Inexact Solution of NLP Subproblems in MINLP

M. Li [*]   L. N. Vicente[†]

October 25, 2012

Abstract

In the context of convex mixed integer nonlinear programming (MINLP), we investigate how the outer approximation method and the generalized Benders decomposition method are affected when the respective nonlinear programming (NLP) subproblems are solved inexactly. We show that the cuts in the corresponding master problems can be changed to incorporate the inexact residuals, still rendering equivalence and finiteness in the limit case. Some numerical results will be presented to illustrate the behavior of the methods under NLP subproblem inexactness.

Keywords: Mixed integer nonlinear programming, outer approximation, generalized Benders decomposition, inexactness, convexity.

1 Introduction

Recently, mixed integer nonlinear programming (MINLP) has become again a very active research area [3, 4, 6, 7, 8, 9, 17, 19]. Benders [5] developed in the 60's a technique for solving mixed integer linear programming (MILP) problems, later called Benders decomposition. Geoffrion [13] extended it to MINLP in 1972, in what has become known as generalized Benders decomposition (GBD). Much later, in 1986, Duran and Grossmann [10] derived a new outer approximation (OA) 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 and one MILP master problem.

For these particular MINLP problems, Quesada and Grossmann [18] 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 [14, 18] associate this fact with the observation that GBD requires a large number of major iterations to converge). Fletcher and Leyffer [11] generalized the OA method of Duran and Grossmann [10] 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 [14] in this wider context of MINLP problems, showing once more that the lower bound predicted by the relaxed master problem of OA is greater than or equal to the one predicted by the relaxed master problem of GBD (see also Flippo and Rinnooy Kan [12] for the relationship between the two techniques). Recently, Bonami et al. [6] 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 [11]), where the cuts in the master MILP problems do not involve linearizations of the objective function in the infeasible case.

Westerlund and Pettersson [20] generalized the cutting plane method [15] from convex NLP to convex MINLP, in what is known as the extended cutting plane (ECP) method (see also [21, 22]). While OA and GBD alternate between the solution of MILP and NLP subproblems, the ECP relies only on the solution of MILP problems.

In the above mentioned OA and GBD approaches, the NLP subproblems are solved exactly, at least for the derivation of the theoretical properties, such as equivalence between original and master problem and finite termination of the corresponding algorithms. In this paper we investigate the effect of NLP subproblem inexactness in these two techniques. 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 convexity properties. Our treatment of inexactness does not, however, address inexactness in the classification of NLP subproblems feasibility.

In this paper, we will adopt the MINLP formulation

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

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 will also assume that P is convex, i.e., that f and g are convex functions.

Let x^j be any element of $X \cap \mathbb{Z}^{n_d}$. Consider, then, the (convex) subproblem

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

and suppose it is feasible. In this case, y^j will represent an approximate optimal solution of NLP(x^j). For an x^k in $X \cap \mathbb{Z}^{n_d}$ for which NLP(x^k) is infeasible, y^k is instead defined as an approximate optimal solution of the following feasibility (convex) subproblem

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

where one minimizes the ℓ_∞-norm of the measure of infeasibility of subproblem NLP(x^k).

For a matter of simplificity, and without loss of generality, we suppose that the constraints $y \in Y$ are part of the constraints $g(x^j, y) \le 0$ and $g_i(x^k, y) \le u$, $i = 1, \ldots, m$, in the subproblems NLP(x^j) and NLPF(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 Karush-Kuhn-Tucker (KKT) conditions. More particularly, in the case of 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

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

$$\lambda_i^j g_i(x^j, y^j) = s_i^j, \quad i = 1, \ldots, m. \tag{2}$$

When NLP(x^k) is infeasible, we assume, for NLPF(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

$$\sum_{i=1}^m \mu_i^k \nabla_y g_i(x^k, y^k) = v^k, \tag{3}$$

$$1 - \sum_{i=1}^m \mu_i^k = w^k, \tag{4}$$

$$\mu_i^k(g_i(x^k, y^k) - u^k) = z_i^k, \quad i = 1, \ldots, m. \tag{5}$$

Points satisfying the inexact KKT conditions can be seen as solutions of appropriate perturbed subproblems (see the Appendix). The size of the residuals r^j, s^j, v^k, w^k, and z^k will be required to satisfy appropriate assumptions. The following two sets will then be used to index these two sets of approximate optimal solutions:

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

and

$$S = \{k : x^k \in X \cap \mathbb{Z}^{n_d}, \text{NLP}(x^k) \text{ is infeasible}\}.$$

Let y^j and y^k denote approximate solutions of NLP(x^j) and NLPF(x^k), respectively, in the sense of (1)–(2) and (3)–(5). The inexact versions of OA and GBD studied in this paper 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 $\mathcal{P}$:

$$\mathcal{P} \quad \min_{j \in T} f(x^j, y^j). \tag{6}$$

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

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

3

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.

We organize the paper in the following way. In Section 2, we extend OA for the inexact solution of the NLP subproblems, rederiving the corresponding background theory and main algorithm. In Section 3 we proceed similarly for GBD, also discussing the relationship between the inexact forms of OA and GBD. Section 4 describes a set of preliminary numerical experiments, reported to better understand some of the theoretical features encountered in our study of inexactness in MINLP.

2 Inexact outer approximation

2.1 Equivalence between perturbed and master problems for OA

OA relies on the fact that the original problem P is equivalent to a MILP (master problem) formed by minimizing the least of the linearized forms of f for indices in T subject to the linearized forms of g for indices in S and T. When the NLP subproblems are solved inexactly, 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

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

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

$$
t_i^j \ = \ \begin{cases} \frac{s_i^j}{\lambda_i^j}, & \text{if } \lambda_i^j \ > \ 0, \\ 0, & \text{if } \lambda_i^j \ = \ 0, \end{cases} \tag{7}
$$

$$
a_i^k \ = \ \begin{cases} \frac{m_0 z_i^k - w^k u^k}{m_0 \mu_i^k}, & \text{if } \mu_i^k \ > \ 0, \\ 0, & \text{if } \mu_i^k \ = \ 0 \end{cases} \tag{8}
$$

(m_0 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, for instance, in [6].

From the convexity and continuous differentiability of f and g, we know that, for any

4

$(x^l, y^l) \in \mathbb{R}^{n_d} \times \mathbb{R}^{n_c}$,

$$f(x,y) \geq f(x^l, y^l) + \nabla f(x^l, y^l)^\top \begin{pmatrix} x - x^l \\ y - y^l \end{pmatrix}, \tag{9}$$

$$g(x,y) \geq g(x^l, y^l) + \nabla g(x^l, y^l)^\top \begin{pmatrix} x - x^l \\ y - y^l \end{pmatrix}. \tag{10}$$

In addition, when y^j is a feasible point of NLP(x^j), we obtain from (10) and $g(x^j, y^j) \leq 0$ that

$$0 \geq g(x^l, y^l) + \nabla g(x^l, y^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix}. \tag{11}$$

The inexact OA method reported in this section, as well as the GBD method of the next section, 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 2.1 *Given any $l, j \in T$, with $l \neq j$, assume that*

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

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

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

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 2.1 (see Figure 1). In the case of exact cuts, convexity trivially enforces the validness of the cuts (see also Figure 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 2.2 *Given any $j \in T$ and any $k \in S$, and for all $i \in \{1, \ldots, m\}$, if $\mu_i^k \neq 0$, assume that*

$$\frac{1}{1 - w^k} \|v^k\| \|y^j - y^k\| + \frac{1}{\mu_i^k} |z_i^k| + \frac{u^k}{m_0 \mu_i^k} |w^k| \leq -\beta_i [(\nabla g_i^k)^\top \begin{pmatrix} x^j - x^k \\ y^j - y^k \end{pmatrix} + g_i^k],$$

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

$$\frac{1}{1 - w^k} \|v^k\| \|y^j - y^k\| \leq -\eta_i [(\nabla g_i^k)^\top \begin{pmatrix} x^j - x^k \\ y^j - y^k \end{pmatrix} + g_i^k],$$
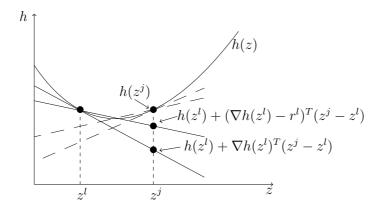
for some $\eta_i \in [0,1]$.

Figure 1: Exact and inexact cuts under convexity (for a generic function h).

The motivation and restrictiveness of Assumption 2.2 are the same as in Assumption 2.1.

It is important to note that the complete satisfaction of the inequalities stated in Assumptions 2.1 and 2.2 requires the inexact solution for all possible assignments of the discrete variables. As one can see from the proof of Theorem 2.1 below, the non satisfaction of one of the inequalities in these assumptions may have as potential effect the deterioration of the upper bound on the optimal value of the original MINLP and consequently failure to determine an optimal solution.

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 2.1 *Let* P *be a convex* MINLP *problem and* $\mathcal{P}$ *be its perturbed problem as defined in the Introduction (where the classification of the* NLP *subproblems feasibility is detected 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 2.1 and 2.2. Then* $\mathcal{P}^{\mathrm{OA}}$ *and* $\mathcal{P}$ *have the same optimal value.*

Proof. The proof follows closely the lines of the proof of [6, Theorem 1]. Since problem P has a finite optimal value it follows 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, that the sets T and S are well defined, and that the set T is nonempty. Now, given any $x^l \in X \cap \mathbb{Z}^{n_d}$ with $l \in T \cup S$, 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 $k \in S$.

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

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

$$\left(\nabla_y g_i(x^k, y^k) - \frac{1}{1 - w^k} v^k\right)^\top (y - y^k) + g_i(x^k, y^k) \;\leq\; a_i^k, \tag{12}$$

for $i = 1, \ldots, m$. Multiplying the inequalities in (12) by the nonnegative multipliers $\mu_1^k, \ldots, \mu_m^k$,

and summing them up, one obtains

$$(\sum_{i=1}^{m} \mu_i^k \nabla_y g_i(x^k, y^k) - v^k)^\top (y - y^k) \leq \sum_{i=1}^{m} (z_i^k - \mu_i^k g_i(x^k, y^k)) - w^k u^k. \tag{13}$$

By using (3), one can see that the left hand side of the inequality in (13) is equal to 0. On the other hand, by using equation (5), the right hand side of the inequality in (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 (4). Since $\text{NLP}(x^k)$ is infeasible, $-u^k$ must be strictly negative. We have thus proved that the inequality (13) has no solution y.

This derivation implies that the minimum value of $\mathcal{P}^{\text{OA}}$ should be found as the minimum value of $\mathcal{P}^{\text{OA}}_{x^j}$ 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 $\mathcal{P}^{\text{OA}}_{x^j}$ coincides with the approximate optimal value of $\text{NLP}(x^j)$.

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

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

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

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

$$\begin{pmatrix} \nabla_x f(x^l, y^l) \\ \nabla_y f(x^l, y^l) - r^l \end{pmatrix}^\top \begin{pmatrix} x^j - x^l \\ y - y^l \end{pmatrix} + f(x^l, y^l) \leq \alpha, \tag{14}$$

$$\nabla g(x^l, y^l)^\top \begin{pmatrix} x^j - x^l \\ y - y^l \end{pmatrix} + g(x^l, y^l) \leq t^l, \tag{15}$$

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

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

as well as all the constraints indexed by $k \in S$

$$\nabla f(x^k, y^k)^\top \begin{pmatrix} x^j - x^k \\ y - y^k \end{pmatrix} + f(x^k, y^k) \leq \alpha, \tag{16}$$

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

$$\begin{pmatrix} \nabla_x g_i(x^k, y^k) \\ \nabla_y g_i(x^k, y^k) - \frac{1}{1-w^k} v^k \end{pmatrix}^\top \begin{pmatrix} x^j - x^k \\ y - y^k \end{pmatrix} + g_i(x^k, y^k) \leq a_i^k, \tag{17}$$

where a_i^k is given as in (8).

First take any $l \in T$ and assume that y^l is an approximate optimal solution of $\text{NLP}(x^l)$ with corresponding inexact nonnegative Lagrange multipliers λ^l. If $l = j$, it is easy to verify that $(y^j, f(x^j, y^j))$ satisfies (14) and (15). Assume then that $l \neq j$. From Assumption 2.1, we know that, for some $\tau \in [0, 1]$,

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

Thus,

$$[(\nabla f^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + f^l - f^j] - (r^l)^\top (y^j - y^l)$$

$$\leq (1-\tau)[(\nabla f^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + f^l - f^j] \leq 0,$$

where the last inequality comes from $1 - \tau \geq 0$ and (9) with $(x,y) = (x^j, y^j)$. We then see that (14) is satisfied with $\alpha = f(x^j, y^j)$ and $y = y^j$.

Now, from Assumption 2.1, one has for some $\sigma_i \in [0,1], i = 1, \ldots, m$,

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

$$- \sigma_i \lambda_i^l[(\nabla g_i^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + g_i^l]$$

$$\leq (1 - \sigma_i)\lambda_i^l[(\nabla g_i^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + g_i^l]$$

$$\leq 0,$$

where the last inequality is justified by (11) and $\sigma_i \in [0,1]$. Thus,

$$\lambda_i^l[(\nabla g_i^l)^\top \begin{pmatrix} x^j - x^l \\ y^j - y^l \end{pmatrix} + g_i^l] \leq s_i^l, \; i = 1, \ldots, m. \tag{18}$$

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 (15) with $y = y^j$. If $\lambda_i^l \neq 0$, then (18) can be written as:

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

which also shows that the constraints (15) hold with $y = y^j$.

Finally, we take any $k \in S$ and assume that y^k is an approximate optimal solution of NLPF(x^k) with corresponding inexact Lagrange multipliers μ^k. It results trivially from the assumption on convexity that the constraints (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 2.2, we have for some $\beta_i \in [0,1]$, that

$$-\frac{1}{1-w^k}(v^k)^\top(y^j - y^k) - \frac{1}{\mu_i^k}z_i^k + \frac{u^k}{m_0\mu_i^k}w^k \leq -\beta_i[(\nabla g_i^k)^\top \begin{pmatrix} x^j - x^k \\ y^j - y^k \end{pmatrix} + g_i^k],$$

i.e.,

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

by the definition of a_i^k. Thus, the constraints (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 2.2, we have that, for some $\eta_i \in [0,1]$,

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

$$\leq 0.$$

This also shows that the constraints (17) hold with $y = y^j$.

We can therefore say that $(y^j, f(x^j, y^j))$ is a feasible point of $\mathcal{P}_{x^j}^{OA}$, 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 $\mathcal{P}_{x^j}^{OA}$ for $j \in T$.

Recall that y^j is an approximate optimal solution of $\text{NLP}(x^j)$ satisfying the inexact KKT conditions (1) and (2). By construction, any solution of $\mathcal{P}_{x^j}^{OA}$ has to satisfy the inexact outer-approximation constraints:

$$(\nabla_y f(x^j, y^j) - r^j)^\top (y - y^j) + f(x^j, y^j) \leq \alpha, \tag{19}$$

$$\nabla_y g(x^j, y^j)^\top (y - y^j) + g(x^j, y^j) \leq t^j. \tag{20}$$

Multiplying the inequalities (20) by the nonnegative multipliers $\lambda_1^j, \ldots, \lambda_m^j$ and summing them together with (19), one obtains

$$(\nabla_y f(x^j, y^j) - r^j)^\top (y - y^j) + f(x^j, y^j) + \sum_{i=1}^m \lambda_i^j (\nabla_y g_i(x^j, y^j)^\top (y - y^j) + g_i(x^j, y^j) - s_i^j)$$
$$\leq \alpha. \tag{21}$$

The left hand side of the inequality (21) can be rewritten as:

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

By using (1) and (2), this quantity is equal to $f(x^j, y^j)$, and it follows that inequality (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 $\mathcal{P}_{x^j}^{OA}$ and perturbed $\text{NLP}(x^j)$ have the same optimal value. In other words, the MILP problem $\mathcal{P}^{OA}$ has the same optimal value as the perturbed problem $\mathcal{P}$ given by (6). ∎

Since in the exact case, all the KKT residuals are zero, it results from Theorem 2.1 what is well known for OA:

Corollary 2.1 *Let* P *be a convex* MINLP *problem. Assume that* P *is feasible with a finite optimal value and the residuals of the* KKT *conditions of the* NLP *subproblems are zero. Then* $\mathcal{P}^{OA}$ *and* P *have the same optimal value.*

In the paper [6], the feasibility NLP subproblem is stated as

$$\text{P}_{x^k}^F \begin{cases} \min & \sum_{i=1}^m u_i \\ \text{s.t.} & g(x^k, y) \leq u, \\ & u \geq 0, \\ & y \in Y, u \in \mathbb{R}^m. \end{cases}$$

Note that one could easily rederive a result similar to [6, Theorem 1] replacing their $\text{P}_{x^k}^F$ by our $\text{NLPF}(x^k)$. In fact, the argument needed here is essentially Part I of the proof of Theorem 2.1 with v, w, and z set to zero.

2.2 Inexact-OA algorithm

One knows that the outer approximation algorithm terminates finitely in the convex case and when the optimal solutions of the NLP subproblems satisfy the first order KKT conditions (see [11]). 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, at each step of the inexact OA algorithm, one tries to solve a subproblem $\text{NLP}(x^p)$, where x^p is chosen as a new discrete assignment. Two results can then occur: either $\text{NLP}(x^p)$ is feasible and an approximate optimal solution y^p can be given, or this subproblem is found infeasible and another NLP subproblem, $\text{NLPF}(x^p)$, is solved, yielding an approximate optimal solution y^p. In the algorithm, the sets T and S defined in the Introduction will be replaced by:

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

and

$$S^p = \{k : k \le p,\ x^k \in X \cap \mathbb{Z}^{n_d}, \text{NLP}(x^k) \text{ is infeasible}\}.$$

As before, y^j and y^k denote the approximate solutions of $\text{NLP}(x^j)$ and $\text{NLPF}(x^k)$, 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–iteration, one needs to add the constraint

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

where

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

Then we define the following inexact relaxed MILP master problem

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

where t^j and a_i^k were defined in (7) and (8), respectively. The presentation of the inexact OA algorithm (given next) and the proof of its finiteness in Theorem 2.2 follows the lines in [11].

Algorithm 2.1 (Inexact Outer Approximation)

Initialization

   Let x^0 be given. Set $p = 0, T^{-1} = \emptyset, S^{-1} = \emptyset$, and $\text{UBD} = +\infty$.

REPEAT

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

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

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

4. Solve the relaxed master problem $(\mathcal{P}^{\text{OA}})^p$, 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 $\text{UBD} = +\infty$, 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 2.2 *Let* P *be a convex* MINLP *problem and* $\mathcal{P}$ *be its perturbed problem as defined in the Introduction (where the classification of the* NLP *subproblems feasibility is detected 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 2.1 and 2.2. Then Algorithm 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.*

   Proof. Since the set X is bounded by assumption, finite termination of Algorithm 2.1 will be established by proving that no discrete assignment is generated twice by the algorithm.
   Let $q \leq p$. If $q \in S^p$, it has been shown in Part I of the proof of Theorem 2.1 that the corresponding constraint in $\mathcal{P}^{\text{OA}}_{x^p}$, derived from the feasibility problem $\text{NLPF}(x^q)$, 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 $\text{NLP}(x^q)$ 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

$$\nabla_y f^q + \sum_{i=1}^m \lambda_i^q \nabla_y g_i(x^q, y^q) \;\; = \;\; r^q, \tag{22}$$

$$\lambda_i^q g_i(x^q, y^q) \;\; = \;\; s_i^q, \quad i = 1, \dots, m. \tag{23}$$

11

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

$$\alpha^p \; < \; \text{UBD}^p \; \leq \; f^q, \tag{24}$$

$$\left(\begin{array}{c} \nabla_x f^q \\ \nabla_y f^q - r^q \end{array} \right)^\top \left(\begin{array}{c} 0 \\ y - y^q \end{array} \right) + f^q \; \leq \; \alpha^p, \tag{25}$$

$$(\nabla g^q)^\top \left(\begin{array}{c} 0 \\ y - y^q \end{array} \right) + g^q \; \leq \; t^q, \tag{26}$$

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

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

Multiplying the rows in (26) by the Lagrange multipliers $\lambda_i^q \geq 0$, $i = 1, \ldots, m$, and adding (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:

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

Using (22) and (23), this is equal to f^q and therefore we obtain the inequality

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

which contradicts (24).

The rest of the proof is exactly as in [11, Theorem 2] but we repeat here for completeness and possible changes in notation. Finally, we will show that Algorithm 2.1 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 2.1, (x^*, y^*, f^*) is also an optimal solution of $\mathcal{P}^{OA}$. Now assume that the algorithm terminates indicating a nonoptimal point (x', y') with $f' > f^*$. In such a situation, the previous relaxation of the master problem $\mathcal{P}^{OA}$ after adding the constraints at the point (x', y', f'), called $(\mathcal{P}^{OA})^p$, is infeasible, causing the above mentioned termination. We will get a contradiction by showing that (x^*, y^*, f^*) is feasible for $(\mathcal{P}^{OA})^p$. First, by the assumption that UBD $= f' > f^*$, the first constraint $\alpha = f^* <$ UBD of $(\mathcal{P}^{OA})^p$ holds. Secondly, since (x^*, y^*, f^*) is an optimal solution to $\mathcal{P}^{OA}$, it must be feasible for all other constraints of $(\mathcal{P}^{OA})^p$. Therefore, the algorithm could not terminate at (x', y') with UBD $= f'$. ∎

3 Inexact generalized Benders decomposition

3.1 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:

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

One can easily recognize the classical form of (exact) GBD master problem when $w^k = 0$ and $z^k = 0$. Moreover, as we show in the Appendix, this MILP can also be derived in the inexact case from a perturbed duality representation of the original, perturbed problem.

A proof similar to the one of exact GBD and exact and inexact OA (Theorem 2.1) allows us to establish the desired equivalence between the original, perturbed MINLP problem and the MILP master problem $\mathcal{P}^{\mathrm{GBD}}$.

Theorem 3.1 *Let* P *be a convex* MINLP *problem and* $\mathcal{P}$ *be its perturbed problem as defined in the Introduction (where the classification of the* NLP *subproblems feasibility is detected 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 2.1 and 2.2. 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 $l \in T \cup S$, 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 $k \in S$. When we set $x = x^k$, in the corresponding constraint of $\mathcal{P}^{\mathrm{GBD}}$, one obtains

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

From (4) and (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 $\mathcal{P}^{\mathrm{GBD}}_{x^j}$

$$f(x^l, y^l) + \nabla_x f(x^l, y^l)^\top (x^j - x^l) + \sum_{i=1}^m \lambda_i^l \nabla_x g_i(x^l, y^l)^\top (x^j - x^l) \;\leq\; \alpha, \; \forall l \in T, \qquad (27)$$

$$\sum_{i=1}^m \mu_i^k [g_i(x^k, y^k) + \nabla_x g_i(x^k, y^k)^\top (x^j - x^k)] + w^k u^k - \sum_{i=1}^m z_i^k \;\leq\; 0, \; \forall k \in S \qquad (28)$$

are satisfied with $\alpha = f(x^j, y^j)$. Under Assumptions 2.1 and Assumption 2.2, we know from the proof of Theorem 2.1 (Part II–A) that the following hold: (14) with $y = y^j$ and $\alpha = f(x^j, y^j)$, (15) with $y = y^j$, and (17) with $y = y^j$.

When $l \in T$, multiplying the inequalities (15) with $y = y^j$ by the nonnegative multipliers $\lambda_1^l, \ldots, \lambda_m^l$ and summing them together with (14) with $y = y^j$ and $\alpha = f(x^j, y^j)$, one obtains

$$f(x^l, y^l) + \nabla_x f(x^l, y^l)^\top (x^j - x^l) + \sum_{i=1}^m \lambda_i^l \nabla_x g_i(x^l, y^l)^\top (x^j - x^l)$$

$$\leq \; f(x^j, y^j) - [\nabla_y f(x^l, y^l) + \sum_{i=1}^m \lambda_i^l \nabla_y g_i(x^l, y^l) - r^l]^\top (y^j - y^l) - \sum_{i=1}^m \lambda_i^l g(x^l, y^l) + \sum_{i=1}^m \lambda_i^l t_i^l.$$

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 (27) holds with $\alpha = f(x^j, y^j)$.

When $k \in S$, multiplying the inequalities in (17) with $y = y^j$ by the nonnegative multipliers $\mu_1^k, \ldots, \mu_m^k$, and summing them up, one obtains using (3) and (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 (28).

Thus, $f(x^j, y^j)$ is a feasible point of $\mathcal{P}_{x^j}^{\text{GBD}}$, and therefore $f(x^j, y^j)$ is an upper bound on the optimal value $\bar{\alpha}^j$ of $\mathcal{P}_{x^j}^{\text{OA}}$. To show that is also a lower bound, i.e., that $\bar{\alpha}^j \geq f(x^j, y^j)$, note that from (27), when $l = j$, $\mathcal{P}_{x^j}^{\text{GBD}}$ 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 $\mathcal{P}_{x^j}^{\text{GBD}}$ and perturbed NLP(x^j) have the same optimal value, which concludes the proof. ■

When all the KKT residuals are zero we obtain as a corollary the known equivalence result in GBD:

Corollary 3.1 *Let* P *be a convex* MINLP *problem. Assume that* P *is feasible with a finite optimal value and the residuals of the* KKT *conditions of the* NLP *subproblems are zero. Then* $\mathcal{P}^{\text{GBD}}$ *and* P *have the same optimal value.*

Remark 3.1 *It is well known that the constraints of the* GBD *master problem can be derived from the corresponding ones of the* OA *master problem, in the convex, exact case (see [18]). The same happens naturally in the inexact case. In fact, from the proof of Theorem 3.1 above, we can see that the constraints in* $\mathcal{P}_{x^j}^{\text{OA}}$, *for* $j \in T$, *imply the corresponding ones in* $\mathcal{P}_{x^j}^{\text{GBD}}$. *Moreover, one can easily see that any of the constraints in* $\mathcal{P}^{\text{OA}}$ *imply the corresponding ones in* $\mathcal{P}^{\text{GBD}}$.

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).*

3.2 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^p are the same as those in Section 2.2. The relaxed MILP to be solved at each iteration is thus given by

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

The inexact GBD algorithm is given next (and follows the presentation in [11] for OA).

Algorithm 3.1 (Inexact GBD Approximation)

Initialization
 Let x^0 be given. Set $p = 0, T^{-1} = \emptyset, S^{-1} = \emptyset$, and $\text{UBD} = +\infty$.

REPEAT

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

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

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

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

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

Similarly as in Theorem 2.2 for OA, one can establish that the above inexact GBD algorithm terminates in a finite number of steps.

Theorem 3.2 *Let* P *be a convex* MINLP *problem and* $\mathcal{P}$ *be its perturbed problem as defined in the Introduction (where the classification of the* NLP *subproblems feasibility is detected 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 2.1 and 2.2. Then Algorithm 3.1 terminates in a finite number of steps at an optimal solution of* $\mathcal{P}$ *or with an indication that* $\mathcal{P}$ *is infeasible.*

4   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 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 [10].

The implementation and testing of Algorithms 2.1 and 3.1 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 master problems. The constraint $\alpha < \text{UBD}^p$ was implemented as $\alpha \leq \text{UBD}^p - 10^{-5}$. (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:

$$\min\{10^{-2}, \max\{10^{-6}, 10^{-\theta} \times |\text{UBD}^{p-1} - \text{UBD}^{p-2}|\}\}, \quad p \geq 2, \tag{29}$$

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`.

In the tables of results we report the number N of iterations taken by Algorithms 2.1 and 3.1. We also report, in the rows corresponding to the variant `i.cuts`, the number C of constraint inequalities of Assumptions 2.1 and 2.2 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 2.1 and 2.2 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 master program being infeasible, or the number of iterations exceeding 50 (MAX in the tables), or the solution of the MILP 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.

4.1   Results for inexact OA method

Tables 2–3 summarize the application of inexact OA (Algorithm 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`) to our test set. For problems synthes2 and synthes3 in Table 2 and

16

Table 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 [16] 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

Table 2: Application of inexact OA to problems from the MacMINLP collection [16]. The table reports the number N of iterations taken as well as the number C of inequalities found to violate Assumptions 2.1 and 2.2.

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

problems CLay0203H, CLay0303H, CLay0304H, and FLay04M in Table 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 most of the cases while neglecting the imposition of the inequalities of Assumptions 2.1 and 2.2.

4.2 Results for inexact GBD method

Tables 4–5 summarize the application of inexact GBD (Algorithm 3.1) (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 solution, 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 3.1, might be expected since inexact GBD yields weaker lower bounds and has been observed to take more major iterations to converge than OA (see [14, 18]). The number of inequalities of Assumptions 2.1 and 2.2 violated in inexact GBD is also higher than the one in inexact OA.

Table 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 2.1 and 2.2.

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 4: Application of inexact GBD to problems from the MacMINLP collection [16]. The table reports the number N of iterations taken as well as the number C of inequalities found to violate Assumptions 2.1 and 2.2.

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

5 Conclusions and final remarks

In this paper 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 [20].

Regarding the conditions required on the residuals of the inexact KKT conditions, one can see from Assumptions 2.1 and 2.2 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 preliminary numerical tests we disregarded the conditions of Assumptions 2.1 and 2.2 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 2.1 and 2.2, and that the number of violated inequalities was relatively low. 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 paper, although it seems a necessary step to further validate these indications.

Table 5: 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 2.1 and 2.2.

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

Our study was performed under the assumption of convexity of the functions involved. More-over, we also assumed that the approximate optimal solutions of the NLP subproblems were feasible in these subproblems, and that the corresponding inexact Lagrange multipliers were nonnegative. Relaxing these assumptions 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 miss classification of infeasibility can indeed cause numerical trouble. This is certainly another topic deserving a deeper study.

A Appendix

A.1 Inexact KKT conditions and perturbed problems

As we said in the Introduction of the paper, the point y^j satisfying the inexact KKT conditions (1)–(2) of the subproblem $\mathrm{NLP}(x^j)$ can be interpreted as a solution of a perturbed NLP subproblem, which has the form

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

where t^j is given by (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)–(5) of the subproblem $\mathrm{NLPF}(x^k)$ can be interpreted as a solution of the following perturbed NLP subproblem

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

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

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

A.2 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:

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

In fact, we will show next that the constraints in problem $\mathcal{P}^{\mathrm{GBD}\,1}$ imply those of $\mathcal{P}^{\mathrm{GBD}}$.

When $l \in T$, one knows that $\mathrm{NLP}(x^l)$ 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 (9) and (10)),

$$f(x,y) + (\lambda^l)^\top g(x,y) - (r^l)^\top (y - y^l) \geq f(x^l, y^l) + \nabla_x f(x^l, y^l)^\top (x - x^l) + \nabla_y f(x^l, y^l)^\top (y - y^l)$$
$$+ \sum_{i=1}^{m} \lambda_i^l [g_i(x^l, y^l) + \nabla_x g_i(x^l, y^l)^\top (x - x^l)$$
$$+ \nabla_y g_i(x^l, y^l)^\top (y - y^l)] - (r^l)^\top (y - y^l).$$

Thus, using the inexact KKT conditions (1),

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

The last equality holds due to (2).

When $l \in S$, we know that $\text{NLPF}(x^l)$ 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 (10)), we have that

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

Then, using the inexact KKT conditions (4),

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

In summary we have the following property.

Property A.1 *Given some sets T and S, the lower bound predicted by the master problem $\mathcal{P}^{\text{GBD}\,1}$ is greater than or equal to the one predicted by the master problem $\mathcal{P}^{\text{GBD}}$.*

References

[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] A. Ahlatçıoğlu and M. Guignard. Convex hull relaxation (CHR) for convex and nonconvex MINLP problems with linear constraints. 2011.

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

[5] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[6] 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.*, 5:186–204, 2008.

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

[8] P. Bonami and J. P. M. Gonçalves. Heuristics for convex mixed integer nonlinear programs. *Comput. Optim. Appl.*, pages 1–19, 2010.

[9] 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. *Computers and Chemical Engineering*, 30:54–69, 2005.

[10] M. A. Duran and I. E. Grossmann. An out-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Program.*, 36:307–339, 1986.

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

[12] O. E. Flippo and A. H. G. Rinnooy Kan. Decomposition in general mathematical programming. *Math. Program.*, 60:361–382, 1993.

[13] A. M. Geoffrion. Generalized Benders decomposition. *J. Optim. Theory Appl.*, 10:237–260, 1972.

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

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

[16] S. Leyffer. MacMINLP. `http://wiki.mcs.anl.gov/leyffer/index.php/MacMINLP`.

[17] G. Nannicini, P. Belotti, J. Lee, J. Linderoth, F. Margot, and A. Wächter. A probing algorithm for MINLP with failure prediction by SVM. Technical Report RC25103 (W1101-112), IBM Research, Computer Science, 2011.

[18] I. Quesada and I. E. Grossmann. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering*, 16:937–947, 1992.

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

[20] T. Westerlund and F. Pettersson. An extended cutting plane method for solving convex MINLP problems. *Computers and Chemical Engineering*, 19:131–136, 1995.

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

[22] T. Westerlund, H. Skrifvars, I. Harjunkoski, and R. Pörn. An extended cutting plane method for a class of non-convex MINLP problems. *Computers and Chemical Engineering*, 22:357–365, 1998.

Acknowledgement