Min Li

Inexactness in Decomposition Methods for MINLP

Coimbra
2012

Min Li

Inexactness in Decomposition Methods for MINLP

Dissertação apresentada à Faculdade de Ciências e Tecnologia da Universidade de Coimbra, para a obtenção do grau de Doutor em Matemática, na especialidade de Matemática Aplicada.

Coimbra
2012

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Professor Luís Nunes Vicente, for his inspirational supervision, patient guidance, and invaluable encouragement throughout my PhD studies. I greatly admire his comprehensive knowledge of Mathematical Optimization and his attitude towards scientific research. His effort, experience, and perseverance led me step by step to advance in my research, and finally to make this dissertation possible.

I would also like to thank Prof. Zhang Yulin (Univ. of Minho) who pointed me to Prof. António Leal Duarte (Univ. of Coimbra), with whom I studied Matrix Analysis in 2007/08. Moreover, under their recommendations, I succeeded in being accepted in 2008 as a PhD student in the UC/UP PhD Program in Mathematics. I take this opportunity to thank all the people involved in the program, the coordinators, the course instructors, and my classmates.

I deeply thank Prof. Ana Luísa Custódio (New Univ. of Lisbon) for all her support in the development of the MATLAB code. I am also very grateful to Prof. A. Ismael F. Vaz (Univ. of Minho) for his help in AMPL and MATLAB.

I cannot possibly forget Mrs. Rute Andrade for helping me so many times in my personal and academic matters.

I acknowledge the financial support from Fundação para a Ciência e a Tecnologia (PhD scholarship SFRH/BD/33369/2008).

Finally, I would like to thank my parents, my husband, my brother, and my daughters for their unlimited love, understanding, encouragement and unmeasurable support during all these years.

Abstract

In the context of convex mixed integer nonlinear programming (MINLP), we investigate how decomposition methods such as the outer approximation (OA) method and the generalized Benders decomposition (GBD) method are affected when the respective nonlinear programming (NLP) subproblems are solved inexactly.

We assume that the solution procedure chosen for the NLP subproblems is such that one can impose a maximum size for the residuals of the first-order necessary conditions (duality and complementarity). We show that the cuts in the master problems of OA and GBD can be changed to incorporate such inexact residuals, still rendering the properties of equivalence and finiteness in the limit case.

We then present some numerical results to illustrate the behavior of the OA and GBD methods under NLP subproblem inexactness (for the case where the non-negativity of the returned Lagrange multipliers of the NLP subproblems is not subject to inexactness).

A number of other studies are also made in terms of extending what is known from the exact case to the inexact one. For instance, one can also conclude that the constraints of the inexact GBD master problem can be derived from the corresponding ones of the inexact OA master problem and that the study of inexactness can also accommodate the non-negativity of the Lagrange multipliers of the NLP subproblems.

Resumo

No contexto da programação não linear inteira mista (MINLP), investigamos como é que métodos de decomposição, como o método da aproximação exterior (OA) e a decomposição generalizada de Benders (GBD), são afectados pela inexactidão na resolução dos respectivos subproblemas de programação não linear (NLP).

Supomos que o procedimento de resolução dos subproblemas NLP é tal que seja possível impor um tamanho máximo nos resíduos das condições necessárias de primeira ordem (no que diz respeito à dualidade e complementaridade). Mostramos que os cortes em OA e GBD podem ser modificados de forma a incorporar os resíduos inexactos, conseguindo-se ainda satisfazer as propriedades de equivalência e finitude no caso limite.

Apresentamos alguns resultados numéricos que ilustram o desempenho dos métodos OA e GBD sob a inexactidão na resolução dos subproblemas NLP (para o caso em que a não-negatividade dos multiplicadores de Lagrange associados a esta resolução inexacta não está sujeita a inexactidão).

Estudam-se, ainda, outras extensões do que é conhecido no caso exacto para o inexacto. Por exemplo, também concluímos que as restrições do método GBD inexacto podem ser obtidas através das correspondentes restrições do método OA inexacto e que este estudo de inexactidão também acomoda a não-negatividade dos multiplicadores de Lagrange dos subproblemas NLP.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General comments

A mixed integer nonlinear programming (MINLP) problem is an optimization problem involving both continuous and discrete variables and where at least one of the functions defining the objective function or the constraints is nonlinear. MINLP problems appear in a diversity of applications, coming from the Industry or Engineering Sectors as well as from Finance and Management Sciences. They include problems in process systems synthesis [23, 24, 39], process flow sheets [38], portfolio selection [48, 49], batch processing in chemical engineering [54, 59], optimal design of gas or water transmission networks [40], and so on. Moreover, a vast collection of MINLP applications can be found in [33, 34]. Recently, and due perhaps to a renewed practical interest, MINLP has become again a very active research area [3, 5, 7, 11, 12, 13, 14, 15, 18, 19, 21, 22, 25, 26, 37, 50, 58], with the development of new theories and algorithms and of new, sophisticated implementations.

In our thesis, we will adopt the MINLP formulation

$$\mathrm{P}\left\{\begin{array}{lll} \min & f(x,y) & \\ \text{s.t.} & g(x,y) & \leq\; 0, \\ & x \in X \cap \mathbb{Z}^{n_d}, y \in Y, & \end{array}\right. \tag{1.1}$$

where X is a bounded polyhedral subset of $\mathbb{R}^{n_d}$ and Y a polyhedral subset of $\mathbb{R}^{n_c}$. The functions $f : X \times Y \longrightarrow \mathbb{R}$ and $g : X \times Y \longrightarrow \mathbb{R}^m$ are assumed continuously differentiable. We say that the MINLP problem P is convex if both $f(x,y)$ and $g(x,y)$ are convex functions over

$X \times Y$. For a fixed x, the MINLP reduces to a nonlinear programming (NLP) subproblem:

$$\text{NLP}(x) \begin{cases} \min\limits_{y} & f(x,y) \\ \text{s.t.} & g(x,y) \ \leq \ 0, \\ & y \in Y. \end{cases} \tag{1.2}$$

Problem P is one of the most difficult optimization problems. It falls into the class of $\mathcal{NP}$-complete problems, meaning that no polynomial time algorithm is known for P and that, if one existed, it would also be a polynomial time algorithm for any other problem in $\mathcal{NP}$ (see, e.g., Nemhauser and Wolsey [51] and Schrijver [57]).

There are essentially two broad classes of methods for solving MINLP problems: the exact or rigorous methods and the heuristic ones. We will now briefly review them.

1.1.1 Exact or rigorous methods for MINLP

Exact methods terminate when an optimal solution[1] is computed (or approximately computed) or an indication is given saying that the problem is infeasible, provided enough computing time is given. These methods are rigorous in the sense that there is a mathematical guarantee of convergence. The most well-known exact methods proposed for solving MINLPs include branch-and-bound (B&B), outer approximation (OA), generalized Benders decomposition (GBD), the extended cutting plane (ECP) method, and LP / NLP-based branch-and-bound (LP/NLP-B&B).

The B&B method was originally designed as an algorithm for mixed integer linear programming (MILP) problems by Land and Doig [41] in 1960. Branching was modified to how we commonly know it now by Dakin [20], and it was applied to MINLP problems by Gupta and Ravindran [35] (see also [12, 16, 45]). This method starts by solving the continuous NLP relaxation of the original problem (i.e., the related formulation where the discrete variables x are fixed to a point in X). Let $(\tilde{x}, \tilde{y})$ be an optimal solution of this continuous relaxation. If all the discrete variables $\tilde{x}$ take integer values, the search is stopped and $(\tilde{x}, \tilde{y})$ is an optimal solution of the MINLP problem. Otherwise, this relaxed NLP problem provides a lower bound to the optimal value. Then, assume that $\tilde{x}_j$ is a fractional value and 'divide' the relaxed NLP problem into two subproblems by adding the constraint $x_j \leq \lfloor \tilde{x}_j \rfloor$ to one and the constraint $x_j \geq \lfloor \tilde{x}_j \rfloor + 1$ to the other. An enumeration tree is then initiated. When an optimal solution is found for a node subproblem for which the discrete variables x take integer values, such node optimal value provides an upper bound to the original one. Otherwise, the lower bound may be improved. Fathoming of nodes occurs in the following cases: the

[1]By solution or optimal solution of an optimization problem we mean a global or absolute minimizer.

node subproblem is infeasible, or the discrete variables of the solution of this subproblem are integer, or the current lower bound exceeds the current upper bound. The algorithm stops when there is no node to explore. What we have described before is what is called the standard or spatial B&B (see the classification in [17]). There are several variations of B&B for MINLPs, such as branch-and-reduce (see Ryoo and Sahinidis [55]) and α-branch-and-bound (see Androulakis et al. [6]).

Benders [8] developed in the 60s a technique for solving MILP problems, later called Benders decomposition. Geoffrion [31] extended it to MINLP with the help of nonlinear duality theory [30, 31] in 1972, in what has become known as the generalized Benders decomposition. Much later, in 1986, Duran and Grossmann [24] derived a new outer approximation method to solve a particular class of MINLP problems, which has become widely used in practice. Although the authors showed finiteness of the OA algorithm, their theory was restricted to problems where the discrete variables appear linearly and the functions involving the continuous variables are convex. Both OA and GBD are iterative schemes requiring at each iteration the solution of a (feasible or infeasible) NLP subproblem of the form (1.2) and one MILP master problem involving a (finite) number of cuts.

For the class of MINLP problems mentioned in the previous paragraph, Quesada and Grossmann [53] then proved that the cuts in the master problem of OA imply the cuts in the master problem of GBD, showing that the GBD algorithm provides weaker lower bounds (some authors [32, 46, 53, 56] associate this fact with the observation that GBD requires a large number of major iterations to converge). Fletcher and Leyffer [27] generalized the OA method of Duran and Grossmann [24] into a wider class of problems where nonlinearities in the discrete variables are allowed as long as the corresponding functions are convex in these variables. They also introduced a new and simpler proof of finiteness of the OA algorithm. The relationship between OA and GBD was then addressed, again, by Grossmann [32] in this wider context of MINLP problems, showing once more that the lower bound predicted by the relaxed master problem[2] of OA is greater than or equal to the one predicted by the relaxed master problem of GBD (see also Flippo and Rinnooy Kan [28] for the relationship between the two techniques). Recently, Bonami et al. [12] suggested a different OA algorithm using linearizations of both the objective function and the constraints, independently of being taken at the feasible or infeasible NLP subproblem, to build the MILP master problem. This technique is, in fact, different from the traditional OA (see [27]), where the cuts in the master MILP problems do not involve linearizations of the objective function in the infeasible case.

Westerlund and Pettersson [61] generalized the cutting plane method [36] from convex

[2]By relaxed master problem, in this thesis, we mean a problem with fewer constraints than needed to state the equivalence between the original problem and the master one.

NLP to convex MINLP, in what is known as the extended cutting plane method (see also [62, 63]). While OA and GBD alternate between the solution of MILP and NLP subproblems, the ECP method relies only on the solution of MILP problems.

The LP/NLP-B&B method proposed by Quesada and Grossmann [53] is an extension of both B&B and OA for convex MINLP (see also [3]). It solves an alternating sequence of NLP subproblems and MILP master problems, but avoids solving completely the MILP master problems (as in the case of the OA method) by interrupting the MILP tree search whenever a pair $(\tilde{x}, \tilde{y})$ is found such that $\tilde{x} \in X \cap \mathbb{Z}^{n_d}$ and $\tilde{y}$ solves the NLP subproblem NLP$(\tilde{x})$ (see (1.2)). Because the NLP subproblems are not necessarily solved at each node, LP/NLP-B&B cannot also be labeled as a B&B.

1.1.2 Heuristic methods for MINLP

Heuristic methods do not provide a guarantee of optimality at termination, and thus the incumbent or best point found so far is not guaranteed to be an optimal solution. Any of the decomposition methods mentioned above (OA or GBD) applied to a problem which does not satisfy for instance a convexity assumption becomes a heuristic method. There are several families of heuristics that have been successfully used for MILP problems: diving heuristics [10], the feasibility pump technique (see [15, 25, 26]), and the relaxation induced neighborhood search (see [22]). Later, some of these heuristics were extended to MINLP problems by Bonami and Gonçalves [14]. We will only review here one of these heuristic methods, the feasibility pump (FP). FP was introduced by Fischetti et al. [25] to find a good feasible candidate point for MILPs (in other words, a point $(\tilde{x}, \tilde{y}) \in (X \cap \mathbb{Z}^{n_d}) \times Y$ satisfying $g(\tilde{x}, \tilde{y}) \leq 0$ and possibly not having a very high value $f(\tilde{x}, \tilde{y})$, where now f and g are affine functions). It can be viewed as a clever way to round a sequence points verifying all the constraints of the original MILP, but possibly fractional in the discrete variables. Given a point $(\hat{x}, \hat{y})$ in these conditions, it is solved the linear program $\min\{\|x - [\hat{x}]\|_1 : g(x, y) \leq 0, (x, y) \in X \times Y\}$, where the i-th component $[\hat{x}]_i$ represents $\hat{x}_i$ rounded to the nearest integer, with the aim of rounding $\hat{x}$, and if its solution is not integer in the discrete variables, the process is repeated. It was successfully applied to MILPs with binary discrete variables, but not so well to instances with general discrete variables. Bertacco et al. [9] improved FP in this more general scenario by applying a restarting scheme. Then, Achterberg and Berthold [4] proposed a slight modification of the algorithm in order to find a better MILP feasible candidate point, i.e., some other $(\tilde{x}', \tilde{y}') \in (X \cap \mathbb{Z}^{n_d}) \times Y$ still satisfying $g(\tilde{x}', \tilde{y}') \leq 0$, but with a much lower value $f(\tilde{x}', \tilde{y}')$. Later, in 2009, Bonami et al. [15] showed that FP can be adapted to MINLP problems.

1.2 Summary of our work

In the above mentioned OA and GBD approaches (but also in the FP heuristic method), the NLP subproblems are solved exactly, at least such property is assumed in the derivation of the theoretical properties, such as the equivalence between original and master problems and the finite termination of the corresponding algorithms. In this thesis we investigate the effect of NLP subproblem inexactness in these techniques (see also our paper accepted for publication [47]). For OA and GBD, we show how the cuts in the master problems can be changed to incorporate the inexact residuals of the first-order necessary conditions of the NLP subproblems, in a way that still renders the equivalence and finiteness properties, as long as the size of these residuals allow inferring the cuts from the convexity properties. Such residuals refer to the right hand sides of the duality and complementarity equations of the first-order necessary conditions, which are no longer considered zero but of controllable size. The first-order necessary conditions of the NLP subproblems can also be satisfied inexactly due to the wrong sign of some of the Lagrange multipliers (being negative, or negative by little, instead of non-negative). We extend our inexact OA and GBD methods to the case where the negative part of these Lagrange multipliers is of controllable size.

We organize the thesis in the following way. In Chapter 2, we describe the exact methods OA, GBD, and ECP and their main convergence properties. In Chapter 3, we extend OA for the inexact solution of the NLP subproblems, rederiving the corresponding background theory and main algorithm. In Chapter 4 we proceed similarly for GBD, also discussing the relationship between the inexact forms of OA and GBD. Chapter 5 describes a set of preliminary numerical experiments, reported to better understand some of the theoretical features encountered in our study of inexactness in MINLP. In Chapter 6, we extend our inexact OA and GBD approaches to the case where the Lagrange multipliers are (slightly) negative. We end the thesis in Chapter 7 with some concluding remarks and prospects of future work.

Chapter 2

Decomposition methods for MINLP

In this chapter, we review three well-known decomposition methods for solving convex MINLP problems: outer approximation (OA), generalized Benders decomposition (GBD), and the extended cutting plane (ECP) method. We start, in Section 2.1, by presenting and discussing NLP formulations of subproblems related to (1.1) in Section 1.1, including their first-order necessary conditions and covering feasible and infeasible cases. Then OA is introduced in Section 2.2, GBD is described in Section 2.3, and the ECP method is presented in Section 2.4. The algorithms OA and GBD are similar, in the sense that both build iteratively a mixed integer linear programming (MILP) master problem by adding cuts taken at the solution of NLP subproblems. The ECP method, however, does not require the solution of NLP subproblems.

2.1 NLP subproblems and equivalent formulations

We assume that problem P defined in (1.1) (see Section 1.1) is convex. Let x^j be any element of $X \cap \mathbb{Z}^{n_d}$. Consider, then, the (convex) subproblem

$$\mathrm{NLP}(x^j) \left\{ \begin{array}{lrl} \min\limits_{y} & f(x^j, y) & \\ \text{s.t.} & g(x^j, y) & \leq 0, \\ & y \in Y, & \end{array} \right.$$

and suppose it is feasible, in the sense that there exists $y \in Y$ such that $g(x^j, y) \leq 0$. In this case, y^j will represent an optimal solution of $\mathrm{NLP}(x^j)$. One can then see that $\mathrm{NLP}(x^j)$ yields the upper bound $f(x^j, y^j)$ to the optimal value of problem P. Given an x^k in $X \cap \mathbb{Z}^{n_d}$ such that $\mathrm{NLP}(x^k)$ is infeasible, we will introduce two kinds of feasibility subproblems ('feasibility' in the sense of enforcing feasibility) and denote their optimal solutions by y^k. The first subproblem is

$$\min_{y\in Y} \max_{i\in\{1,\ldots,m\}} g_i^+(x^k, y),$$

where $t^+ = \max\{t, 0\}$, or equivalently

$$\mathrm{NLPF}_\infty(x^k) \begin{cases} \min\limits_{y,u} & u \\ \text{s.t.} & g_i(x^k, y) \ \le \ u, \ i = 1, \ldots, m, \\ & y \in Y, u \in \mathbb{R}, \end{cases}$$

(see Fletcher and Leyffer [27], Grossmann [32], and Quesada and Grossmann [53]). The second feasibility problem is

$$\min_{y\in Y} \sum_{i=1}^{m} g_i^+(x^k, y),$$

or equivalently

$$\mathrm{NLPF}_1(x^k) \begin{cases} \min\limits_{y,u} & \sum_{i=1}^m u_i \\ \text{s.t.} & g(x^k, y) \ \le \ u, \\ & u \ \ge \ 0, \\ & y \in Y, u \in \mathbb{R}^m, \end{cases}$$

(see Fletcher and Leyffer [27] and Bonami et al. [12]). Note that u is a scalar in $\mathrm{NLPF}_\infty(x^k)$ and a vector in $\mathrm{NLPF}_1(x^k)$.

As we have seen above, $\mathrm{NLPF}_\infty(x^k)$ and $\mathrm{NLPF}_1(x^k)$ can be interpreted as the minimization of the ℓ_∞-norm and ℓ_1-norm, respectively, of the measure of infeasibility of the corresponding $\mathrm{NLP}(x^k)$ subproblem. So, one can easily see that these two subproblems are feasible when $\mathrm{NLP}(x^k)$ is infeasible. Although the ℓ_∞-norm and ℓ_1-norm are equivalent in finite dimensional spaces, the optimal solutions of these two NLP subproblems might not be the same in their y component, as the following example indicates.

Example 2.1.1. *Consider the following* MINLP *problem*

$$\begin{cases} \min & x^2 + y \\ \text{s.t.} & -x + y^2 - y + 1 \ \le \ 0, \\ & -x + 2y \ \le \ 0, \\ & x \in \{0, 1\}, y \in [0, 1]. \end{cases}$$

One can see obviously that the subproblem NLP(0) is infeasible. The optimal solutions of the two feasibility subproblems $\mathrm{NLPF}_\infty(0)$ and $\mathrm{NLPF}_1(0)$, with respect to y, are $\frac{3-\sqrt{5}}{2}$ and 0, respectively, and thus different.

When we know *a priori* which constraints are identified as infeasible in $\mathrm{NLPF}_\infty(x^k)$ and $\mathrm{NLPF}_1(x^k)$, one can fit these two subproblems in a more general framework by using the non-negative weights $\omega_1^k, \ldots, \omega_m^k$, and by posing the feasibility NLP subproblem

$$\mathrm{NLPF}_\omega(x^k) \begin{cases} \min & \sum_{j \notin I^k} \omega_j^k g_j^+(x^k, y) \\ \text{s.t.} & g_i(x^k, y) \ \leq \ 0, \ i \in I^k, \\ & y \in Y, \end{cases}$$

where, again, $t^+ = \max\{t, 0\}$, and I^k are the constraints identified as feasible. One can see that $\mathrm{NLPF}_\omega(x^k)$ is the same as $\mathrm{NLPF}_1(x^k)$ when $\omega_j^k = 1$, $j \in \{1, \ldots, m\} \backslash I^k$ and I^k is the set of indices i leading to $u_i^* = 0$ in the solution of $\mathrm{NLPF}_1(x^k)$. Analogously, $\mathrm{NLPF}_\omega(x^k)$ is the same as $\mathrm{NLPF}_\infty(x^k)$ when $\omega_j^k = 1$ if $g_j(x^k, y)$ attains the maximum value u^* in the solution of $\mathrm{NLPF}_\infty(x^k)$ and $\omega_j^k = 0$ otherwise.

As we will see in the next lemma, the linear constraints obtained by linearizing g_i, $i = 1, \ldots, m$, around the optimal solutions of the feasibility NLP subproblems will be violated at the solution of these subproblems. This fact will later allow the master MILP in OA algorithms to exclude previous pairs (x^k, y^k) as its optimal solutions and therefore generate finite convergence for these methods. We state such a result for the feasibility NLP subproblems in the more general format $\mathrm{NLPF}_\omega(x^k)$, as done in [27].

Lemma 2.1.1. *For $x^k \in X \cap \mathbb{Z}^{n_d}$, if $\mathrm{NLP}(x^k)$ is infeasible, so that y^k solves $\mathrm{NLPF}_\omega(x^k)$ with*

$$\sum_{j \notin I^k} \omega_j^k g_j^+(x^k, y^k) \ > \ 0,$$

for some $I^k \subset \{1, \ldots, m\}$, then $x = x^k$ violates the constraints

$$\nabla g(x^k, y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + g(x^k, y^k) \ \leq \ 0,$$

for all $y \in Y$.

Then because $\mathrm{NLPF}_\infty(x^k)$ and $\mathrm{NLPF}_1(x^k)$ are two special cases of $\mathrm{NLPF}_\omega(x^k)$, this lemma applies to these two feasibility NLP subproblems as well. The proof of Lemma 2.1.1 is simple and based on the first-order necessary conditions for $\mathrm{NLPF}_\omega(x^k)$. In Chapter 3 (at the beginning of the proof of Theorem 3.4.1), we give a more general argument taking the case of $\mathrm{NLPF}_\infty(x^k)$, where these conditions are satisfied inexactly.

Now we need to enumerate the assignments of the discrete variables leading to feasible and infeasible NLP subproblems:

$$T^e \;=\; \{j : x^j \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^j) \text{ is feasible, and } y^j \text{ solves} \,\mathrm{NLP}(x^j)\} \tag{2.1}$$

and

$$S^e_\infty \;=\; \{k : x^k \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^k) \text{ is infeasible, and } y^k \text{ solves} \,\mathrm{NLPF}_\infty(x^k)\}. \tag{2.2}$$

Correspondingly, we also have

$$S^e_1 \;=\; \{k : x^k \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^k) \text{ is infeasible, and } y^k \text{ solves} \,\mathrm{NLPF}_1(x^k)\}$$

and

$$S^e_\omega \;=\; \{k : x^k \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^k) \text{ is infeasible, and } y^k \text{ solves} \,\mathrm{NLPF}_\omega(x^k)\}.$$

For each of these discrete assignments, one needs also an algebraic characterization of the optimal solutions of the corresponding NLP subproblems. For such a purpose, and since the subproblems are convex, it suffices to work with first-order necessary conditions. For a matter of simplicity, and without loss of generality, we suppose that the constraints $y \in Y$ are included in the constraints $g(x^j, y) \le 0$ in problem $\mathrm{NLP}(x^j)$, in the constraints $g_i(x^k, y) \le u$, $i = 1, \ldots, m$, in problem $\mathrm{NLPF}_\infty(x^k)$, and in the constraints $g(x^k, y) \le u$ in problem $\mathrm{NLPF}_1(x^k)$. Let us then assume that the optimal solutions of the NLP subproblems satisfy the first-order necessary conditions also known as first-order necessary Karush-Kuhn-Tucker conditions (or just KKT conditions in short). More particularly, in the case of $\mathrm{NLP}(x^j)$, we assume the existence of $\lambda^j \in \mathbb{R}^m_+$ such that

$$\begin{aligned} \nabla_y f(x^j, y^j) + \sum_{i=1}^m \lambda_i^j \nabla_y g_i(x^j, y^j) \;&=\; 0, \\ \lambda_i^j g_i(x^j, y^j) \;&=\; 0, \quad i = 1, \ldots, m. \end{aligned}$$

When $\mathrm{NLP}(x^k)$ is infeasible, we assume, for $\mathrm{NLPF}_\infty(x^k)$, the existence of $\mu^k \in \mathbb{R}^m_+$ such that

$$\begin{aligned} \sum_{i=1}^m \mu_i^k \nabla_y g_i(x^k, y^k) \;&=\; 0, \\ 1 - \sum_{i=1}^m \mu_i^k \;&=\; 0, \\ \mu_i^k (g_i(x^k, y^k) - u^k) \;&=\; 0, \quad i = 1, \ldots, m. \end{aligned}$$

As it is well known, these KKT conditions are necessary for local optimality under the presence of a constraint qualification (see, e.g., [52]).

In the sequel, we use the superscripts l, p, and q to denote the iteration count, superscript j to index the feasible NLP subproblems defined above, and k to indicate infeasible subproblems. The following notation is adopted to distinguish between function values and functions. $f^l = f(x^l, y^l)$ denotes the value of f evaluated at the point (x^l, y^l), similarly, $\nabla f^l = \nabla f(x^l, y^l)$ is the value of the gradient of f at the point (x^l, y^l), $\nabla_x f^l = \nabla_x f(x^l, y^l)$ is the value of the gradient of f with respect to x at the point (x^l, y^l), and $\nabla_y f^l = \nabla_y f(x^l, y^l)$ is the value of the gradient of f with respect to y at the point (x^l, y^l). Moreover, the same conventions apply for all other functions in our thesis.

2.2 Outer approximation

In this section, we plan to compare two OA algorithms for solving a wide class of MINLP problems where nonlinearities in the discrete variables are allowed as long as the corresponding functions are convex in these variables. The OA algorithms were detailed by Fletcher and Leyffer [27] and Bonami et al. [12]. They are both based on the standard outer approximation decomposition method for solving MINLPs of Duran and Grossmann [24] (and later improved in [23, 60]), where the MILP master problem is obtained by linearizing the objective function and the constraints only at feasible points of the original problem P.

On the one hand, Fletcher and Leyffer [27] showed that the argument given by Duran and Grossmann in [24] was incomplete by providing one simple example where it is indeed necessary to include information coming from the infeasible points of P. Then, they defined an MILP master problem equivalent to the original MINLP by adding linearizations of the constraints g also at the optimal solutions of the feasibility NLP subproblems $\mathrm{NLPF}_\omega(x^k)$. The master MILP is posed in the variables α, x, and y and its cuts or constraints are defined by linearizations of f at (x^j, y^j), $j \in T^e$ and by linearizations of g at both (x^j, y^j), $j \in T^e$, and (x^k, y^k), $k \in S^e_\infty$, and is stated below,

$$
\mathrm{P}^{\mathrm{OA}}_{FL}
\begin{cases}
\min & \alpha \\
\text{s.t.} & \nabla f(x^j,y^j)^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + f(x^j,y^j) \;\leq\; \alpha, \\
& \nabla g(x^j,y^j)^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + g(x^j,y^j) \;\leq\; 0, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall j \in T^e \\
& \nabla g(x^k,y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + g(x^k,y^k) \;\leq\; 0, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall k \in S^e_\infty \\
& x \in X \cap \mathbb{Z}^{n_d}, y \in Y, \alpha \in \mathbb{R}.
\end{cases}
$$

We remark that the authors in [27] used S^e_ω rather than S^e_∞, but we use the latter one above since it is the one adopted in our thesis and, moreover, we know that $\mathrm{NLPF}_\infty(x^k)$ can be seen as a special case of $\mathrm{NLPF}_\omega(x^k)$.

On the other hand, by also including cuts or constraints involving linearizations of the objective function at the points (x^k, y^k), $k \in S^e_\infty$, Bonami et al. [12] built an MILP master problem different from the above one as follows:

$$
\mathrm{P}^{\mathrm{OA}}_{PB}
\begin{cases}
\min & \alpha \\
\text{s.t.} & \nabla f(x^j,y^j)^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + f(x^j,y^j) \;\leq\; \alpha, \\
& \nabla g(x^j,y^j)^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + g(x^j,y^j) \;\leq\; 0, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall j \in T^e \\
& \nabla f(x^k,y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + f(x^k,y^k) \;\leq\; \alpha, \\
& \nabla g(x^k,y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + g(x^k,y^k) \;\leq\; 0, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall k \in S^e_\infty \\
& x \in X \cap \mathbb{Z}^{n_d}, y \in Y, \alpha \in \mathbb{R}.
\end{cases}
$$

We should also note that, strictly speaking, the authors in [12] used instead the index set S^e_1. Despite the fact that in Section 2.1 we showed that the feasibility NLP subproblems $\mathrm{NLPF}_\infty(x^k)$ and $\mathrm{NLPF}_1(x^k)$ may not have the same optimal solution, we have seen (in Lemma 2.1.1) that there is no difference in using one or the other for establishing the

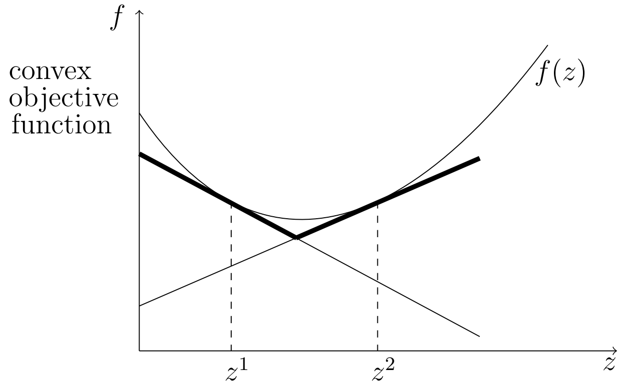

Figure 2.1: Underestimating a function by linear representations.

equivalence between the convex MINLP problem P and the MILP master problem, under the satisfaction of the KKT conditions at every solution of the NLP subproblems (as we will see in Theorem 2.2.1).

Both OA methods are based on the equivalence between the corresponding master MILP problem and the original MINLP problem P. By minimizing α in $\mathrm{P}_{FL}^{\mathrm{OA}}$ we are minimizing the continuous piecewise linear function

$$\max_{j \in T^e} \nabla f(x^j, y^j)^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + f(x^j, y^j) \tag{2.3}$$

over the remaining constraints not involving f (in the case of $\mathrm{P}_{PB}^{\mathrm{OA}}$ the max in (2.3) is taken with respect to $T^e \cup S_\infty^e$). So one is essentially replacing f by an underestimation of the form represented in Figure 2.1. On the other hand, the constraints

$$\left\{ \begin{array}{llll} \nabla g(x^j, y^j)^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + g(x^j, y^j) & \leq & 0, & j \in T^e \\ \nabla g(x^k, y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + g(x^k, y^k) & \leq & 0, & k \in S_\infty^e \end{array} \right. \tag{2.4}$$

define a polyhedral set which is a convex overestimation of the set $\{(x, y) \in X \times Y : g(x, y) \leq 0\}$, as it is depicted in Figure 2.2.

As more terms are included in the piecewise linear underestimation (2.3) and more constraints are included in the polyhedral overestimation (2.4), the better are the corresponding estimations. It comes therefore as no surprise that if the whole sets T^e (or $T^e \cup S_\infty^e$) and

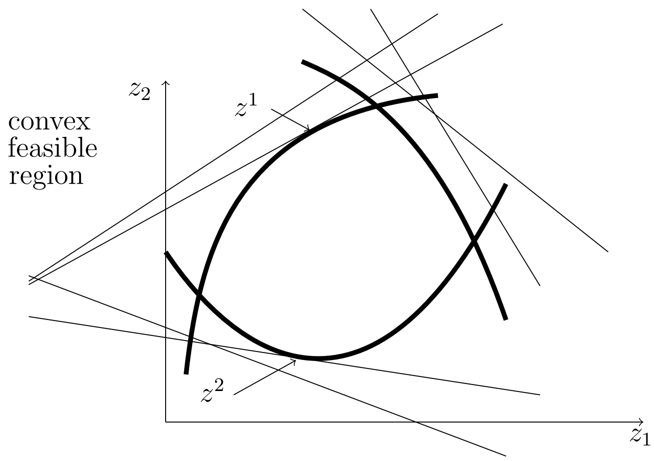

Figure 2.2: Overestimating a feasible region by linear constraints.

$T^e \cup S^e_\infty$ are included in (2.3) and (2.4), then the master MILP becomes equivalent to the original MINLP problem P. This statement is formally stated below (for a proof see [27] or [12], respectively for $\mathrm{P}^{\mathrm{OA}}_{FL}$ or $\mathrm{P}^{\mathrm{OA}}_{PB}$, or Theorem 3.4.1 in Chapter 3 when the inexact scenario there considered is taken as exact). The equivalence says that the optimal solutions (x^*, y^*) of P correspond to the optimal solutions (x^*, y^*, α^*) of $\mathrm{P}^{\mathrm{OA}}_{FL}$ or $\mathrm{P}^{\mathrm{OA}}_{PB}$ with $\alpha^* = f(x^*, y^*)$. The KKT conditions of the NLP subproblems play a major role in the proof.

Theorem 2.2.1. *Let* P *be a convex* MINLP *problem as defined in (1.1). Assume that* P *is feasible with a finite optimal value and that the* KKT *conditions are satisfied at every optimal solution of* $\mathrm{NLP}(x^j)$ *and* $\mathrm{NLPF}_\infty(x^k)$*. Then* P, $\mathrm{P}^{\mathrm{OA}}_{FL}$*, and* $\mathrm{P}^{\mathrm{OA}}_{PB}$ *have the same optimal value.*

We now describe in more detail the OA approach. At each step of an OA algorithm, one tries to solve a subproblem $\mathrm{NLP}(x^p)$, where x^p is chosen as a new discrete assignment. Two results can then occur: either the $\mathrm{NLP}(x^p)$ is feasible and an optimal solution y^p is computed, or this subproblem is found infeasible and an NLP subproblem, say $\mathrm{NLPF}_\infty(x^p)$, is solved, yielding an optimal solution y^p. In the algorithm, the sets T^e and S^e_∞ defined in Section 2.1 will be replaced by:

$$(T^e)^p \; = \; \{j : j \le p,\; x^j \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^j) \text{ is feasible and } y^j \text{ solves } \mathrm{NLP}(x^j)\}$$

and

$$(S^e_\infty)^p \; = \; \{k : k \le p,\; x^k \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^k) \text{ is infeasible and } y^k \text{ solves } \mathrm{NLPF}_\infty(x^k)\}.$$

In order to prevent any x^j, $j \in (T^e)^p$, from becoming the solution of the relaxed master problem to be solved at the p-th iteration, one needs to add the constraint

$$\alpha \; < \; \mathrm{UBD}_e^p,$$

where

$$\mathrm{UBD}_e^p \; = \; \min_{j \leq p,\, j \in (T^e)^p} f(x^j, y^j).$$

The relaxed MILP master problem is then defined in the (x, y, α) variables as follows:

$$(\mathrm{P}_{PB}^{\mathrm{OA}})^p \left\{ \begin{array}{ll} \min & \alpha \\ \text{s.t.} & \alpha \; < \; \mathrm{UBD}_e^p, \\ & \nabla f(x^j, y^j)^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + f(x^j, y^j) \; \leq \; \alpha, \\ & \nabla g(x^j, y^j)^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + g(x^j, y^j) \; \leq \; 0, \\ & \qquad\qquad\qquad\qquad\qquad\qquad \forall j \in (T^e)^p \\ & \nabla f(x^k, y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + f(x^k, y^k) \; \leq \; \alpha, \\ & \nabla g(x^k, y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + g(x^k, y^k) \; \leq \; 0, \\ & \qquad\qquad\qquad\qquad\qquad\qquad \forall k \in (S_\infty^e)^p \\ & x \in X \cap \mathbb{Z}^{n_d}, y \in Y, \alpha \in \mathbb{R}. \end{array} \right.$$

Here we have chosen the OA approach of [12], corresponding to $\mathrm{P}_{PB}^{\mathrm{OA}}$, but a formulation based on $\mathrm{P}_{FL}^{\mathrm{OA}}$ would lead to the same convergence result given later in Theorem 2.2.2. A detailed description of such an outer approximation algorithm can be introduced as follows:

Algorithm 2.2.1 (Outer Approximation).

Initialization

Let x^0 be given. Set $p = 0$, $(T^e)^{-1} = \emptyset$, $(S_\infty^e)^{-1} = \emptyset$, and $\mathrm{UBD}_e = +\infty$.

REPEAT

1. Solve the subproblem $\mathrm{NLP}(x^p)$, or the feasibility subproblem $\mathrm{NLPF}_\infty(x^p)$ provided $\mathrm{NLP}(x^p)$ is infeasible, and let y^p be an optimal solution.

2. Linearize the objective function and constraints at (x^p, y^p). Renew $(T^e)^p = (T^e)^{p-1} \cup \{p\}$ or $(S_\infty^e)^p = (S_\infty^e)^{p-1} \cup \{p\}$.

3. If NLP(x^p) is feasible and $f(x^p, y^p) < \mathrm{UBD}_e$, then update current best point by setting $\bar{x} = x^p, \bar{y} = y^p$, and $\mathrm{UBD}_e = f(x^p, y^p)$.

4. Solve the master problem $(\mathrm{P}_{PB}^{\mathrm{OA}})^p$, obtaining a new discrete assignment x^{p+1} to be tested in the algorithm. Increment p by one unit.

UNTIL (($\mathrm{P}_{PB}^{\mathrm{OA}})^p$ is infeasible).

Such an algorithm terminates in a finite number of steps (a proof is given in Section 3.5, see Theorem 3.5.1, for the case where the KKT conditions of the subproblems are possibly inexact).

Theorem 2.2.2. *Let* P *be a convex* MINLP *problem as defined in (1.1). Assume that either* P *has a finite optimal value or is infeasible, and that the* KKT *conditions are satisfied at every optimal solution of all* NLP *subproblems. Then Algorithm 2.2.1 terminates in a finite number of steps at an optimal solution of* P *or with an indication that* P *is infeasible.*

2.3 Generalized Benders decomposition

In this section, we briefly review the generalized Benders decomposition (GBD) for solving MINLP problems developed by Geoffrion [31].

Suppose that the KKT conditions are satisfied at every solution of the NLP subproblems introduced in Section 2.2. For any $x^j \in X \cap \mathbb{Z}^{n_d}$, if NLP($x^j$) is feasible, let λ^j be the Lagrange multiplier corresponding to its optimal solution y^j. Otherwise, denote by μ^j the Lagrange multiplier corresponding to the optimal solution y^j to $\mathrm{NLPF}_\infty(x^j)$. Then the master problem for GBD can be built as follows (in the variables x and α):

$$\mathrm{P}_{original}^{\mathrm{GBD}} \left\{ \begin{array}{lll} \min & \alpha & \\ \text{s.t.} & \inf_{y \in Y} \left\{ f(x,y) + (\lambda^j)^T g(x,y) \right\} \ \leq \ \alpha, & \forall j \in T^e \\ & \inf_{y \in Y} \left\{ (\mu^k)^T g(x,y) \right\} \ \leq \ 0, & \forall k \in S_\infty^e \\ & x \in X \cap \mathbb{Z}^{n_d}, \alpha \in \mathbb{R}, & \end{array} \right.$$

where T^e and S_∞^e are defined as in Section 2.1.

There is an alternative formulation for the MILP master problem $\mathrm{P}_{original}^{\mathrm{GBD}}$ (in the sense that is implied by $\mathrm{P}^{\mathrm{GBD}}_{original}$) provided by Flippo et al. [29] (see [44] and also [32, 46, 53]):

$$\mathrm{P}^{\mathrm{GBD}} \begin{cases} \min & \alpha \\ \text{s.t.} & f(x^j, y^j) + \nabla_x f(x^j, y^j)^\top (x - x^j) \\ & +(\lambda^j)^\top \left[g(x^j, y^j) + \nabla_x g(x^j, y^j)^\top (x - x^j) \right] \ \leq \ \alpha, \quad j \in T^e \\ & (\mu^k)^\top \left[g(x^k, y^k) + \nabla_x g(x^k, y^k)^\top (x - x^k) \right] \ \leq \ 0, \qquad k \in S^e_\infty \\ & x \in X \cap \mathbb{Z}^{n_d}, \alpha \in \mathbb{R}. \end{cases}$$

In fact, the cuts in $\mathrm{P}^{\mathrm{GBD}}$ can be interpreted as surrogate constraints of the equations in $\mathrm{P}^{\mathrm{GBD}}_{original}$, and one can read a more general proof of this fact in Section 4.3 for the case where the KKT conditions are possibly satisfied inexactly.

The master problem $\mathrm{P}^{\mathrm{GBD}}$ is also not posed in the continuous variables y and contains only one constraint per NLP subproblem. Although this could seem advantageous when compared to OA, numerical experience shows OA to be a faster method (see [23, 28, 32]). Such a behavior is explained by the following fact, for which the proof is postponed to Section 4.1 (see Theorem 4.1.1) in the inexact context of this thesis.

Property 2.3.1. *Given some sets T^e and S^e_∞ (the ones in (2.1) and (2.2) or any subsets of those), the lower bound predicted by the master problem $\mathrm{P}^{\mathrm{OA}}_{FL}$ or $\mathrm{P}^{\mathrm{OA}}_{PB}$ is greater than or equal to the one predicted by the master problem $\mathrm{P}^{\mathrm{GBD}}$.*

Moreover, by [31, 32], we can show the equivalence between $\mathrm{P}^{\mathrm{GBD}}$ and P as follows, in the sense that they have the same optimal value, and that an optimal solution (x^*, y^*) of P corresponds to an optimal solution (x^*, α^*) of $\mathrm{P}^{\mathrm{GBD}}$ with $\alpha^* = f(x^*, y^*)$.

Theorem 2.3.1. *Let* P *be a convex* MINLP *problem as defined in (1.1). Assume that* P *is feasible with a finite optimal value and that the* KKT *conditions are satisfied at every optimal solution of* $\mathrm{NLP}(x^j)$ *and* $\mathrm{NLPF}_\infty(x^k)$*. Then* P *and* $\mathrm{P}^{\mathrm{GBD}}$ *have the same optimal value.*

The GBD strategy relies on the iterative solution of the following relaxed MILP:

$$(\mathrm{P}^{\mathrm{GBD}})^p \begin{cases} \min & \alpha \\ \text{s.t.} & \alpha \ < \ \mathrm{UBD}^p_e, \\ & f(x^j, y^j) + \nabla_x f(x^j, y^j)^\top (x - x^j) \\ & +(\lambda^j)^\top [g(x^j, y^j) + \nabla_x g(x^j, y^j)^\top (x - x^j)] \ \leq \ \alpha, \quad j \in (T^e)^p \\ & (\mu^k)^\top [g(x^k, y^k) + \nabla_x g(x^k, y^k)^\top (x - x^k)] \ \leq \ 0, \qquad k \in (S^e_\infty)^p \\ & x \in X \cap \mathbb{Z}^{n_d}, \alpha \in \mathbb{R}, \end{cases}$$

where the definitions of UBD_e^p, $(T^e)^p$, and $(S_\infty^e)^p$ are the same as those in Section 2.2. The GBD algorithm is presented below.

Algorithm 2.3.1 (Generalized Benders Decomposition).

Initialization

Let x^0 be given. Set $p = 0$, $(T^e)^{-1} = \emptyset$, $(S_\infty^e)^{-1} = \emptyset$, and $\mathrm{UBD}_e = +\infty$.

REPEAT

1. Solve the subproblem $\mathrm{NLP}(x^p)$, or the feasibility subproblem $\mathrm{NLPF}_\infty(x^p)$ provided $\mathrm{NLP}(x^p)$ is infeasible, and let y^p be an optimal solution.
2. Linearize the objective function and constraints at (x^p, y^p). Renew $(T^e)^p = (T^e)^{p-1} \cup \{p\}$ or $(S_\infty^e)^p = (S_\infty^e)^{p-1} \cup \{p\}$.
3. If $\mathrm{NLP}(x^p)$ is feasible and $f(x^p, y^p) < \mathrm{UBD}_e$, then update current best point by setting $\bar{x} = x^p, \bar{y} = y^p$, and $\mathrm{UBD}_e = f(x^p, y^p)$.
4. Solve the master problem $(\mathrm{P}^{\mathrm{GBD}})^p$, obtaining a new discrete assignment x^{p+1} to be tested in the algorithm. Increment p by one unit.

UNTIL $((\mathrm{P}^{\mathrm{GBD}})^p$ is infeasible).

The GBD algorithm is shown to terminate in a finite number of steps (a proof is given in Section 4.2, see Theorem 4.2.1, for the case where the KKT conditions of the NLP subproblems are possibly inexact).

Theorem 2.3.2. *Let* P *be a convex* MINLP *problem as defined in (1.1). Assume that either* P *has a finite optimal value or is infeasible, and that the* KKT *conditions are satisfied at every optimal solution of all* NLP *subproblems. Then Algorithm 2.3.1 terminates in a finite number of steps at an optimal solution of* P *or with an indication that* P *is infeasible.*

2.4 The extended cutting plane method

Westerlund and Pettersson [61] proposed the extended cutting plane (ECP) method for convex MINLP, which is an extension of Kelly's cutting plane method [36] for convex NLP. ECP does not rely on the solution of NLP subproblems. The main idea is also to keep updating and solving a relaxed MILP master problem. If its optimal solution is not feasible to the original problem, one generates a cutting plane using the most violated constraint at this point and add it to the MILP relaxation, until its solution becomes feasible with

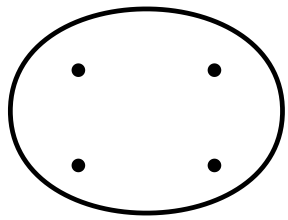

Figure 2.3: The MINLP feasible integer points.

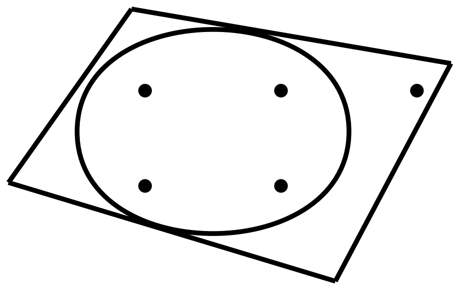

Figure 2.4: An MILP ECP relaxation.

respect to the MINLP constraints $g(x, y) \leq 0$. One can see this process more clearly from Figures 2.3, 2.4, 2.5, and 2.6 (taken from [42]).

For the moment, suppose that $f(x, y)$ is a linear function of the form $f(x, y) = c_x^\top x + c_y^\top y$. The ECP method would start by solving the MILP (obtained from the original MINLP problem P by removing the nonlinear constraints $g(x, y) \leq 0$):

$$\begin{cases} \min & c_x^\top x + c_y^\top y \\ \text{s.t.} & x \in X \cap \mathbb{Z}^{n_d}, y \in Y, \end{cases}$$

or, equivalently,

$$(\mathrm{P}^{\mathrm{ECP}})^0 \begin{cases} \min & \alpha \\ \text{s.t.} & f(x, y) \ \leq \ \alpha, \\ & x \in X \cap \mathbb{Z}^{n_d}, y \in Y, \alpha \in \mathbb{R}. \end{cases}$$

Let (x^0, y^0, α^0) be its optimal solution. To construct the subsequent master MILP problems, given a new incoming point $(x^k, y^k) \in (X \cap \mathbb{Z}^{n_d}) \times Y$, we define a new function $G_k(x, y)$ as

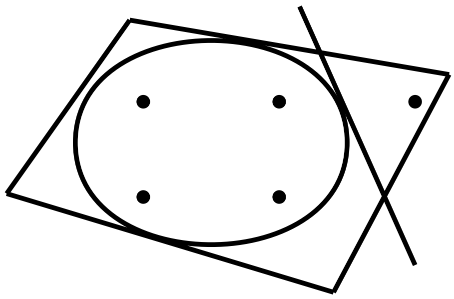

Figure 2.5: A new cutting plane generated by ECP.

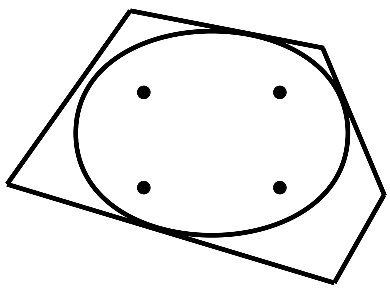

Figure 2.6: The new MILP ECP relaxation.

follows:

$$G_k(x,y) \;=\; g_i(x,y), \quad \text{where} \quad i \in \operatorname*{arg\,max}_{j\in\{1,\ldots,m\}} g_j(x^k,y^k).$$

Then one can build the relaxed master MILP problem of the method ECP in the following way (assuming $p \geq 1$ and considering as variables x, y, and α):

$$(\mathrm{P}^{\mathrm{ECP}})^p \left\{ \begin{array}{lll} \min & \alpha & \\ \text{s.t.} & \nabla f(x^k,y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + f(x^k,y^k) \;\leq\; \alpha, & \forall k = 0,\ldots,p-1, \\ & \nabla G_k(x^k,y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + G_k(x^k,y^k) \;\leq\; 0, & \forall k = 0,\ldots,p-1, \\ & x \in X \cap \mathbb{Z}^{n_d}, y \in Y, \alpha \in \mathbb{R}. & \end{array} \right.$$

Let (x^p, y^p, α^p) denote its optimal solution and α^p the corresponding optimal value. One can easily see that the optimal values form a monotonically increasing sequence, i.e., that

$$\alpha^p \;\geq\; \alpha^{p-1} \;\geq\; \cdots \;\geq\; \alpha^0.$$

Moreover, once a solution (x^k, y^k, α^k) of the relaxed MILP problem is feasible, i.e., $g(x^k, y^k) \;\leq$ 0, (x^k, y^k) will be a solution of the original problem P. The ECP algorithm is detailed next for the case $f(x,y) = c_x^\top x + c_y^\top y$.

Algorithm 2.4.1 (Extended Cutting Plane Method).

Initialization

Set $p = 0$.

REPEAT

1. Solve the relaxed master problem $(\mathrm{P}^{\mathrm{ECP}})^p$, and let an optimal solution be denoted by (x^p, y^p, α^p).

2. If $G_p(x^p, y^p) \leq 0$, update current best point by setting $\bar{x} = x^p$ and $\bar{y} = y^p$. Otherwise, linearize the objective function and the constraint $G_p(x,y)$ at (x^p, y^p) and use these linearizations to define the relaxed master problem $(\mathrm{P}^{\mathrm{ECP}})^{p+1}$. Increment p by one unit.

UNTIL $(G_p(x^p, y^p) \leq 0)$.

The convergence of the extended cutting plane method is formally stated as follows (see [61] for a proof).

Theorem 2.4.1. *Let* P *be a convex* MINLP *problem as defined in (1.1) for which* Y *is compact and assume that* P *has a finite optimal value. Then Algorithm 2.4.1 terminates in a finite number of steps at an optimal solution of* P*.*

If $f(x, y)$ is not linear, the final point obtained by Algorithm 2.4.1 may not be the optimal solution of the original problem. But we can guarantee that all the above considerations would still be true by transforming the original problem so that it has a linear objective function. One just needs to introduce an auxiliary variable $\xi \in \mathbb{R}$ and move the original objective function into the constraints using $f(x, y) - \xi \leq 0$, and replace the original continuous variable $y \in Y \subseteq \mathbb{R}^{n_c}$ by $(y, \xi) \in (Y \times \mathbb{R}) \subseteq \mathbb{R}^{n_c+1}$.

Chapter 3

Inexact outer approximation

The OA algorithm described in Chapter 2 required the exact solution of all NLP subproblems. In this chapter, we want to investigate how the OA method is affected when the NLP subproblems are solved inexactly. We start in Section 3.1 by introducing the inexact version of the first-order necessary conditions for the NLP subproblems. Then we show in Section 3.2 how the cuts in the master OA problem can be changed to incorporate the inexact residuals of these KKT conditions. Section 3.3 introduces and comments on the conditions required for validating the analysis in the inexact case. This analysis is given in Section 3.4 for the equivalence between a perturbed MINLP problem and the corresponding master problem, and in Section 3.5 for the convergence of the algorithm.

3.1 Inexact solution of NLP subproblems

Similarly to the exact case in Section 2.1, given any $x^j \in X \cap \mathbb{Z}^{n_d}$, if the NLP subproblem

$$\mathrm{NLP}(x^j) \begin{cases} \min & f(x^j, y) \\ \text{s.t.} & g(x^j, y) \ \leq \ 0, \\ & y \in Y, \end{cases}$$

is feasible, we use y^j to denote its approximate optimal solution. For an $x^k \in X \cap \mathbb{Z}^{n_d}$ for which $\mathrm{NLP}(x^k)$ is infeasible, we use y^k to represent an approximate optimal solution of the feasibility subproblem

$$\mathrm{NLPF}_\infty(x^k) \begin{cases} \min & u \\ \text{s.t.} & g_i(x^k, y) \ \leq \ u, \ i = 1, \ldots, m, \\ & y \in Y, u \in \mathbb{R}. \end{cases}$$

Following what we wrote in the exact case, we enumerate all the discrete assignments using the following sets of indices:

$$T = \{j : x^j \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^j) \text{ is feasible and } y^j \text{ approximately solves } \mathrm{NLP}(x^j)\}$$

and

$$S_\infty = \{k : x^k \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^k) \text{ is infeasible and } y^k \text{ approximately solves } \mathrm{NLPF}_\infty(x^k)\}.$$

As in the exact case, we suppose that the constraints $y \in Y$ are part of the constraints $g(x^j, y) \leq 0$ and $g_i(x^k, y) \leq u$, $i = 1, \ldots, m$, in the subproblems $\mathrm{NLP}(x^j)$ and $\mathrm{NLPF}_\infty(x^k)$, respectively. In addition, let us assume that the approximate optimal solutions of the NLP subproblems satisfy an inexact form of the corresponding first-order necessary KKT conditions. More particularly, in the case of $\mathrm{NLP}(x^j)$, we assume the existence of $\lambda^j \in \mathbb{R}^m_+, r^j \in \mathbb{R}^{n_c}$, and $s^j \in \mathbb{R}^m$, such that

$$\nabla_y f(x^j, y^j) + \sum_{i=1}^m \lambda_i^j \nabla_y g_i(x^j, y^j) = r^j, \tag{3.1}$$

$$\lambda_i^j g_i(x^j, y^j) = s_i^j, \quad i = 1, \ldots, m. \tag{3.2}$$

When $\mathrm{NLP}(x^k)$ is infeasible, we assume, for $\mathrm{NLPF}_\infty(x^k)$, the existence of $\mu^k \in \mathbb{R}^m_+$, $z^k \in \mathbb{R}^m$, $w^k \in \mathbb{R}$, and $v^k \in \mathbb{R}^{n_c}$, such that

$$\sum_{i=1}^m \mu_i^k \nabla_y g_i(x^k, y^k) = v^k, \tag{3.3}$$

$$1 - \sum_{i=1}^m \mu_i^k = w^k, \tag{3.4}$$

$$\mu_i^k (g_i(x^k, y^k) - u^k) = z_i^k, \quad i = 1, \ldots, m. \tag{3.5}$$

The size of the residuals r^j, s^j, v^k, w^k, and z^k will be required to satisfy appropriate assumptions. The inexact version of OA (and the inexact GBD in Chapter 4) studied in this thesis will attempt to find the best pair among all of the form (x^j, y^j) corresponding to $j \in T$. Implicitly, we are thus redefining a perturbed version of problem P and will denote it by $\mathcal{P}$:

$$\mathcal{P} \left\{ \min_{j \in T} \quad f(x^j, y^j) \right. \tag{3.6}$$

This problem is well defined if $T \neq \emptyset$ which in turn can be assumed when the original MINLP problem P has a finite optimal value.

Finally we would like to stress that the point y^j satisfying the inexact KKT conditions (3.1)–(3.2) of the subproblem NLP(x^j) can be interpreted as a solution of a perturbed NLP subproblem, which has the form

$$\text{perturbed NLP}(x^j)\begin{cases} \min & f(x^j,y)-(r^j)^\top(y-y^j) \\ \text{s.t.} & g(x^j,y)-t^j \;\le\; 0, \\ & y\in Y, \end{cases}$$

where, for $i=1,\ldots,m$,

$$t_i^j \;=\; \begin{cases} \frac{s_i^j}{\lambda_i^j}, & \text{if } \lambda_i^j \;>\; 0, \\ 0, & \text{if } \lambda_i^j \;=\; 0. \end{cases} \tag{3.7}$$

The data of this perturbed subproblem depends, however, on the approximate optimal solution y^j and inexact Lagrange multipliers λ^j. Similarly, the point y^k satisfying the inexact KKT conditions (3.3)–(3.5) of the subproblem NLPF$_\infty$(x^k) can be interpreted as a solution of the following perturbed NLP subproblem

$$\text{perturbed NLPF}_\infty(x^k)\begin{cases} \min & u-(v^k)^\top(y-y^k)-w^k(u-u^k) \\ \text{s.t.} & g_i(x^k,y)-u-c_i^k \;\le\; 0,\; i=1,\ldots,m, \\ & y\in Y, u\in\mathbb{R}, \end{cases}$$

where, for $i=1,\ldots,m$,

$$c_i^k \;=\; \begin{cases} \frac{z_i^k}{\mu_i^k}, & \text{if } \mu_i^k \;>\; 0, \\ 0, & \text{if } \mu_i^k \;=\; 0, \end{cases}$$

which also depends on unknown information (approximate optimal solution (y^k,u^k) and inexact Lagrange multiplies μ^k).

3.2 Inexact OA master problem

As shown in Section 2.2, OA relies on the fact that the original problem P is equivalent to an MILP (master problem) formed by minimizing the least of the linearized forms of f for indices in T^e (or $T^e \cup S^e_\infty$) subject to the linearized forms of g for indices in T^e and S^e_∞. When the NLP subproblems are solved inexactly (see the definitions in Section 3.1), one has to consider perturbed forms of such cuts or linearized forms in order to keep an equivalence, this time to the perturbed problem $\mathcal{P}$. In turn, these inexact cuts lead to a different, perturbed

MILP (master problem) given by

$$
\mathcal{P}^{\mathrm{OA}} \begin{cases} \min & \alpha \\ \text{s.t.} & \begin{pmatrix} \nabla_x f(x^j, y^j) \\ \nabla_y f(x^j, y^j) - r^j \end{pmatrix}^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + f(x^j, y^j) \ \leq \ \alpha, \\ & \nabla g(x^j, y^j)^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + g(x^j, y^j) \ \leq \ t^j, \ \ \forall j \in T, \\ & \nabla f(x^k, y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + f(x^k, y^k) \ \leq \ \alpha, \\ & \begin{pmatrix} \nabla_x g_i(x^k, y^k) \\ \nabla_y g_i(x^k, y^k) - \frac{1}{1-w^k} v^k \end{pmatrix}^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + g_i(x^k, y^k) \ \leq \ a_i^k, \\ & i = 1, \ldots, m, \ \forall k \in S_\infty, \\ & x \in X \cap \mathbb{Z}^{n_d}, y \in Y, \alpha \in \mathbb{R}, \end{cases}
$$

where t^j is given by (3.7), for $i = 1, \ldots, m$,

$$
a_i^k \ = \ \begin{cases} \frac{m_+^k z_i^k - w^k u^k}{m_+^k \mu_i^k}, & \text{if } \mu_i^k \ > \ 0, \\ 0, & \text{if } \mu_i^k \ = \ 0 \end{cases} \tag{3.8}
$$

(m_+^k is the number of positive components in μ^k), and $w^k < 1$. Note that when r, s, v, w, and z are zero, we obtain the well-known master problem in OA. We have added the linearizations of the objective function in the infeasible cases, as suggested in [12] and discussed in Section 2.2.

3.3 Assumptions for the inexact case

From the convexity and continuous differentiability of f and g, we know that, for any $(x^l, y^l) \in \mathbb{R}^{n_d} \times \mathbb{R}^{n_c}$,

$$
f(x, y) \ \geq \ f(x^l, y^l) + \nabla f(x^l, y^l)^\top \begin{pmatrix} x - x^l \\ y - y^l \end{pmatrix}, \tag{3.9}
$$

$$
g(x, y) \ \geq \ g(x^l, y^l) + \nabla g(x^l, y^l)^\top \begin{pmatrix} x - x^l \\ y - y^l \end{pmatrix}. \tag{3.10}
$$

In addition, when y^j is a feasible point of NLP(x^j), we obtain from (3.10) and $g(x^j, y^j) \leq 0$ that

$$0 \;\geq\; g(x^l, y^l) + \nabla g(x^l, y^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix}. \tag{3.11}$$

The inexact OA method reported in this chapter, as well as the GBD method of the next chapter, requires the residuals of the inexact KKT conditions to satisfy the bounds given in the next two assumptions, in order to validate the equivalence between perturbed and master problems, and to ensure finiteness of the respective algorithms. We first give the bounds on the residuals r and s for the feasible case.

Assumption 3.3.1. *Given any $l, j \in T$, with $l \neq j$, assume that*

$$\|r^l\| \;\leq\; \frac{-\tau \left[(\nabla f^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + f^l - f^j \right]}{\|y^j - y^l\|},$$

for some $\tau \in [0,1]$, and

$$|s_i^l| \;\leq\; -\sigma_i \lambda_i^l \left[(\nabla g_i^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + g_i^l \right],$$

for some $\sigma_i \in [0,1]$, $i = 1, \ldots, m$.

Essentially, these bounds will ensure that the above convexity properties will still imply the inexact cuts for all combinations of discrete assignments $l, j \in T$. In fact, the inexact cuts are defined by perturbed gradients of the form $\nabla_y f(x^j, y^j) - r^j$, yielding perturbed slopes. However, if the size of the perturbation r^j is relatively small, the corresponding inexact or perturbed cut is still valid at all remaining discrete assignments, and this is what is in fact imposed by Assumption 3.3.1 (see Figure 3.1). In the case of exact cuts, convexity trivially enforces the validness of the cuts (see also Figure 3.1), but the presence of perturbed gradients may destroy it unless we restrict the size of such perturbations.

Now, we state the bounds for the residuals v, w, and z in the infeasible case.

Assumption 3.3.2. *Given any $j \in T$ and any $k \in S_\infty$, and for all $i \in \{1, \ldots, m\}$, if $\mu_i^k \neq 0$, assume that*

$$\frac{1}{1 - w^k} \|v^k\| \, \|y^j - y^k\| + \frac{1}{\mu_i^k} |z_i^k| + \frac{u^k}{m_+^k \mu_i^k} |w^k| \;\leq\; -\beta_i \left[(\nabla g_i^k)^\top \begin{pmatrix} x^j - x^k \\ y^j - y^k \end{pmatrix} + g_i^k \right],$$

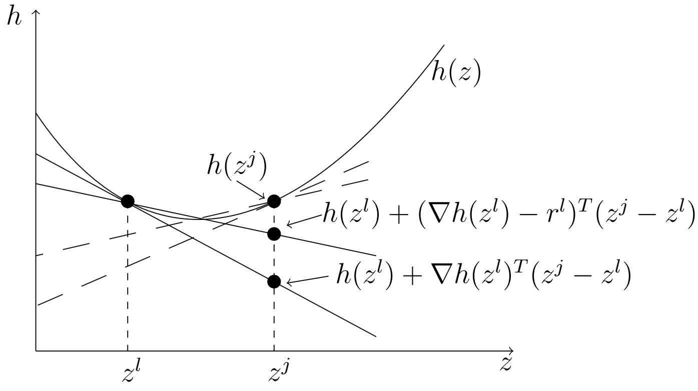

Figure 3.1: Exact and inexact cuts under convexity (for a generic function h).

for some $\beta_i \in [0,1]$, *otherwise, assume that*

$$\frac{1}{1-w^k}\|v^k\|\,\|y^j - y^k\| \;\leq\; -\eta_i \left[(\nabla g_i^k)^\top \begin{pmatrix} x^j - x^k \\ y^j - y^k \end{pmatrix} + g_i^k\right],$$

for some $\eta_i \in [0,1]$.

The motivation and restrictiveness of Assumption 3.3.2 are the same as in Assumption 3.3.1.

It is important to note that the complete satisfaction of the inequalities stated in Assumptions 3.3.1 and 3.3.2 requires the inexact solution for all possible assignments of the discrete variables. As one can see from the proof of Theorem 3.4.1 below, the non satisfaction of one of the inequalities in these assumptions may have as a potential effect the deterioration of the upper bound on the optimal value of the original MINLP and consequently failure to determine an optimal solution.

3.4 Equivalence between perturbed and master problems for OA

We are now in a position to state the equivalence between the original, perturbed MINLP problem and the MILP master problem $\mathcal{P}^{\mathrm{OA}}$.

Theorem 3.4.1. *Let* P *be the convex* MINLP *problem (1.1) and* $\mathcal{P}$ *be its perturbed problem as defined in (3.6). Suppose that the classification of feasibility for the* NLP *subproblems* NLP(x^j) *is made exactly. Assume that* P *is feasible with a finite optimal value and that the residuals of the* KKT *conditions of the* NLP *subproblems satisfy Assumptions 3.3.1 and 3.3.2. Then* $\mathcal{P}^{\mathrm{OA}}$ *and* $\mathcal{P}$ *have the same optimal value.*

Proof. The proof follows closely the lines of the proof of [12, Theorem 1]. Since problem P has a finite optimal value it follows (i) that for every $x \in X \cap \mathbb{Z}^{n_d}$, either problem NLP$(x)$ is feasible with a finite optimal value or it is infeasible, (ii) that the sets T and S_∞ are well defined, and (iii) that the set T is nonempty. Now, given any $x^l \in X \cap \mathbb{Z}^{n_d}$ with $l \in T \cup S_\infty$, let $\mathcal{P}^{\mathrm{OA}}_{x^l}$ denote the problem in α and y obtained from $\mathcal{P}^{\mathrm{OA}}$ when x is fixed to x^l. First we will prove that problem $\mathcal{P}^{\mathrm{OA}}_{x^k}$ is infeasible for every $k \in S_\infty$.

Part I. Establishing infeasibility of $\mathcal{P}^{\mathrm{OA}}_{x^k}$ for $k \in S_\infty$.

In this case, problem NLP(x^k) is infeasible and y^k is an approximate optimal solution of NLPF$_\infty(x^k)$ with corresponding inexact non-negative Lagrange multipliers μ^k. When we set $x = x^k$, the corresponding constraints in $\mathcal{P}^{\mathrm{OA}}$ will result in

$$\left(\nabla_y g_i(x^k, y^k) - \frac{1}{1-w^k} v^k\right)^\top (y - y^k) + g_i(x^k, y^k) \ \le\ a_i^k, \tag{3.12}$$

for $i = 1, \ldots, m$. Multiplying the inequalities in (3.12) by the non-negative multipliers $\mu_1^k, \ldots, \mu_m^k$, and summing them up, one obtains

$$\left(\sum_{i=1}^{m} \mu_i^k \nabla_y g_i(x^k, y^k) - v^k\right)^\top (y - y^k) \ \le\ \sum_{i=1}^{m} (z_i^k - \mu_i^k g_i(x^k, y^k)) - w^k u^k. \tag{3.13}$$

By using (3.3), one can see that the left hand side of the inequality in (3.13) is equal to 0. On the other hand, by using equation (3.5), the right hand side of the inequality in (3.13) results in $\sum_{i=1}^{m}(z_i^k - \mu_i^k g_i(x^k, y^k)) - w^k u^k = -(\Sigma_{i=1}^{m} \mu_i^k + w^k) u^k$, which is equal to $-u^k$ by (3.4). Since NLP(x^k) is infeasible, $-u^k$ must be strictly negative. We have thus proved that the inequality (3.13) has no solution y.

This derivation implies that the minimum value of $\mathcal{P}^{\mathrm{OA}}$ should be found as the minimum value of $\mathcal{P}^{\mathrm{OA}}_{x^j}$ over all $x^j \in X \cap \mathbb{Z}^{n_d}$ with $j \in T$. We prove in the next two separate subparts that, for every $j \in T$, the optimal value $\bar{\alpha}^j$ of $\mathcal{P}^{\mathrm{OA}}_{x^j}$ coincides with the approximate optimal value of NLP(x^j).

Part II. Establishing that $\mathcal{P}^{\mathrm{OA}}_{x^j}$ has the same objective value as the perturbed NLP(x^j) for $j \in T$.

We will show next that $(y^j, f(x^j, y^j))$ is a feasible solution of $\mathcal{P}^{\mathrm{OA}}_{x^j}$, and therefore that $f(x^j, y^j)$ is an upper bound on the optimal value $\bar{\alpha}^j$ of $\mathcal{P}^{\mathrm{OA}}_{x^j}$.

Part II–A. Establishing that $f(x^j, y^j)$ is an upper bound for the optimal value of $\mathcal{P}^{\mathrm{OA}}_{x^j}$ for $j \in T$.

In this case, it is easy to see that $\mathcal{P}^{\mathrm{OA}}_{x^j}$ contains all the constraints indexed by $l \in T$

$$\begin{pmatrix} \nabla_x f(x^l, y^l) \\ \nabla_y f(x^l, y^l) - r^l \end{pmatrix}^\top \begin{pmatrix} x^j - x^l \\ y - y^l \end{pmatrix} + f(x^l, y^l) \;\leq\; \alpha, \tag{3.14}$$

$$\nabla g(x^l, y^l)^\top \begin{pmatrix} x^j - x^l \\ y - y^l \end{pmatrix} + g(x^l, y^l) \;\leq\; t^l, \tag{3.15}$$

where, for $i = 1, \ldots, m$,

$$t^l_i \;=\; \begin{cases} \frac{s^l_i}{\lambda^l_i}, & \text{if } \lambda^l_i \;>\; 0, \\ 0, & \text{if } \lambda^l_i \;=\; 0, \end{cases}$$

as well as all the constraints indexed by $k \in S_\infty$

$$\nabla f(x^k, y^k)^\top \begin{pmatrix} x^j - x^k \\ y - y^k \end{pmatrix} + f(x^k, y^k) \;\leq\; \alpha, \tag{3.16}$$

and by $k \in S_\infty$ and $i \in \{1, \ldots, m\}$

$$\begin{pmatrix} \nabla_x g_i(x^k, y^k) \\ \nabla_y g_i(x^k, y^k) - \frac{1}{1-w^k} v^k \end{pmatrix}^\top \begin{pmatrix} x^j - x^k \\ y - y^k \end{pmatrix} + g_i(x^k, y^k) \;\leq\; a^k_i, \tag{3.17}$$

where a^k_i is given as in (3.8).

First take any $l \in T$ and assume that y^l is an approximate optimal solution of NLP(x^l) with corresponding inexact non-negative Lagrange multipliers λ^l. If $l = j$, it is easy to verify that $(y^j, f(x^j, y^j))$ satisfies (3.14) and (3.15). Assume then that $l \neq j$. From Assumption 3.3.1, we know that, for some $\tau \in [0, 1]$,

$$-(r^l)^\top (y^j - y^l) \;\leq\; \|r^l\| \, \|y^j - y^l\| \;\leq\; -\tau \left[(\nabla f^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + f^l - f^j \right].$$

Thus,

$$\begin{aligned} &\left[(\nabla f^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + f^l - f^j \right] - (r^l)^\top (y^j - y^l) \\ &\qquad \leq\; (1 - \tau) \left[(\nabla f^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + f^l - f^j \right] \;\leq\; 0, \end{aligned}$$

where the last inequality comes from $1 - \tau \geq 0$ and (3.9) with $(x, y) = (x^j, y^j)$. We then see that (3.14) is satisfied with $\alpha = f(x^j, y^j)$ and $y = y^j$.

Now, from Assumption 3.3.1, one has for some $\sigma_i \in [0,1], i = 1, \ldots, m$,

$$\begin{aligned}
\lambda_i^l \left[(\nabla g_i^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + g_i^l \right] - s_i^l \;&\le\; \lambda_i^l \left[(\nabla g_i^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + g_i^l \right] \\
&\quad - \sigma_i \lambda_i^l \left[(\nabla g_i^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + g_i^l \right] \\
&\le\; (1 - \sigma_i) \lambda_i^l \left[(\nabla g_i^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + g_i^l \right] \\
&\le\; 0,
\end{aligned}$$

where the last inequality is justified by (3.11) and $\sigma_i \in [0,1]$. Thus,

$$\lambda_i^l \left[(\nabla g_i^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + g_i^l \right] \;\le\; s_i^l, \; i = 1, \ldots, m. \tag{3.18}$$

If λ_i^l is equal to 0, so is t_i^l by its definition and we see that $(y^j, f(x^j, y^j))$ satisfies the constraints (3.15) with $y = y^j$. If $\lambda_i^l \neq 0$, then (3.18) can be written as:

$$\nabla g_i(x^l, y^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + g_i(x^l, y^l) \;\le\; \frac{s_i^l}{\lambda_i^l} \;=\; t_i^l,$$

which also shows that the constraints (3.15) hold with $y = y^j$.

Finally, we take any $k \in S_\infty$ and assume that y^k is an approximate optimal solution of $\mathrm{NLPF}_\infty(x^k)$ with corresponding inexact Lagrange multipliers μ^k. It results trivially from the assumption on convexity that the constraints (3.16) are satisfied with $y = y^j$ and $\alpha = f(x^j, y^j)$. Now, for every $i \in \{1, \ldots, m\}$, if $\mu_i^k \neq 0$, from Assumption 3.3.2, we have for some $\beta_i \in [0,1]$, that

$$-\frac{1}{1 - w^k}(v^k)^\top (y^j - y^k) - \frac{1}{\mu_i^k} z_i^k + \frac{u^k}{m_+^k \mu_i^k} w^k \;\le\; -\beta_i \left[(\nabla g_i^k)^\top \begin{pmatrix} x^j - x^k \\ y^j - y^k \end{pmatrix} + g_i^k \right],$$

i.e.,

$$-\frac{1}{1 - w^k}(v^k)^\top (y^j - y^k) - a_i^k \;\le\; -\beta_i \left[(\nabla g_i^k)^\top \begin{pmatrix} x^j - x^k \\ y^j - y^k \end{pmatrix} + g_i^k \right]$$

by the definition of a_i^k. Thus, the constraints (3.17) are satisfied with $y = y^j$. When $\mu_i^k = 0$, it results that $a_i^k = 0$ by its definition and, also by Assumption 3.3.2, we have that, for some

$\eta_i \in [0, 1]$,

$$
\begin{aligned}
(\nabla g_i^k)^\top \begin{pmatrix} x^j - x^k \\ y^j - y^k \end{pmatrix} + g_i^k - \frac{1}{1 - w^k}(v^k)^\top (y^j - y^k) \;\; &\leq \;\; (1 - \eta_i) \left[(\nabla g_i^k)^\top \begin{pmatrix} x^j - x^k \\ y^j - y^k \end{pmatrix} + g_i^k \right] \\
&\leq \;\; 0.
\end{aligned}
$$

This also shows that the constraints (3.17) hold with $y = y^j$.

We can therefore say that $(y^j, f(x^j, y^j))$ is a feasible point of $\mathcal{P}_{x^j}^{\mathrm{OA}}$, and thus $\bar{\alpha}^j \leq f(x^j, y^j)$. Next, we will prove that $f(x^j, y^j)$ is also a lower bound, i.e., $\bar{\alpha}^j \geq f(x^j, y^j)$.

Part II–B. Establishing that $f(x^j, y^j)$ is a lower bound for the optimal value of $\mathcal{P}_{x^j}^{\mathrm{OA}}$ for $j \in T$.

Recall that y^j is an approximate optimal solution of NLP(x^j) satisfying the inexact KKT conditions (3.1) and (3.2). By construction, any solution of $\mathcal{P}_{x^j}^{\mathrm{OA}}$ has to satisfy the inexact outer-approximation constraints:

$$
\begin{aligned}
(\nabla_y f(x^j, y^j) - r^j)^\top (y - y^j) + f(x^j, y^j) \;\; &\leq \;\; \alpha, &(3.19)\\
\nabla_y g(x^j, y^j)^\top (y - y^j) + g(x^j, y^j) \;\; &\leq \;\; t^j. &(3.20)
\end{aligned}
$$

Multiplying the inequalities (3.20) by the non-negative multipliers $\lambda_1^j, \ldots, \lambda_m^j$ and summing them together with (3.19), one obtains

$$
(\nabla_y f(x^j, y^j) - r^j)^\top (y - y^j) + f(x^j, y^j) + \sum_{i=1}^{m} \lambda_i^j (\nabla_y g_i(x^j, y^j)^\top (y - y^j) + g_i(x^j, y^j) - s_i^j) \leq \; \alpha. \tag{3.21}
$$

The left hand side of the inequality (3.21) can be rewritten as:

$$
\left(\nabla_y f(x^j, y^j) + \sum_{i=1}^{m} \lambda_i^j \nabla_y g_i(x^j, y^j) - r^j \right)^\top (y - y^j) + \sum_{i=1}^{m} (\lambda_i^j g_i(x^j, y^j) - s_i^j) + f(x^j, y^j).
$$

By using (3.1) and (3.2), this quantity is equal to $f(x^j, y^j)$, and it follows that inequality (3.21) is equivalent to $f(x^j, y^j) \leq \alpha$.

In conclusion, for any $x^j \in X \cap \mathbb{Z}^{n_d}$ with $j \in T$, problems $\mathcal{P}_{x^j}^{\mathrm{OA}}$ and perturbed NLP(x^j) have the same optimal value. In other words, the MILP problem $\mathcal{P}^{\mathrm{OA}}$ has the same optimal value as the perturbed problem $\mathcal{P}$ given by (3.6). $\square$

It is easy to see that when the KKT conditions are exact (i.e., all the right hand side residuals are zero), we recover Theorem 2.2.1 in Chapter 2. Moreover, this result would remain true if we had used, as NLP feasibility subproblem, NLPF$_1(x^k)$ instead of NLPF$_\infty(x^k)$ (recall that NLPF$_1(x^k)$ has been introduced in Section 2.1).

3.5 Inexact OA algorithm

One knows that the outer approximation algorithm terminates finitely when the MINLP problem P is convex and when the optimal solutions of the NLP subproblems satisfy the first-order KKT conditions (see Section 2.2). In this section, we will extend the outer approximation algorithm to the inexact solution of the NLP subproblems by incorporating the corresponding residuals in the cuts of the master problems.

As in the exact case (see Section 2.2), at each step of the inexact OA algorithm, one tries to solve a subproblem $\mathrm{NLP}(x^p)$, where x^p is chosen as a new discrete assignment. Two results can then occur: either $\mathrm{NLP}(x^p)$ is feasible and an approximate optimal solution y^p can be given, or this subproblem is found infeasible and another NLP subproblem, say $\mathrm{NLPF}_\infty(x^p)$, is solved, yielding an approximate optimal solution y^p. In the algorithm, the sets T and S_∞ defined in the Section 3.1 will be replaced by:

$$T^p = \{j : j \le p,\ x^j \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^j) \text{ is feasible and } y^j \text{ approximately solves} \, \mathrm{NLP}(x^j)\}$$

and

$$\begin{aligned}(S_\infty)^p = & \{k : k \le p,\ x^k \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^k) \text{ is infeasible and } y^k \text{ approximately solves} \\ & \mathrm{NLPF}_\infty(x^k)\}.\end{aligned}$$

As before, y^j and y^k denote the approximate solutions of $\mathrm{NLP}(x^j)$ and $\mathrm{NLPF}_\infty(x^k)$, respectively. In order to prevent any x^j, $j \in T^p$, from becoming the solution of the relaxed master problem to be solved at the p-th iteration, one needs to add the constraint

$$\alpha < \mathrm{UBD}^p,$$

where

$$\mathrm{UBD}^p = \min_{j \le p, j \in T^p} f(x^j, y^j).$$

Then we define the following inexact relaxed MILP master problem

$$
(\mathcal{P}^{\mathrm{OA}})^p \begin{cases} \min & \alpha \\ \text{s.t.} & \alpha \ < \ \mathrm{UBD}^p, \\ & \begin{pmatrix} \nabla_x f(x^j, y^j) \\ \nabla_y f(x^j, y^j) - r^j \end{pmatrix}^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + f(x^j, y^j) \ \leq \ \alpha, \\ & \nabla g(x^j, y^j)^\top \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix} + g(x^j, y^j) \ \leq \ t^j, \ \ \forall j \in T^p, \\ & \nabla f(x^k, y^k)^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + f(x^k, y^k) \ \leq \ \alpha, \\ & \begin{pmatrix} \nabla_x g_i(x^k, y^k) \\ \nabla_y g_i(x^k, y^k) - \frac{1}{1-w^k} v^k \end{pmatrix}^\top \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + g_i(x^k, y^k) \ \leq \ a_i^k, \\ & i = 1, \ldots, m, \ \ \forall k \in (S_\infty)^p, \\ & x \in X \cap \mathbb{Z}^{n_d}, y \in Y, \alpha \in \mathbb{R}, \end{cases}
$$

where t^j and a_i^k were defined in (3.7) and (3.8), respectively. The presentation of the inexact OA algorithm (given next) and the proof of its finiteness in Theorem 3.5.1 follows the lines in [27].

Algorithm 3.5.1 (Inexact Outer Approximation).

Initialization

Let x^0 be given. Set $p = 0, T^{-1} = \emptyset, (S_\infty)^{-1} = \emptyset$, and $\mathrm{UBD} = +\infty$.

REPEAT

1. Inexactly solve the subproblem $\mathrm{NLP}(x^p)$, or the feasibility subproblem $\mathrm{NLPF}_\infty(x^p)$ provided $\mathrm{NLP}(x^p)$ is infeasible, and let y^p be an approximate optimal solution. At the same time, obtain the corresponding inexact Lagrange multipliers λ^p of $\mathrm{NLP}(x^p)$ (resp. μ^p of $\mathrm{NLPF}_\infty(x^p)$). Evaluate the residuals r^p and s^p of $\mathrm{NLP}(x^p)$ (resp. v^p, w^p, and z^p of $\mathrm{NLPF}_\infty(x^p)$).

2. Linearize the objective functions and constraints at (x^p, y^p). Renew $T^p = T^{p-1} \cup \{p\}$ or $(S_\infty)^p = (S_\infty)^{p-1} \cup \{p\}$.

3. If $\mathrm{NLP}(x^p)$ is feasible and $f(x^p, y^p) < \mathrm{UBD}$, then update current best point by setting $\bar{x} = x^p, \bar{y} = y^p$, and $\mathrm{UBD} = f(x^p, y^p)$.

4. Solve the relaxed master problem $(\mathcal{P}^{\mathrm{OA}})^p$, obtaining a new discrete assignment x^{p+1} to be tested in the algorithm. Increment p by one unit.

UNTIL $((\mathcal{P}^{\text{OA}})^p$ is infeasible).

If termination occurs with UBD $= +\infty$, then the algorithm visited every discrete assignment $x \in X \cap \mathbb{Z}^{n_d}$ but did not obtain a feasible point for the original MINLP problem P, or perturbed version $\mathcal{P}$. In this case, the MINLP is declared infeasible. Next, we will show that the inexact OA algorithm also terminates in a finite number of steps.

Theorem 3.5.1. *Let* P *be the convex* MINLP *problem (1.1) and* $\mathcal{P}$ *be its perturbed problem as defined in (3.6). Suppose that the classification of feasibility for the* NLP *subproblems* NLP(x^j) *is made exactly. Assume that either* P *has a finite optimal value or is infeasible, and that the residuals of the* KKT *conditions of the* NLP *subproblems satisfy Assumptions 3.3.1 and 3.3.2. Then Algorithm 3.5.1 terminates in a finite number of steps at an optimal solution of* $\mathcal{P}$ *or with an indication that* $\mathcal{P}$ *is infeasible.*

Proof. Since the set X is bounded by assumption, finite termination of Algorithm 3.5.1 will be established by proving that no discrete assignment is generated twice by the algorithm.

Let $q \le p$. If $q \in (S_\infty)^p$, it has been shown in Part I of the proof of Theorem 3.4.1 that the corresponding constraint in $\mathcal{P}^{\text{OA}}_{x^p}$, derived from the feasibility problem NLPF$_\infty(x^q)$, cannot be satisfied, showing that x^q cannot be feasible for $(\mathcal{P}^{\text{OA}})^p$.

We will now show that x^q cannot be feasible for $(\mathcal{P}^{\text{OA}})^p$ when $q \in T^p$. For this purpose, let us assume that x^q is feasible in $(\mathcal{P}^{\text{OA}})^p$ and try to reach a contradiction. Let y^q be an approximate optimal solution of NLP(x^q) satisfying the inexact KKT conditions, that is, there exist $\lambda^q \in \mathbb{R}^m_+, r^q \in \mathbb{R}^{n_c}$, and $s^q \in \mathbb{R}^m$, such that

$$\nabla_y f^q + \sum_{i=1}^{m} \lambda_i^q \nabla_y g_i(x^q, y^q) \;=\; r^q, \tag{3.22}$$

$$\lambda_i^q g_i(x^q, y^q) \;=\; s_i^q, \quad i = 1, \dots, m. \tag{3.23}$$

If x^q would be feasible for $(\mathcal{P}^{\text{OA}})^p$ it would satisfy the following set of inequalities for some y:

$$\alpha^p \;<\; \text{UBD}^p \;\le\; f^q, \tag{3.24}$$

$$\begin{pmatrix} \nabla_x f^q \\ \nabla_y f^q - r^q \end{pmatrix}^\top \begin{pmatrix} 0 \\ y - y^q \end{pmatrix} + f^q \;\le\; \alpha^p, \tag{3.25}$$

$$(\nabla g^q)^\top \begin{pmatrix} 0 \\ y - y^q \end{pmatrix} + g^q \;\le\; t^q, \tag{3.26}$$

where, for $i = 1, \dots, m$,

$$t_i^q \;=\; \begin{cases} \frac{s_i^q}{\lambda_i^q}, & \text{if } \lambda_i^q \;>\; 0, \\ 0, & \text{if } \lambda_i^q \;=\; 0. \end{cases}$$

Multiplying the rows in (3.26) by the Lagrange multipliers $\lambda_i^q \geq 0$, $i = 1, \ldots, m$, and adding (3.25), we obtain that

$$(\nabla_y f^q - r^q)^\top (y - y^q) + f^q + \sum_{i=1}^{m} \lambda_i^q \nabla_y g_i(x^q, y^q)^\top (y - y^q) + \sum_{i=1}^{m} \lambda_i^q g_i^q \;\leq\; \alpha^p + \sum_{i=1}^{m} \lambda_i^q t_i^q,$$

which, by the definition of t^q, is equivalent to

$$(\nabla_y f^q - r^q)^\top (y - y^q) + f^q + \sum_{i=1}^{m} \lambda_i^q \nabla_y g_i(x^q, y^q)^\top (y - y^q) + \sum_{i=1}^{m} (\lambda_i^q g_i^q - s_i^q) \;\leq\; \alpha^p.$$

The left hand side of this inequality can be written as:

$$\left[\nabla_y f^q - r^q + \sum_{i=1}^{m} \lambda_i^q \nabla_y g_i(x^q, y^q)\right]^\top (y - y^q) + \sum_{i=1}^{m} (\lambda_i^q g_i^q - s_i^q) + f^q.$$

Using (3.22) and (3.23), this is equal to f^q and therefore we obtain the inequality

$$f^q \;\leq\; \alpha^p,$$

which contradicts (3.24).

The rest of the proof is exactly as in [27, Theorem 2] but we repeat here for completeness and possible changes in notation. Finally, we will show that Algorithm 3.5.1 always terminates at a solution of $\mathcal{P}$ or with an indication that $\mathcal{P}$ is infeasible (which occurs when UBD $= +\infty$ at the exit). If $\mathcal{P}$ is feasible, then let (x^*, y^*) be an optimal solution of $\mathcal{P}$ with optimal value f^*. Without loss of generality, we will not distinguish between (x^*, y^*) and any other optimal solution with the same objective value f^*. Note that from Theorem 3.4.1, (x^*, y^*, f^*) is also an optimal solution of $\mathcal{P}^{\mathrm{OA}}$. Now assume that the algorithm terminates indicating a non-optimal point (x', y') with $f' > f^*$. In such a situation, the previous relaxation of the master problem $\mathcal{P}^{\mathrm{OA}}$ after adding the constraints at the point (x', y', f'), called $(\mathcal{P}^{\mathrm{OA}})^p$, is infeasible, causing the above mentioned termination. We will get a contradiction by showing that (x^*, y^*, f^*) is feasible for $(\mathcal{P}^{\mathrm{OA}})^p$. First, by the assumption that UBD $= f' > f^*$, the first constraint $\alpha = f^* <$ UBD of $(\mathcal{P}^{\mathrm{OA}})^p$ holds. Secondly, since (x^*, y^*, f^*) is an optimal solution to $\mathcal{P}^{\mathrm{OA}}$, it must be feasible for all other constraints of $(\mathcal{P}^{\mathrm{OA}})^p$. Therefore, the algorithm could not terminate at (x', y') with UBD $= f'$. □

Chapter 4

Inexact generalized Benders decomposition

In this chapter, similarly to OA in Chapter 3, we generalize GBD to the case where the NLP subproblems are solved inexactly, rederiving the corresponding background theory and main algorithm. Moreover, we discuss the relationship between the inexact forms of OA and GBD. At last, we show that the inexact MILP master problem can also be derived in the inexact case from a perturbed duality representation of the original, perturbed problem $\mathcal{P}$ in (3.6). In this chapter, we will use the definitions and assumptions given in Sections 3.1 and 3.3.

4.1 Equivalence between perturbed and master problems for GBD

In the generalized Benders decomposition (GBD), the MILP master problem involves only the discrete variables. When considering the inexact case, the master problem of GBD is the following:

$$\mathcal{P}^{\mathrm{GBD}} \begin{cases} \min & \alpha \\ \text{s.t.} & f(x^j, y^j) + \nabla_x f(x^j, y^j)^\top (x - x^j) + \sum_{i=1}^m \lambda_i^j \nabla_x g_i(x^j, y^j)^\top (x - x^j) \ \leq \ \alpha, \\ & \forall j \in T, \\ & \sum_{i=1}^m \mu_i^k [g_i(x^k, y^k) + \nabla_x g_i(x^k, y^k)^\top (x - x^k)] + w^k u^k - \sum_{i=1}^m z_i^k \ \leq \ 0, \\ & \forall k \in S_\infty, \\ & x \in X \cap \mathbb{Z}^{n_d}, \alpha \in \mathbb{R}, \end{cases}$$

where, remember, the pair (y^j, λ^j) satisfies the inexact KKT conditions (3.1)–(3.2) and the pair (y^k, μ^k) satisfies the inexact KKT conditions (3.3)–(3.5), with residuals (r^j, s^j) and (v^k, w^k, z^k), respectively. One can easily recognize the classical form of (exact) GBD master problem $\mathrm{P}^{\mathrm{GBD}}$ in Section 2.3 when $w^k = 0$ and $z^k = 0$.

A proof similar to the one of exact GBD and exact and inexact OA (Theorem 3.4.1) allows us to establish the desired equivalence between the original, perturbed MINLP problem and the MILP master problem $\mathcal{P}^{\mathrm{GBD}}$.

Theorem 4.1.1. *Let* P *be the convex* MINLP *problem (1.1) and* $\mathcal{P}$ *be its perturbed problem as defined in (3.6). Suppose that the classification of feasibility for the* NLP *subproblems* $\mathrm{NLP}(x^j)$ *is made exactly. Assume that* P *is feasible with a finite optimal value and that the residuals of the* KKT *conditions of the* NLP *subproblems satisfy Assumptions 3.3.1 and 3.3.2. Then* $\mathcal{P}^{\mathrm{GBD}}$ *and* $\mathcal{P}$ *have the same optimal value.*

Proof. Given any $x^l \in X \cap \mathbb{Z}^{n_d}$ with $l \in T \cup S_\infty$, let $\mathcal{P}^{\mathrm{GBD}}_{x^l}$ denote the problem in α obtained from $\mathcal{P}^{\mathrm{GBD}}$ when x is fixed to x^l. First we will prove that problem $\mathcal{P}^{\mathrm{GBD}}_{x^k}$ is infeasible for every $k \in S_\infty$. When we set $x = x^k$, in the corresponding constraint of $\mathcal{P}^{\mathrm{GBD}}$, we obtain

$$\sum_{i=1}^{m} \mu_i^k g_i(x^k, y^k) + w^k u^k - \sum_{i=1}^{m} z_i^k \ \leq \ 0.$$

From (3.4) and (3.5), it results that $u^k \leq 0$, but one knows that u^k is strictly positive when $\mathrm{NLP}(x^k)$ is infeasible.

Next, we will prove that for each $x^j \in X \cap \mathbb{Z}^{n_d}$, with $j \in T$, $\mathcal{P}^{\mathrm{GBD}}_{x^j}$ has the same optimal value as the perturbed $\mathrm{NLP}(x^j)$. First, we will prove that the following constraints of $\mathcal{P}^{\mathrm{GBD}}_{x^j}$

$$f(x^l, y^l) + \nabla_x f(x^l, y^l)^\top (x^j - x^l) + \sum_{i=1}^{m} \lambda_i^l \nabla_x g_i(x^l, y^l)^\top (x^j - x^l) \ \leq \ \alpha, \ \forall l \in T, \quad (4.1)$$

$$\sum_{i=1}^{m} \mu_i^k [g_i(x^k, y^k) + \nabla_x g_i(x^k, y^k)^\top (x^j - x^k)] + w^k u^k - \sum_{i=1}^{m} z_i^k \ \leq \ 0, \ \forall k \in S_\infty \quad (4.2)$$

are satisfied with $\alpha = f(x^j, y^j)$. Under Assumptions 3.3.1–3.3.2, we know from the proof of Theorem 3.4 (Part II–A) that the following hold: (3.14) with $y = y^j$ and $\alpha = f(x^j, y^j)$, (3.15) with $y = y^j$, and (3.17) with $y = y^j$.

When $l \in T$, multiplying the inequalities (3.15) with $y = y^j$ by the non-negative multipliers $\lambda_1^l, \ldots, \lambda_m^l$ and summing them together with (3.14) with $y = y^j$ and $\alpha = f(x^j, y^j)$, one obtains

$$
\begin{aligned}
& f(x^l, y^l) + \nabla_x f(x^l, y^l)^\top (x^j - x^l) + \sum_{i=1}^m \lambda_i^l \nabla_x g_i(x^l, y^l)^\top (x^j - x^l) \\
& \leq f(x^j, y^j) - \left[\nabla_y f(x^l, y^l) + \sum_{i=1}^m \lambda_i^l \nabla_y g_i(x^l, y^l) - r^l \right]^\top (y^j - y^l) \\
& \quad - \sum_{i=1}^m \lambda_i^l g(x^l, y^l) + \sum_{i=1}^m \lambda_i^l t_i^l.
\end{aligned}
$$

The right hand side is equal to $f(x^j, y^j)$ by the definitions of r^l, s^l, and t^l, showing that (4.1) holds with $\alpha = f(x^j, y^j)$.

When $k \in S_\infty$, multiplying the inequalities in (3.17) with $y = y^j$ by the non-negative multipliers $\mu_1^k, \ldots, \mu_m^k$, and summing them up, one obtains, using (3.3) and (3.4),

$$
\sum_{i=1}^m \mu_i^k \nabla_x g_i(x^k, y^k)^\top (x^j - x^k) + \sum_{i=1}^m \mu_i^k g_i(x^k, y^k) \ \leq \ \sum_{i=1}^m \mu_i^k a_i^k,
$$

which, by the definition of a^k, is the same as (4.2).

Thus, $f(x^j, y^j)$ is a feasible point of $\mathcal{P}_{x^j}^{\mathrm{GBD}}$, and therefore $f(x^j, y^j)$ is an upper bound on the optimal value $\bar{\alpha}^j$ of $\mathcal{P}_{x^j}^{\mathrm{OA}}$. To show that is also a lower bound, i.e., that $\bar{\alpha}^j \geq f(x^j, y^j)$, note that from (4.1), when $l = j$, $\mathcal{P}_{x^j}^{\mathrm{GBD}}$ contains the constraint:

$$
f(x^j, y^j) \ \leq \ \alpha.
$$

We have thus proved that for any $x^j \in X \cap \mathbb{Z}^{n_d}$, with $j \in T$, problems $\mathcal{P}_{x^j}^{\mathrm{GBD}}$ and perturbed NLP(x^j) have the same optimal value, which concludes the proof. $\square$

It is also easy to see that when the residuals are zero, we recover Theorem 2.3.1 in Chapter 2. In addition we have the following remark.

Remark 4.1.1. *It is well known, in the convex case, that the constraints of the* GBD *master problem can be derived from the corresponding ones of the* OA *master problem, a fact reported already in Property 2.3.1 for the exact case. The same happens naturally in the inexact case. In fact, from the proof of Theorem 4.1.1 above, we can see that the constraints in $\mathcal{P}_{x^j}^{\mathrm{OA}}$, for $j \in T$, imply the corresponding ones in $\mathcal{P}_{x^j}^{\mathrm{GBD}}$. Moreover, one can easily see that any of the constraints in $\mathcal{P}^{\mathrm{OA}}$ imply the corresponding ones in $\mathcal{P}^{\mathrm{GBD}}$.*

Thus, one can also say in the inexact case that the lower bounds produced iteratively by the OA *algorithm are stronger than the ones provided by the corresponding* GBD *algorithm (given next).*

4.2 Inexact GBD algorithm

As we know for exact GBD, it is possible to derive an algorithm for the inexact case, terminating finitely, by solving at each iteration a relaxed MILP formed by the cuts collected so far. The definitions of UBD^p, T^p, and $(S_\infty)^p$ are the same as those in Section 3.5. The relaxed MILP to be solved at each iteration is thus given by

$$(\mathcal{P}^{\mathrm{GBD}})^p \begin{cases} \min \quad \alpha \\ \text{s.t.} \quad \alpha \; < \; \mathrm{UBD}^p \\ \qquad f(x^j,y^j) + \nabla_x f(x^j,y^j)^\top (x-x^j) + \sum_{i=1}^m \lambda_i^j \nabla_x g_i(x^j,y^j)^\top (x-x^j) \; \leq \; \alpha, \\ \qquad \forall j \in T^p \\ \qquad \sum_{i=1}^m \mu_i^k [g_i(x^k,y^k) + \nabla_x g_i(x^k,y^k)^\top (x-x^k)] + w^k u^k - \sum_{i=1}^m z_i^k \; \leq \; 0, \\ \qquad \forall k \in (S_\infty)^p \\ \qquad x \in X \cap \mathbb{Z}^{n_d}, \alpha \in \mathbb{R}. \end{cases}$$

The inexact GBD algorithm is given next (and follows the presentation in [27] for OA).

Algorithm 4.2.1 (Inexact GBD Approximation).

Initialization

Let x^0 be given. Set $p=0, T^{-1}=\emptyset, (S_\infty)^{-1}=\emptyset$, and $\mathrm{UBD}=+\infty$.

REPEAT

1. Inexactly solve the subproblem $\mathrm{NLP}(x^p)$, or the feasibility subproblem $\mathrm{NLPF}_\infty(x^p)$ provided $\mathrm{NLP}(x^p)$ is infeasible, and let y^p be an approximate optimal solution. At the same time, obtain the corresponding inexact Lagrange multipliers λ^p of $\mathrm{NLP}(x^p)$ (resp. μ^p of $\mathrm{NLPF}_\infty(x^p)$). Evaluate the residuals r^p and s^p of $\mathrm{NLP}(x^p)$ (resp. v^p, w^p, and z^p of $\mathrm{NLPF}_\infty(x^p)$).

2. Linearize the objective functions and constraints at x^p. Renew $T^p = T^{p-1} \cup \{p\}$ or $(S_\infty)^p = (S_\infty)^{p-1} \cup \{p\}$.

3. If $\mathrm{NLP}(x^p)$ is feasible and $f^p < \mathrm{UBD}$, then update current best point by setting $\bar{x} = x^p, \bar{y} = y^p$, and $\mathrm{UBD} = f^p$.

4. Solve the relaxed master problem $(\mathcal{P}^{\mathrm{GBD}})^p$, obtaining a new discrete assignment x^{p+1} to be tested in the algorithm. Increment p by one unit.

UNTIL $((\mathcal{P}^{\mathrm{GBD}})^p$ is infeasible).

Similarly as in Theorem 3.5.1 for OA, one can establish that the above inexact GBD algorithm terminates in a finite number of steps.

Theorem 4.2.1. *Let* P *be the convex* MINLP *problem (1.1) and* $\mathcal{P}$ *be its perturbed problem as defined in (3.6). Suppose that the classification of feasibility for the* NLP *subproblems* NLP(x^j) *is made exactly. Assume that either* P *has a finite optimal value or is infeasible, and that the residuals of the* KKT *conditions of the* NLP *subproblems satisfy Assumptions 3.3.1 and 3.3.2. Then Algorithm 4.2.1 terminates in a finite number of steps at an optimal solution of* $\mathcal{P}$ *or with an indication that* $\mathcal{P}$ *is infeasible.*

4.3 Derivation of the master problem for inexact GBD

As in the exact case, the MILP master problem $\mathcal{P}^{\mathrm{GBD}}$ can be derived from a more general master problem closer to the original duality motivation of GBD:

$$\mathcal{P}^{\mathrm{GBD}}_{original}\begin{cases} \min & \alpha \\ \text{s.t.} & \inf_{y\in Y}\left\{f(x,y)+(\lambda^j)^\top g(x,y)-(r^j)^\top(y-y^j)\right\}-\sum_{i=1}^m s_i^j \ \le\ \alpha,\ \forall j\in T, \\ & \inf_{y\in Y}\left\{(\mu^k)^\top g(x,y)-(v^k)^\top(y-y^k)\right\}+w^k u^k-\sum_{i=1}^m z_i^k \ \le\ 0,\ \forall k\in S_\infty, \\ & x\in X\cap\mathbb{Z}^{n_d},\alpha\in\mathbb{R}. \end{cases}$$

It is easy to recognize the classical form of (exact) GBD master problem $\mathrm{P}^{\mathrm{GBD}}_{original}$ in Section 2.3 when $s^j=0$, $w^k=0$, and $z^k=0$. In fact, we will show next that the constraints in problem $\mathcal{P}^{\mathrm{GBD}}_{original}$ imply those of $\mathcal{P}^{\mathrm{GBD}}$.

When $l\in T$, one knows that NLP(x^l) has an approximate optimal solution y^l, satisfying the corresponding inexact KKT conditions with inexact Lagrange multipliers λ^l. By the convexity of f and g (see (3.9) and (3.10)),

$$\begin{aligned} f(x,y)+(\lambda^l)^\top g(x,y)-(r^l)^\top(y-y^l) \ \ge\ & f(x^l,y^l)+\nabla_x f(x^l,y^l)^\top(x-x^l) \\ & +\nabla_y f(x^l,y^l)^\top(y-y^l) \\ & +\sum_{i=1}^m \lambda_i^l\left[g_i(x^l,y^l)+\nabla_x g_i(x^l,y^l)^\top(x-x^l)\right. \\ & +\left.\nabla_y g_i(x^l,y^l)^\top(y-y^l)\right]-(r^l)^\top(y-y^l). \end{aligned}$$

Thus, using the inexact KKT conditions (3.1),

$$\begin{aligned}
\alpha &\geq \inf_{y\in Y}\left\{f(x,y)+(\lambda^l)^\top g(x,y)-(r^l)^\top(y-y^l)\right\}-\sum_{i=1}^m s_i^l \\
&\geq \inf_{y\in Y}\left\{f(x^l,y^l)+\nabla_x f(x^l,y^l)^\top(x-x^l)+\sum_{i=1}^m \lambda_i^l\nabla_x g_i(x^l,y^l)^\top(x-x^l)+\sum_{i=1}^m \lambda_i^l g_i(x^l,y^l)\right\} \\
&\quad -\sum_{i=1}^m s_i^l \\
&= f(x^l,y^l)+\nabla_x f(x^l,y^l)^\top(x-x^l)+\sum_{i=1}^m \lambda_i^l\nabla_x g_i(x^l,y^l)^\top(x-x^l)+\sum_{i=1}^m \lambda_i^l g_i(x^l,y^l) \\
&\quad -\sum_{i=1}^m s_i^l \\
&= f(x^l,y^l)+\nabla_x f(x^l,y^l)^\top(x-x^l)+\sum_{i=1}^m \lambda_i^l\nabla_x g_i(x^l,y^l)^\top(x-x^l).
\end{aligned}$$

The last equality holds due to (3.2).

When $l\in S_\infty$, we know that $\mathrm{NLPF}_\infty(x^l)$ has an approximate optimal solution y^l satisfying the corresponding inexact KKT conditions with inexact Lagrange multipliers μ^l. Also by the convexity of g (see (3.10)), we have that

$$\begin{aligned}
(\mu^l)^\top g(x,y)-(v^l)^\top(y-y^l) &\geq (\mu^l)^\top\left[g(x^l,y^l)+\nabla_x g(x^l,y^l)^\top(x-x^l)\right] \\
&\quad +\left(\sum_{i=1}^m \mu_i^l\nabla_y g_i(x^l,y^l)-v^l\right)^\top(y-y^l).
\end{aligned}$$

Then, using the inexact KKT conditions (3.4),

$$\begin{aligned}
0 &\geq \inf_{y\in Y}\left\{(\mu^l)^\top g(x,y)-(v^l)^\top(y-y^l)\right\}+w^l u^l-\sum_{i=1}^m z_i^l \\
&\geq \inf_{y\in Y}\left\{\sum_{i=1}^m \mu_i^l[g_i(x^l,y^l)+\nabla_x g_i(x^l,y^l)^\top(x-x^l)]\right\}+w^l u^l-\sum_{i=1}^m z_i^l \\
&= \sum_{i=1}^m \mu_i^l\left[g_i(x^l,y^l)+\nabla_x g_i(x^l,y^l)^\top(x-x^l)\right]+w^l u^l-\sum_{i=1}^m z_i^l.
\end{aligned}$$

In summary we have the following property.

Property 4.3.1. *Given some sets T and S_∞, the lower bound predicted by the master problem $\mathcal{P}^{\mathrm{GBD}}_{original}$ is greater than or equal to the one predicted by the master problem $\mathcal{P}^{\mathrm{GBD}}$.*

Chapter 5

Numerical experiments

We will illustrate some of the practical features of inexact OA and GBD algorithms by reporting numerical results on the test set of AMPL problems described in Table 5.1. All the problems are convex in the continuous variables. The first three problems are linear in the discrete variables, and consist of simplified versions of process synthesis problems [24].

The implementation and testing of Algorithms 3.5.1 and 4.2.1 were made in MATLAB (version 7.9.0, R2009b). In both algorithms, we used the MATLAB function `fmincon` to solve the NLP subproblems, and the function `cplexmilp` from CPLEX [2] (version 12.4 called from MATLAB) to solve the MILP problems.

The linear equality constraints possibly present in the original problems (as well as the bounds in the variables) were kept in the MILP relaxed master problems. The constraint $\alpha < \mathrm{UBD}^p$ was implemented as $\alpha \leq \mathrm{UBD}^p - 10^{-5}$. (Slight variations of the value 10^{-5} change the results but do not affect their pattern and the conclusions that can be drawn.)

For both methods, we report results for two variants, depending on the form of the cuts. In the first variant (`e.cuts`), the cuts are the exact ones, while in the second variant (`i.cuts`), the cuts are the inexact ones. In both variants, the NLP subproblems are solved inexactly, with a tolerance (for function values in `fmincon`) varying dynamically. In the first two iterations ($p = 0, 1$), we set this tolerance to 10^{-2}, and then possibly decreased it according to the absolute reduction in the upper bound UBD, as follows:

$$\min\left\{10^{-2}, \max\{10^{-6}, 10^{-\theta} \times |\mathrm{UBD}^{p-1} - \mathrm{UBD}^{p-2}|\}\right\}, \quad p \geq 2 \tag{5.1}$$

with $\theta = 5$ (different values of θ did not change the results significantly, neither taking a relative reduction on the upper bounds instead). To ensure the achievement of feasibility in the solution of the NLP subproblems, we set to 10^{-6} the tolerance corresponding to the constraint violation in `fmincon` (and we recall here that our theory does not cover a misclassification of feasibility).

Table 5.1: The number of variables and constraints, and the optimal values of all tested problems. The number of constraints include linear equalities and inequalities and nonlinear inequalities. The AMPL code for Problems 1–6 was taken from the MacMINLP collection [43] and for Problems 7–19 from the Open Source CMU-IBM Project [1].

Problem	n_c	n_d	# of constraints	optimal value f^*
synthes1	3	3	6	6.01
synthes2	6	5	14	73.04
synthes3	9	8	23	68.01
batch	22	24	69	285506.51
trimloss2	6	31	24	5.30
optprloc	5	25	29	-8.06
CLay0203H	72	18	132	3760.00
CLay0203M	12	18	54	41573.26
CLay0204H	132	32	234	6545.00
CLay0204M	20	32	90	6545.00
CLay0205M	30	50	135	8092.50
CLay0303H	78	21	150	3760.00
CLay0303M	12	21	66	26669.11
CLay0304H	140	36	258	6920.00
FLay03H	110	12	144	48.99
FLay03M	14	12	24	48.99
FLay04M	18	24	42	54.41
fo7-2	72	42	211	17.74
Syn10M04M	140	80	516	-4557.06

In the tables of results we report the number N of iterations taken by Algorithms 3.5.1 and 4.2.1. We also report, in the rows corresponding to the variant `i.cuts`, the number C of constraint inequalities of Assumptions 3.3.1 and 3.3.2 that were violated by more than 10^{-8} during the course of the algorithm, i.e., for all discrete assignments considered by each application of the algorithm. (The maximum number for such a C when solving problems synthes1, synthes2, and synthes3, given that no infeasible case occurred for these problems for both OA and GBD, is $N(N-1)(1+c)/2$, where $c = 12$ (synthes1), $c = 26$ (synthes2), and $c = 41$ (synthes3).) We should point out here again that Assumptions 3.3.1 and 3.3.2 refer to all possible discrete assignments and not only to those generated by an application of the algorithm, but such a surrogate number will provide us a good indication of the potential violation of these assumptions.

The stopping criteria of both algorithms consisted of the corresponding relaxed master program being infeasible, or the number of iterations exceeding 50 (MAX in the tables), or the solution of the MILP relaxed master program repeating a previous one (upper script ^{b} in the tables). The upper script ^{a} denotes the cases where the algorithms did not stop at an optimal value.

5.1 Results for the inexact OA method

Tables 5.2–5.3 summarize the application of inexact OA (Algorithm 3.5.1) (variant inexact solution of NLP subproblems and exact cuts, `e.cuts`, and variant inexact solution of NLP subproblems and inexact cuts, `i.cuts`) to our test set. For problems synthes2 and synthes3 in Table 5.2 and problems CLay0203H, CLay0303H, CLay0304H, and FLay04M in Table 5.3, the variant `e.cuts` entered in cycle, repeating the solution of the MILP, indicating that the inclusion of inexactness in the cuts renders OA more robust. Moreover, looking at the results of both tables, one can see that the number of iterations taken by `i.cuts` is in general smaller than in `e.cuts`, except for problem CLay0204H. We also observe that inexact OA converged in most of the cases even neglecting the imposition of the inequalities of Assumptions 3.3.1 and 3.3.2.

We note that the inexact tolerance is changing dynamically along the iterations (according to (5.1)) in both inexact methods (OA and GBD) and for both variants (`e.cuts` and `i.cuts`). Tables 5.4 and 5.5 present the outcome of the formula (5.1) for two given problems and the `i.cuts` version.

Table 5.2: Application of inexact OA to problems from the MacMINLP collection [43]. The table reports the number N of iterations taken as well as the number C of inequalities found to violate Assumptions 3.3.1 and 3.3.2.

Problem	Cuts	N	C
synthes1	`e.cuts`	3	-
synthes1	`i.cuts`	3	0
synthes2	`e.cuts`	5^b	-
synthes2	`i.cuts`	4	0
synthes3	`e.cuts`	5^b	-
synthes3	`i.cuts`	4	1
batch	`e.cuts`	4	-
batch	`i.cuts`	3	3
trimloss2	`e.cuts`	8	-
trimloss2	`i.cuts`	8	0
optprloc	`e.cuts`	3	-
optprloc	`i.cuts`	3	0

5.2 Results for the inexact GBD method

Tables 5.6–5.7 summarize the application of inexact GBD (Algorithm 4.2.1) (variant inexact solution of NLP subproblems and exact cuts, `e.cuts`, and variant inexact solution of NLP subproblems and inexact cuts, `i.cuts`). There is perhaps less difference between the two variants compared to OA, for which a possible explanation is the fact that, for GBD, the 'exact' cuts in the variant `e.cuts` already incorporate inexact information coming from the inexact Lagrange multipliers. As in OA, there is more tendency to enter in a cycle for `e.cuts`, repeating the solution of the MILP, than for `i.cuts`. One can see that the `e.cuts` variant stopped without finding a feasible point, entering in cycle, for problems batch, trimloss2, and CLay0204H, which did not happen for the variant `i.cuts`. One observes that inexact GBD takes more iterations than inexact OA in these problems, which, according to Remark 4.1.1, might be expected since inexact GBD yields weaker lower bounds and has been observed to take more major iterations to converge than OA (see [32, 53]). The number of inequalities of Assumptions 3.3.1 and 3.3.2 violated in inexact GBD is also higher than the one in inexact OA.

Table 5.3: Application of inexact OA to problems from the Open Source CMU-IBM Project [1]. The table reports the number N of iterations taken as well as the number C of inequalities found to violate Assumptions 3.3.1 and 3.3.2.

Problem	Cuts	N	C
CLay0203H	`e.cuts`	5^b	-
CLay0203H	`i.cuts`	5	3
CLay0203M	`e.cuts`	11	-
CLay0203M	`i.cuts`	11	8
CLay0204H	`e.cuts`	2^a (6885.00)	-
CLay0204H	`i.cuts`	9^a (6885.00)	2
CLay0204M	`e.cuts`	4	-
CLay0204M	`i.cuts`	4	6
CLay0205M	`e.cuts`	7	-
CLay0205M	`i.cuts`	7	6
CLay0303H	`e.cuts`	5^b	-
CLay0303H	`i.cuts`	2	0
CLay0303M	`e.cuts`	11	-
CLay0303M	`i.cuts`	11	0
CLay0304H	`e.cuts`	7^b	-
CLay0304H	`i.cuts`	2	0
FLay03H	`e.cuts`	11^b	-
FLay03H	`i.cuts`	6^b	5
FLay03M	`e.cuts`	8	-
FLay03M	`i.cuts`	8	13
FLay04M	`e.cuts`	31^b	-
FLay04M	`i.cuts`	28	103
fo7-2	`e.cuts`	6	-
fo7-2	`i.cuts`	6	30
Syn10M04M	`e.cuts`	6	-
Syn10M04M	`i.cuts`	3	1

Table 5.4: The tolerance chosen in every iteration according to (5.1) for problem FLay03H (inexact OA and inexact cuts). For $p = 0, 1$, we always set the tolerance to 10^{-2}.

Iteration	Tolerance
p=0	1×10^{-2}
p=1	1×10^{-2}
p=2	4.80×10^{-4}
p=3	1×10^{-2}
p=4	1×10^{-2}
p=5	1×10^{-2}

Table 5.5: The tolerance chosen in every iteration according to (5.1) for problem CLay0203M (inexact OA and inexact cuts). For $p = 0, 1$, we always set the tolerance to 10^{-2}.

Iteration	Tolerance
p=0	1×10^{-2}
p=1	1×10^{-2}
p=2	1×10^{-2}
p=3	1×10^{-6}
p=4	1×10^{-6}
p=5	1×10^{-6}
p=6	1×10^{-6}
p=7	1×10^{-2}
p=8	1×10^{-2}
p=9	1×10^{-2}
p=10	1×10^{-2}

Table 5.6: Application of inexact GBD to problems from the MacMINLP collection [43]. The table reports the number N of iterations taken as well as the number C of inequalities found to violate Assumptions 3.3.1 and 3.3.2.

Problem	Cuts	N	C
synthes1	`e.cuts`	4	-
synthes1	`i.cuts`	4	0
synthes2	`e.cuts`	9	-
synthes2	`i.cuts`	9	0
synthes3	`e.cuts`	10	-
synthes3	`i.cuts`	10	19
batch	`e.cuts`	1 (Inf)	-
batch	`i.cuts`	3^a(398558.84)	0
trimloss2	`e.cuts`	16^b (Inf)	-
trimloss2	`i.cuts`	22	0
optprloc	`e.cuts`	MAX	-
optprloc	`i.cuts`	MAX	21

Table 5.7: Application of inexact GBD to problems from the Open Source CMU-IBM Project [1]. The table reports the number N of iterations taken as well as the number C of inequalities found to violate Assumptions 3.3.1 and 3.3.2.

Problem	Cuts	N	C
CLay0203H	`e.cuts`	$14^{a,b}(5240.00)$	-
CLay0203H	`i.cuts`	$42^{a}(4000.00)$	330
CLay0203M	`e.cuts`	42	-
CLay0203M	`i.cuts`	42	155
CLay0204H	`e.cuts`	4^{b} (Inf)	-
CLay0204H	`i.cuts`	$28^{a}(10265.00)$	301
CLay0204M	`e.cuts`	MAX	-
CLay0204M	`i.cuts`	MAX	329
CLay0205M	`e.cuts`	MAX	-
CLay0205M	`i.cuts`	MAX	192
CLay0303H	`e.cuts`	4^{b} (Inf)	-
CLay0303H	`i.cuts`	6 (Inf)	0
CLay0303M	`e.cuts`	MAX	-
CLay0303M	`i.cuts`	MAX	93
CLay0304H	`e.cuts`	$21^{a,b}(9440.00)$	-
CLay0304H	`i.cuts`	$20^{a}(9840.00)$	561
FLay03H	`e.cuts`	11^{b}	-
FLay03H	`i.cuts`	11^{b}	14
FLay03M	`e.cuts`	44	-
FLay03M	`i.cuts`	44	161
FLay04M	`e.cuts`	$5^{a,b}(54.99)$	-
FLay04M	`i.cuts`	MAX	215
fo7-2	`e.cuts`	8^{b} (Inf)	-
fo7-2	`i.cuts`	34 (Inf)	0
Syn10M04M	`e.cuts`	$7^{a,b}(-31.15)$	-
Syn10M04M	`i.cuts`	$7^{a}(-31.15)$	10

Chapter 6

The case of inexact multipliers

In Chapters 3 and 4, we have analyzed the OA and GBD methods when the corresponding NLP subproblems are solved inexactly. We have assumed there that the approximate solutions of these subproblems satisfied an inexact form of the corresponding first-order necessary KKT conditions (see (3.1)–(3.5)). Note that the Lagrange multipliers appearing in these conditions were assumed to be non-negative, i.e., that no inexactness was considered in the non-negativity of the Lagrange multipliers. However, when we solve the NLP subproblems inexactly, depending on the solver chosen, the approximate Lagrange multipliers returned may not be non-negative, by a small residual amount. In this chapter, we will show how we can generalize the approaches of Chapters 3 and 4 to consider inexactness in the non-negativity of the Lagrange multipliers of the NLP subproblems.

Given any vector $v \in \mathbb{R}^n$, let M_v denote the index set of its negative elements,

$$M_v \;=\; \{i : v_i \;<\; 0, i \;=\; 1, \ldots, n\},$$

and P_v its complement,

$$P_v \;=\; \{1, \ldots, n\} \setminus M_v.$$

The MINLP problem considered in this chapter was defined in (1.1) and also assumed convex. The respective NLP subproblems were defined in Section 3.1. Recall that given any element $x^j \in X \cap \mathbb{Z}^{n_d}$, if NLP$(x^j)$ is feasible, y^j denotes an approximate optimal solution of NLP(x^j) satisfying an inexact form of the corresponding KKT conditions. Here, we assume that there exists a vector of inexact multipliers $\lambda^j \in \mathbb{R}^m$ (not necessarily non-negative) and vectors of residuals $r^j \in \mathbb{R}^{n_c}$ and $s^j \in \mathbb{R}^m$ such that the following equations hold:

$$\nabla_y f(x^j, y^j) + \sum_{i=1}^{m} \lambda_i^j \nabla_y g_i(x^j, y^j) \;=\; r^j, \qquad (6.1)$$

$$\lambda_i^j g_i(x^j, y^j) \;=\; s_i^j, \quad i = 1, \ldots, m. \qquad (6.2)$$

When $\mathrm{NLP}(x^k)$ (for $x^k \in X \cap \mathbb{Z}^{n_d}$) is infeasible, recall also that y^k represents an approximate optimal solution of $\mathrm{NLPF}_\infty(x^k)$ satisfying an inexact form of the corresponding KKT conditions. Here, we assume the existence of a vector of Lagrange multipliers $\mu^k \in \mathbb{R}^m$ (not necessarily non-negative) and vectors of residuals $z^k \in \mathbb{R}^m$, $w^k \in \mathbb{R}$, and $v^k \in \mathbb{R}^{n_c}$ such that

$$\sum_{i=1}^{m} \mu_i^k \nabla_y g_i(x^k, y^k) = v^k, \tag{6.3}$$

$$1 - \sum_{i=1}^{m} \mu_i^k = w^k, \tag{6.4}$$

$$\mu_i^k (g_i(x^k, y^k) - u^k) = z_i^k, \quad i = 1, \ldots, m. \tag{6.5}$$

The sets of indices T and S_∞ for collecting the approximate optimal solutions of these two types of NLP subproblems are defined as in Section 3.1, and we remember their definitions here:

$$T = \{j : x^j \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^j) \text{ is feasible and } y^j \text{ approximately solves } \mathrm{NLP}(x^j)\}$$

and

$$S_\infty = \{k : x^k \in X \cap \mathbb{Z}^{n_d}, \mathrm{NLP}(x^k) \text{ is infeasible and } y^k \text{ approximately solves } \mathrm{NLPF}_\infty(x^k)\}.$$

Now, for the feasible index $j \in T$, we begin by redefining a new vector of multipliers $\bar{\lambda}^j$, now non-negative, and corresponding new residuals $\bar{r}^j$ and $\bar{s}^j$. Our idea is to apply the analyses developed in Chapter 3 and 4 directly to this new setting, avoiding long rederivations. Given $j \in T$, we define the new inexact multipliers as

$$\bar{\lambda}_i^j = \begin{cases} \lambda_i^j, & \text{if } i \in P_{\lambda^j}, \\ -\lambda_i^j, & \text{if } i \in M_{\lambda^j}, \end{cases} \tag{6.6}$$

for $i = 1, \ldots, m$, and new residuals as

$$\bar{r}^j = r^j - 2 \sum_{i \in M_{\lambda^j}} \lambda_i^j \nabla_y g_i(x^j, y^j), \tag{6.7}$$

$$\bar{s}_i^j = \begin{cases} s_i^j, & \text{if } i \in P_{\lambda^j}, \\ -s_i^j, & \text{if } i \in M_{\lambda^j}, \end{cases}$$

for $i = 1, \ldots, m$. By using these new inexact multipliers and residuals, the inexact KKT conditions (6.1) and (6.2) can be rewritten equivalently, in the form

$$\begin{aligned} \nabla_y f(x^j, y^j) + \sum_{i=1}^{m} \bar{\lambda}_i^j \nabla_y g_i(x^j, y^j) &= \bar{r}^j, && (6.8)\\ \bar{\lambda}_i^j g_i(x^j, y^j) &= \bar{s}_i^j, \quad i = 1, \ldots, m. && (6.9) \end{aligned}$$

Note that $\bar{\lambda}^j$, from its definition in (6.6), is now a non-negative vector in $\mathbb{R}^m$. Thus, note also that (6.8) and (6.9) have exactly the same form and properties as the inexact KKT conditions (3.1) and (3.2) given in Section 3.1.

We can apply the same idea to the infeasible case. Consider any $k \in S_\infty$ and define the new inexact non-negative multipliers as

$$\bar{\mu}_i^j = \begin{cases} \mu_i^j, & \text{if } i \in P_{\mu^j}, \\ -\mu_i^j, & \text{if } i \in M_{\mu^j}, \end{cases} \tag{6.10}$$

for $i = 1, \ldots, m$. Consider also the new residuals $\bar{v}^k, \bar{w}^k$, and $\bar{z}^k$ defined as

$$\begin{aligned} \bar{v}^k &= v^k - 2 \sum_{i \in M_{\mu^k}} \mu_i^k \nabla_y g_i(x^k, y^k), \\ \bar{w}^k &= w^k + 2 \sum_{i \in M_{\mu^k}} \mu_i^k, \\ \bar{z}_i^k &= \begin{cases} z_i^k, & \text{if } i \in P_{\mu^k}, \\ -z_i^k, & \text{if } i \in M_{\mu^k}, \end{cases} \end{aligned}$$

for $i = 1, \ldots, m$. Then, equations (6.3)–(6.5) under these definitions, can also be rewritten, equivalently, in the form

$$\begin{aligned} \sum_{i=1}^{m} \bar{\mu}_i^k \nabla_y g_i(x^k, y^k) &= \bar{v}^k, && (6.11)\\ 1 - \sum_{i=1}^{m} \bar{\mu}_i^k &= \bar{w}^k, && (6.12)\\ \bar{\mu}_i^k (g_i(x^k, y^k) - u^k) &= \bar{z}_i^k, \quad i = 1, \ldots, m, && (6.13) \end{aligned}$$

where $\bar{\mu}^k$ is now a non-negative vector in $\mathbb{R}^m$, see (6.10). The inexact KKT conditions (6.11)–(6.13) have exactly the same form and properties as (3.3)–(3.5) in Section 3.1.

Thus, the whole approaches of Chapters 3 and 4 carry out to the setting of the current chapter, by simply replacing r^j, s^j, v^k, w^k, and z^k by $\bar{r}^j, \bar{s}^j, \bar{v}^k, \bar{w}^k$, and $\bar{z}^k$.

The question that is posed now is how to control the size of the new residuals, assuming that one can make the old ones as small as we want. Considering just the case of $j \in T$, and looking only at $\bar{r}^j$ (since $\|\bar{s}^j\| = \|s^j\|$), one can see from (6.7) that

$$\|\bar{r}^j\| \;\leq\; \|r^j\| + 2 \sum_{i \in M_{\lambda^j}} |\lambda_i^j| \|\nabla_y g_i(x^j, y^j)\|.$$

Thus, assuming that

$$\max_{i \in M_{\lambda^j}} \|\nabla_y g_i(x^j, y^j)\|$$

is a bounded quantity, one can make $\bar{r}^j$ as small as we want by reducing the size of the residual r^j and the size of the violation of non-negativity in the inexact multipliers,

$$\max_{i \in M_{\lambda^j}} |\lambda_i^j|,$$

possibly by resolving the NLP subproblem under tighter tolerances.

Chapter 7

Concluding remarks

In this thesis we have attempted to gain a better understanding of the effect of inexactness when solving NLP subproblems in two well-known decomposition techniques for Mixed Integer Nonlinear Programming (MINLP), the outer approximation (OA) and the generalized Benders decomposition (GBD).

As pointed out to us by I. E. Grossmann, solving the NLP subproblems inexactly in OA positions this approach somewhere in between exact OA and the extended cutting plane method (Section 2.4). It is part of our future work to better study how the inexact OA relates to the extended cutting plane method.

Regarding the conditions required on the residuals of the inexact KKT conditions, one can see from Assumptions 3.3.1 and 3.3.2 that the complete satisfaction of all those inequalities would ask for repeated NLP subproblem solution for all the possible discrete assignments. Such requirement would then undermine the practical purpose of saving computational effort aimed by the NLP subproblem inexactness. In our numerical tests we disregarded the conditions of Assumptions 3.3.1 and 3.3.2 and verified, after terminating each run of inexact OA or GBD, how many of them were violated among those identified during the course of the algorithm. The results indicated that convergence can be achieved without imposing Assumptions 3.3.1 and 3.3.2, and that the number of violated inequalities was relatively low (especially for OA). The results also seem to indicate that the cuts in OA and GBD must be changed accordingly when the corresponding NLP subproblems are solved inexactly. Testing these inexact approaches in a wider test set of larger problems and investigating how MINLP solvers could benefit from our approach is out of the scope of this thesis, although it seems a necessary step to further validate these indications.

Our study was performed under the assumption of convexity of the functions involved. Moreover, we also assumed that the approximate optimal solutions of the NLP subproblems were feasible in these subproblems. Relaxing this assumption introduces another layer of difficulty but certainly deserves attention in the future. In particular, our treatment of inexactness assumes a proper classification of feasibility. In practice, a misclassification of infeasibility can indeed cause numerical trouble. This is certainly another topic deserving a deeper study.

As we mentioned in the Introduction, the Feasibility Pump, denoted here as FP, is a method for the solution of MINLP problems of the type (1.1) which also alternates between solving NLP subproblems and MILP relaxed master problems. It is a non-rigorous method since at the end it only guarantees the computation of a feasible point, i.e., a point $(\tilde{x}, \tilde{y})$ satisfying $(\tilde{x}, \tilde{y}) \in (X \cap \mathbb{Z}^{n_d}) \times Y$ and $g(\tilde{x}, \tilde{y}) \leq 0$. FP requires the exact solution of the NLP subproblems. In the future, we plan to apply to FP similar ideas as in inexact OA and GBD to relax the exact solution of the NLP subproblems, by redefining the cuts in the master MILP relaxed problems with inexact residual information.

Bibliography

[1] *CMU/IBM MINLP Project.* http://egon.cheme.cmu.edu/ibm/page.htm.

[2] *IBM ILOG CPLEX.* http://www-01.ibm.com/software/integration/optimization/cplex-optimizer. Version 12.4, 2012.

[3] K. Abhishek, S. Leyffer, and J. T. Linderoth, *FilMINT: An outer approximation-based solver for convex mixed-integer nonlinear programs*, INFORMS J. Comput., 22 (2010), pp. 555–567.

[4] T. Achterberg and T. Berthold, *Improving the feasibility pump*, Discrete Optim., 4 (2007), pp. 77–86.

[5] A. Ahlatçioğlu and M. Guignard, *Convex hull relaxation (CHR) for convex and nonconvex MINLP problems with linear constraints.* http://www.optimization-online.org/DB_FILE/2011/01/2903.pdf. 2011.

[6] I. P. Androulakis, C. D. Maranas, and C. A. Floudas, *α-BB: A global optimization method for general constrained nonconvex problems*, J. Global Optim., 7 (1995), pp. 337–363.

[7] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, *Branching and bounds tightening techniques for non-convex MINLP*, Optim. Methods Softw., 24 (2009), pp. 597–634.

[8] J. F. Benders, *Partitioning procedures for solving mixed-variables programming problems*, Numer. Math., 4 (1962), pp. 238–252.

[9] L. Bertacco, M. Fischetti, and A. Lodi, *A feasibility pump heuristic for general mixed-integer problems*, Discrete Optim., 4 (2007), pp. 63–76.

[10] T. Berthold, *Primal Heuristics for Mixed Integer Programs*, master's thesis, Technical University of Berlin, 2006.

[11] T. Berthold and A. M. Gleixner, *Undercover – a primal heuristic for MINLP based on sub-MIPs generated by set covering*, in Proceedings of the European Workshop on Mixed Integer Nonlinear Programming, April 2010, pp. 103–112.

[12] 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*, Discrete Optim., 5 (2008), pp. 186–204.

[13] P. Bonami, G. Cornuéjols, A. Lodi, and F. Margot, *A feasibility pump for mixed integer nonlinear programs*, Math. Program., 119 (2009), pp. 331–352.

[14] P. Bonami and J. P. M. Gonçalves, *Heuristics for convex mixed integer nonlinear programs*, Comput. Optim. Appl., 51 (2012), pp. 729–747.

[15] P. Bonami, M. Kilinc, and J. Linderoth, *Algorithms and software for convex mixed integer nonlinear programs*, in Mixed Integer Nonlinear Programming, vol. 154 of IMA Vol. Math. Appl., Springer, New York, 2012, pp. 1–39.

[16] P. Bonami, J. Lee, S. Leyffer, and A. Wächter, *More branch-and-bound experiments in convex nonlinear integer programming*. Preprint ANL/MCS-P1949-0911, Argonne National Laboratory, Mathematics and Computer Science Division, 2011.

[17] S. Burer and A. N. Letchford, *Non-convex mixed-integer nonlinear programming: A survey*, Surveys in Operations Research and Management Science, 17 (2012), pp. 97–106.

[18] M. R. Bussieck and A. Pruessner, *Mixed-integer nonlinear programming*, SIAG/OPT Views-and-News, 14 (2003), pp. 19–22.

[19] I. Castillo, J. Westerlund, S. Emet, and T. Westerlund, *Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization methods*, Comput. Chem. Eng., 30 (2005), pp. 54–69.

[20] R. J. Dakin, *A tree-search algorithm for mixed integer programming problems*, The Computer Journal, 8 (1965), pp. 250–255.

[21] C. D'Ambrosio and A. Lodi, *Mixed integer nonlinear programming tools: A practical overview*, 4OR, 9 (2011), pp. 329–349.

[22] E. Danna, E. Rothberg, and C. L. Pape, *Exploring relaxation induced neighborhoods to improve MIP solutions*, Math. Program., 102 (2005), pp. 71–90.

[23] M. A. Duran and I. E. Grossmann, *A mixed-integer nonlinear programming algorithm for process systems synthesis*, American Institute of Chemical Engineers Journal, 32 (1986), pp. 592–606.

[24] ——, *An out-approximation algorithm for a class of mixed-integer nonlinear programs*, Math. Program., 36 (1986), pp. 307–339.

[25] M. Fischetti, F. Glover, and A. Lodi, *The feasibility pump*, Math. Program., 104 (2005), pp. 91–104.

[26] M. Fischetti and D. Salvagnin, *Feasibility pump 2.0*, Math. Program. Comput., 1 (2009), pp. 201–222.

[27] R. Fletcher and S. Leyffer, *Solving mixed integer nonlinear programs by outer approximation*, Math. Program., 66 (1994), pp. 327–349.

[28] O. E. Flippo and A. H. G. R. Kan, *Decomposition in general mathematical programming*, Math. Program., 60 (1993), pp. 361–382.

[29] O. E. Flippo, A. H. G. R. Kan, and G. van der Hoek, *Duality and Decomposition in General Mathematical Programming*, Econometric Institute, University of Rotterdam, 1987. Report 8747/B.

[30] A. M. Geoffrion, *Duality in nonlinear programming: A simplified application-oriented development*, SIAM Rev., 13 (1971), pp. 1–37.

[31] ——, *Generalized Benders decomposition*, J. Optim. Theory Appl., 10 (1972), pp. 237–260.

[32] I. E. Grossmann, *Review of nonlinear mixed-integer and disjunctive programming techniques*, Optim. Eng., 3 (2002), pp. 227–252.

[33] I. E. Grossmann and N. V. Sahinidis (eds.), *Special issue on mixed-integer programming and its applications to engineering, Part I*, Optim. Eng., 3 (2002).

[34] ——, *Special issue on mixed-integer programming and its applications to engineering, Part II*, Optim. Eng., 4 (2003).

[35] O. K. Gupta and V. Ravindran, *Branch and bound experiments in convex nonlinear integer programming*, Management Science, 31 (1985), pp. 1533–1546.

[36] J. E. Kelley, *The cutting-plane method for solving convex programs*, J. Soc. Indust. Appl. Math., 8 (1960), pp. 703–712.

[37] 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*, Math. Program. Comput., 3 (2011), pp. 103–163.

[38] G. R. Kocis and I. E. Grossmann, *Relaxation strategy for the structural optimization of process flow sheets*, Industrial and Engineering Chemical Research, 26 (1987), pp. 1869–1880.

[39] ——, *Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis*, Industrial and Engineering Chemical Research, 27 (1988), pp. 1407–1421.

[40] C. D. Laird, L. T. Biegler, and B. V. B. Waanders, *A mixed integer approach for obtaining unique solutions in source inversion of drinking water networks*, Journal of Water Resource Management and Planning, 132 (2006), pp. 242–251.

[41] A. H. Land and A. G. Doig, *An automatic method of solving discrete programming problems*, Econometrica, 28 (1960), pp. 497–520.

[42] A. N. Letchford, *Mixed-integer non-linear programming: A survey*, 1st LANCS Workshop on Discrete and Non-Linear Optimisation, Southampton, 2009.

[43] S. Leyffer, *MacMINLP. http://wiki.mcs.anl.gov/leyffer/index.php/MacMINLP*.

[44] ——, *Deterministic Methods for Mixed Integer Nonlinear Programming*, PhD thesis, University of Dundee, Dundee, Scotland, UK, 1993.

[45] ——, *Integrating SQP and branch-and-bound for mixed integer nonlinear programming*, Comput. Optim. Appl., 18 (2001), pp. 295–309.

[46] S. Leyffer and J. Linderoth, *A practical guide to mixed integer nonlinear programming (MINLP). http://coral.ie.lehigh.edu/wp-content/uploads/presentations/siopt-05-minlp-presentation.pdf*. 2005.

[47] M. Li and L. N. Vicente, *Inexact solution of NLP subproblems in MINLP*, J. Global Optim., (to appear).

[48] H. M. Markowitz, *Portfolio selection*, The Journal of Finance, 7 (1952), pp. 77–91.

[49] ——, *Portfolio Selection: Efficient Diversification of Investments*, Wiley, New York, 1959.

[50] G. Nannicini, P. Belotti, J. Lee, J. Linderoth, F. Margot, and A. Wächter, *A probing algorithm for MINLPs with early detection of failures by SVM*, in CPAIOR 2011: The 8th International Conference on Integration of Artificial Intelligence and Operations Research, vol. 6697 of Lecture Notes in Comput. Sci., Springer, New York, 2011, pp. 154–169.

[51] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York, 1988.

[52] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, Berlin, second ed., 2006.

[53] I. Quesada and I. E. Grossmann, *An LP/NLP based branch and bound algorithm for convex MINLP optimization problems*, Comput. Chem. Eng., 16 (1992), pp. 937–947.

[54] D. E. Ravemark, *Optimization Models for Design and Operation of Chemical Batch Processes*, PhD thesis, Swiss Federal Institute of Technology, 1995.

[55] H. S. Ryoo and N. V. Sahinidis, *A branch-and-reduce approach to global optimization*, J. Global Optim., 8 (1996), pp. 107–138.

[56] S. Sager, *Numerical Methods for Mixed-Integer Optimal Control Problems*, PhD thesis, Interdisciplinary Center for Scientific Computing, Universität Heidelberg, 2005.

[57] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, Chichester, 1986.

[58] M. Tawarmalani and N. V. Sahinidis, *Global optimization of mixed-integer nonlinear programs: A theoretical and computational study*, Math. Program., 99 (2004), pp. 563–591.

[59] A. Vechietti and I. E. Grossmann, *LOGMIP: A disjunctive 0-1 non-linear optimizer for process system models*, Comput. Chem. Eng., 23 (1999), pp. 555–565.

[60] J. Viswanathan and I. E. Grossmann, *A combined penalty function and outer-approximation method for MINLP optimization*, Comput. Chem. Eng., 14 (1990), pp. 769–782.

[61] T. Westerlund and F. Pettersson, *An extended cutting plane method for solving convex MINLP problems*, Comput. Chem. Eng., 19 (1995), pp. 131–136.

[62] T. Westerlund and R. Pörn, *Solving pseudo-convex mixed integer optimization problems by cutting plane techniques*, Optim. Eng., 3 (2002), pp. 253–280.

[63] T. Westerlund, H. Skrifvars, I. Harjunkoski, and R. Pörn, *An extended cutting plane method for a class of non-convex MINLP problems*, Comput. Chem. Eng., 22 (1998), pp. 357–365.