31
  a comparison of, 62
  for the IPCP, a-posteriori, 58
  for the IPCP, a-priori, 59–62

Feasibility problem, 28, 100
  optimal value of, 100
Feasibility test, 27

Feasibility tolerance, 30, 50
of LP solvers, 55
of the IPCP, 55
Feasible point(s)
for (integer) equality constrained MILPs, 92
for granular problems, 30, 64
for non-granular problems, 37
for pseudo-granular problems, 40
quality of the FRA-SOR-generated, 35, 40
quality of the IPCP-generated, 68
rounded and merged, 98
Feasible rounding approach
combined with diving steps, 104
for (integer) equality constrained problems, 85
Shrink-Lift-Optimize-Round, 27
Shrink-Optimize-Round, 27, 28
Fixed inner parallel set(s), *see* Inner parallel set(s)
Fixed relaxed feasible set, 95
Fixing variables
enlargements and, 97
geometrical perspective on, 95–97
order of, 99
selection of indices for, 103, 106–108

Gauss-Jordan elimination, 78
Granularity, 7, 24, 26, 48
a remark on, 14
and Epigraph reformulation, 56
and equality constraints on integer variables, *see* Equality constraint(s)
representable, 14
Greatest common divisor, 15, 19–21, 23, 31, 38, 98
Greedy algorithm, 103, 106, 107
Gurobi, comparison with, 29–37, 86, 89

Hermite normal form (HNF), *see also* Computational results, 80
Hermite normal form theorem, 80
Hoffman constant, 61, 62

Inner parallel cuts
closed-form expression of, 51
objective, 57
reversed, 52, 56, 58, 69
Inner parallel set(s)
closed-form expression of, 7, 8, 49
computation of, 29
enlarged, closed-form expression of, 16
enlargement of, 14
fixed, 95
geometrical definition of, 6
inner approximation of, 8, 9, 21, 22
intersection of, 48, 51
outer approximation of, 52, 53
Integer line search, 36
Integrality conditions, 78
Interval arithmetic, 10, 57
IPOPT, 38

Knapsack problem, *see* Optimization problem

Lipschitz constants
assumption of, 9, 24
computation of, 10, 11, 20, 22, 25–27, 38, 39, 45, 46, 50
dependency on enlargements of the, 21
global information for the computation of, 50, 53
monotonicity of, 21
smaller, 24, 25
Lower bounds via inner parallel cuts, 58, 63, 69

Matlab, 30, 86
MIPLIB
2003, 29, 30
2010, 29, 30, 86
2017, 86
Modeling techniques, 18, 72, 92
Monotonicity
of enlargements, 21
of inner parallel sets, 48
of Lipschitz constants, 21
of the reduction of the feasibility error, 70

Non-binary integer variables, 35
Non-granularity measure, *see also* Feasibility problem
improvement of the, 101
non deterioration of, 100
Nonbasic variables, 79

Objective based problem, 28, 98

example of an, 99
fixed, 98
Optimization problem
binary knapsack, 17, 18, 94
equality constrained mixed-integer linear, 75
mixed-integer (nonconvex) quadratically constrained quadratic, 10–12, 40
mixed-integer convex, 5, 47
mixed-integer linear, 5, 29, 98
mixed-integer nonconvex, 6, 40
mixed-integer nonlinear, 5, 93
reduced, 79–86
with complementarity constraints, 11
Outer approximation, 47, 73

Polynomial, real-valued multivariate, 19
Postprocessing step, 36, 52, 64, 86
Presolving step, 18, 29, 30, 32, 37, 78
Primal heuristic, 6, 35, 45, 68, 106
Pseudo-granular MIQCQP instances, 40
Pseudo-granularity, 22–28, 39, 44, 45, 49
Pyomo, 38, 63, 68

Rounding(s)
definition of, 6, 93
excluded by inner parallel cuts, *see also* Convergence of the IPCP, 53
feasibility of, 7, 8, 23, 25
from an interval, 99
increasing number of, 99
nondecreasing number of, 97
obtainable from fixed inner parallel sets, 95
obtainable from the inner parallel set, 94
Row echelon form
partial, *see also* Computational results, 78
reduced, 78

SCIP, comparison with, 63–71, 109–111
Set covering problem, 103
Sparsity, 39, 92, 114
Sparsity pattern matrix, 82

Unimodular
matrix, 80
column operations, *see* Elementary column operations

Worst case behavior of the IPCP, 73