Augmented Lagrangians with possible infeasibility and finite termination for global nonlinear programming*

E. G. Birgin† J. M. Martínez ‡ L. F. Prudente‡

June 25, 2012§

Abstract

In a recent paper, Birgin, Floudas and Martínez introduced an augmented Lagrangian method for global optimization. In their approach, augmented Lagrangian subproblems are solved using the αBB method and convergence to global minimizers was obtained assuming feasibility of the original problem. In the present research, the algorithm mentioned above will be improved in several crucial aspects. On the one hand, feasibility of the problem will not be required. Possible infeasibility will be detected in finite time by the new algorithms and optimal infeasibility results will be proved. On the other hand, finite termination results that guarantee optimality and/or feasibility up to any required precision will be provided. An adaptive modification in which subproblem tolerances depend on current feasibility and complementarity will also be given. The adaptive algorithm allows the augmented Lagrangian subproblems to be solved without requiring unnecessary potentially high precisions in the intermediate steps of the method, which improves the overall efficiency. Experiments showing how the new algorithms and results are related to practical computations will be given.

Key words: deterministic global optimization, augmented Lagrangians, nonlinear programming, algorithms, numerical experiments.

1 Introduction

Many practical models seek to solve global optimization problems involving continuous functions and constraints. Different aspects of the global optimization field and its applications may be found in several textbooks [16, 35, 39, 47, 64, 72, 74, 76] and review papers [36, 37, 56, 57].

Algorithms for solving non-trivial optimization problems are always iterative. Sometimes, for practical purposes, one only needs optimality properties at the limit points. In many other cases, one wishes to find an iterate x^k for which it can be proved that feasibility and optimality hold up to some previously established precision. Moreover, in the case that no feasible point exists, a certificate of infeasibility could also be required. In simple constrained cases, several well-known algorithms accomplish that purpose efficiently. This is the case of the αBB algorithm [1, 2, 3, 14], that has been used in [21] as subproblems solver in the context of an augmented Lagrangian method.

*This work was supported by PRONEX-CNPq/FAPERJ (E-26/111.449/2010-APQ1), FAPESP (2006/53768-0, 2009/10241-0, and 2010/10133-0) and CNPq (Grant 306802/2010-4).

†Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, São Paulo SP, Brazil. e-mail: egbirgin@ime.usp.br

‡Department of Applied Mathematics, Institute of Mathematics, Statistics and Scientific Computing, University of Campinas, Campinas, SP, Brazil. e-mails: martinez@ime.unicamp.br, lfprudente@gmail.com

§Typo corrected on July 25,2012. Revision made on December 20, 2012.

The algorithm introduced in [21] for constrained global optimization was based on the Powell-Hestenes-Rockafellar (PHR) augmented Lagrangian approach [46, 59, 61]. An implementation in which subproblems were solved by means of the αBB method was described and tested in [21]. The convergence theory of [21] assumes that the nonlinear programming problem is feasible and proves that limit points of sequences generated by the algorithm are ε-global minimizers, where ε is a given positive tolerance. However, a test for verifying ε-optimality at each iterate x^k was not provided. As a consequence, the stopping criterion employed in the numerical implementation was not directly related to ε-optimality and relied on heuristic considerations. This gap will be filled in the present paper. On the one hand, we will not restrict the range of applications to feasible problems. Infeasible cases may also be handled by the methods analyzed in our present contribution, where we will prove that possible infeasibility can be detected in finite time by means of a computable test. On the other hand, we will introduce a practical stopping criterion guaranteeing that, at the approximate solution provided by the algorithm, feasibility holds up to some prescribed tolerance and the objective function value is the optimal one up to tolerance ε.

We will present two versions of the main algorithm. The first coincides essentially with the one introduced in [21] and solves each subproblem with a precision ε_k that tends to zero. In the second version we employ an adaptive precision control that depends on the infeasibility of iterates of internal iterations. In this way, we aim at rapid detection of infeasibility, without solving expensive subproblems with unreliable precision. In the Local Optimization context this problem was considered in [54].

Besides providing practical stopping criteria, the new theoretical results shed light on algorithmic properties and suggest implementation improvements. It is well known that the presence of extreme penalty parameters makes the solution of subproblems in Penalty and augmented Lagrangian methods difficult. In fact, it may become very expensive to solve subproblems up to the desired precision, due to large norms of gradients and Hessians, which cause increasing work to solve subproblems. On the other hand, when the penalty parameter takes an extreme value, the shifts (quotients between multipliers and penalty parameters) employed in subproblems should obviously be close to zero. This justifies the practical decision of maintaining bounded multipliers. Attempts to avoid this algorithmic safeguard are theoretically interesting [53]. In the theory presented in this paper, the role of the norms of multipliers will appear very clearly.

Global optimization theory also clarifies practical algorithmic properties of "local" optimization algorithms, which tend to converge quickly to stationary points. We recall that the augmented Lagrangian methodology based on the PHR approach has been successfully used for defining practical nonlinear programming algorithms [5, 6, 19, 30]. In the local optimization field, which requires near-stationarity (instead of near global optimality) at subproblems, convergence to KKT points was proved using the Constant Positive Linear Dependence constraint qualification [11]. Convergence to KKT points also occurs under more general constraint qualifications recently introduced in [9, 10]. Convergence results involving sequential optimality conditions that do not need constraint qualifications at all were presented in [8, 12].

The Algencan code, available in *http://www.ime.usp.br/∼egbirgin/tango/* and based on the theory presented in [5], has been improved several times in the last few years [7, 18, 20, 24, 26, 25, 29] and, in practice, has been shown to converge to global minimizers more frequently than other Nonlinear Programming solvers. Derivative-free versions of Algencan were introduced in [31] and [49]. There exist many global optimization techniques for nonlinear programming problems, e.g., [2, 3, 4, 14, 38, 40, 41, 42, 43, 44, 50, 52, 53, 58, 62, 63, 65, 66, 67, 68, 69, 70, 73, 75]. The main appeal of the augmented Lagrangian approach in this context is that the structure of this method makes it possible to take advantage of global optimization algorithms for simpler problems, i.e. that a problem with difficult-to-handle constraints may be tackled by solving a sequence of subproblems with simpler constraints using some well-established existing method. In [21] and the present paper we exploit the ability of αBB to solve linearly constrained global optimization problems, which has been corroborated in many applied papers. In order to take advantage of the αBB potentialities, augmented Lagrangian subproblems are "over-restricted" by means of linear constraints that simplify subproblem resolutions and do not affect a successful search of global minimizers. Because of the necessity of dealing with infeasible problems, the definition of the additional constraints has been modified in the present contribution with respect to the one given in [21].

This paper is organized as follows. A first algorithm and its convergence theory will be presented in Section 2. Section 3 will be devoted to an improved version of the method that avoids the employment of an exogenous sequence of tolerances to declare convergence of the augmented Lagrangian subproblems. Section 4 will present numerical experiments and conclusions will be given in Section 5.

Notation. If $v \in I\!R^n$, $v = (v_1, \dots, v_n)$, we denote $v_+ = (\max\{0, v_1\}, \dots, \max\{0, v_n\})$. If $K = (k_1, k_2, \dots) \subseteq I\!N$ (with $k_j < k_{j+1}$ for all j), we denote $K \underset{\infty}{\subset} I\!N$. The symbol $\|\cdot\|$ will denote the Euclidean norm.

2 Algorithm

The problem considered in this paper is:

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{subject to} & h(x) = 0 \\ & g(x) \leq 0 \\ & x \in \Omega, \end{array} \tag{1}$$

where $h : I\!R^n \to I\!R^m, g : I\!R^n \to I\!R^p, f : I\!R^n \to I\!R$ are continuous and $\Omega \subset I\!R^n$ is compact. While the "hard" constraints are represented by $h(x) = 0$ and $g(x) \leq 0$, in general, Ω is defined by "easy" constraints such as linear constraints or box constraints. Since all the iterates x^k generated by our methods will belong to Ω, the constraints related to this set may be called "non-relaxable" in the sense of [15].

The augmented Lagrangian function [46, 59, 61] will be defined by:

$$L_\rho(x, \lambda, \mu) = f(x) + \frac{\rho}{2} \left\{ \sum_{i=1}^{m} \left[h_i(x) + \frac{\lambda_i}{\rho} \right]^2 + \sum_{i=1}^{p} \left[\max\left(0, g_i(x) + \frac{\mu_i}{\rho}\right) \right]^2 \right\} \tag{2}$$

for all $x \in \Omega, \rho > 0, \lambda \in I\!R^m, \mu \in I\!R^p_+$.

At each (outer) iteration, the algorithm considered in this section minimizes the augmented Lagrangian, with precision ε_k, on the set $\Omega \cap P_k$, where $P_k \subseteq I\!R^n$ is built in order to facilitate the work of a subproblem solver like αBB. There is no restriction about the global minimizers of the original problem or the subproblems being at the boundary of the feasible region or not. The assumptions required for the tolerances $\{\varepsilon_k\}$ and the auxiliary sets $\{P_k\}$ are given below.

Assumption A1. The sequence of positive tolerances $\{\varepsilon_k\}$ is bounded.

Assumption A2. The sets P_k are closed and the set of global minimizers of (1) is contained in P_k for all $k \in I\!N$. (Note that this assumption is trivially satisfied if problem (1) is infeasible and, hence, the set of global minimizers is empty.)

If the feasible set of (1) is contained in P_k for all k, Assumption A2 obviously holds. The sequence $\{\varepsilon_k\}$ may be defined in an external or an internal way, in different implementations. In the external case, the sequence is given as a parameter of the algorithm. If one decides for an internal definition, each tolerance ε_k depends on x^k, and is defined as a result of the process evolution. Observe that the existence of global minimizers is not guaranteed at all, since the feasible set could be empty. In [21] the existence of a global minimizer was an assumption on the problem and the sets P_k were assumed to contain at least one global minimizer.

In our implementations the sets P_k contain all the feasible points and, therefore, they contain all the solutions of the problem. The sets P_k are constructed by the method on the fly to cut, from the feasible set of the subproblems, infeasible points of the original problem. In order to do that, sets P_k are constructed using linear relaxations of the penalized nonlinear constraints. This procedure only eliminates infeasible points of the original problem. For the theory, it can be assumed that $P_k \equiv I\!R^n$ for all k.

For defining Algorithm 2.1, we assume that $\varepsilon_{\text{feas}} > 0$ and $\varepsilon_{\text{opt}} > 0$ are user-given tolerances for feasibility and optimality respectively.

Algorithm 2.1

Let $\lambda_{\min} < \lambda_{\max}$, $\mu_{\max} > 0$, $\gamma > 1$, $0 < \tau < 1$. Let $\lambda_i^1 \in [\lambda_{\min}, \lambda_{\max}], i = 1, \ldots, m$, $\mu_i^1 \in [0, \mu_{\max}], i = 1, \ldots, p$, and $\rho_1 > 0$. Assume that $\{\bar{\varepsilon}_k\}$ is a bounded positive sequence and initialize $k \leftarrow 1$.

Step 1 Solve the subproblem

Solve, using global optimization on the set $\Omega \cap P_k$, the subproblem

$$\text{Minimize } L_{\rho_k}(x, \lambda^k, \mu^k) \quad \text{subject to} \quad x \in \Omega \cap P_k. \tag{3}$$

If, in the process of solving (3), the set $\Omega \cap P_k$ is detected to be empty, stop the execution of Algorithm 2.1 declaring **Infeasibility**. Otherwise, define $x^k \in \Omega \cap P_k$ as an approximate solution of (3) that satisfies

$$L_{\rho_k}(x^k, \lambda^k, \mu^k) \le L_{\rho_k}(x, \lambda^k, \mu^k) + \varepsilon_k \tag{4}$$

for all $x \in \Omega \cap P_k$, for some $\varepsilon_k \le \bar{\varepsilon}_k$.

Step 2 Test Infeasibility

Compute $c_k > 0$ such that $|f(x^k) - f(z)| \le c_k$ for all $z \in \Omega \cap P_k$ and define

$$\gamma_k = \frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right].$$

If

$$\gamma_k + \varepsilon_k < -c_k,$$

stop the execution of the algorithm declaring **Infeasibility**.

Step 3 Test Feasibility and optimality

If

$$\|h(x^k)\| + \|g(x^k)_+\| \le \varepsilon_{\text{feas}} \quad \text{and} \quad \gamma_k + \varepsilon_k \le \varepsilon_{\text{opt}},$$

stop the execution of the algorithm declaring **Solution found**.

Step 4 Update penalty parameter

Define

$$V_i^k = \min\left\{ -g_i(x^k), \frac{\mu_i^k}{\rho_k} \right\}, i = 1, \ldots, p.$$

If $k = 1$ or

$$\max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\} \le \tau \ \max\{\|h(x^{k-1})\|_\infty, \|V^{k-1}\|_\infty\}, \tag{5}$$

define $\rho_{k+1} = \rho_k$. Otherwise, define $\rho_{k+1} = \gamma \rho_k$.

Step 5. Update multipliers

Compute $\lambda_i^{k+1} \in [\lambda_{\min}, \lambda_{\max}], i = 1, \ldots, m$ and $\mu_i^{k+1} \in [0, \mu_{\max}], i = 1, \ldots, p$. Set $k \leftarrow k + 1$ and go to Step 1.

The solvability of the subproblems (4) is guaranteed, if $\Omega \cap P_k$ is a bounded polytope, employing global optimization algorithms such as the αBB. By Assumption A2 if one of the sets $\Omega \cap P_k$ is found to be empty, we can conclude that the problem (1) is infeasible and Algorithm 2.1 stops at Step 1 declaring **Infeasibility**. Otherwise, we will see that the generated sequence is stopped, satisfying stopping criteria that guarantee feasibility and optimality, or, perhaps, infeasibility. In order to achieve these goals, we will prove first some results assuming that the generated sequence has, in fact, infinitely many terms. Later, we are going to see that this will never be the case, but the proved results in the infinite case will be useful for ensuring that, eventually, the finite stopping criteria hold.

Theorem 2.1. *Assume that $\{x^k\}$ is an infinite sequence generated by Algorithm 2.1. Let $K \underset{\infty}{\subset} I\!N$ and $x^* \in \Omega$ be such that $\lim_{k \in K} x^k = x^*$. (Such subsequence exists since Ω is compact.) Then, for all $z \in \Omega$ such that z is a limit point of $\{z^k\}_{k \in K}$, with $z^k \in \Omega \cap P_k$ for all $k \in K$, we have:*

$$\|h(x^*)\|^2 + \|g(x^*)_+\|^2 \le \|h(z)\|^2 + \|g(z)_+\|^2. \tag{6}$$

In particular, if the problem (1) is feasible, every limit point of an infinite sequence generated by Algorithm 2.1 is feasible.

Proof. In the case that $\{\rho_k\}$ is bounded, we have, by (5), that $\lim_{k \to \infty} \|h(x^k)\| + \|g(x^k)_+\| = 0$. Taking limits for $k \in K$ implies that $\|h(x^*)\| + \|g(x^*)_+\| = 0$, which trivially implies (6).

Consider now the case in which $\rho_k \to \infty$. Let $z \in \Omega$ and $K_1 \underset{\infty}{\subset} K$ be such that

$$\lim_{k \in K_1} z^k = z,$$

with $z^k \in \Omega \cap P_k$ for all $k \in K_1$. By (4), we have:

$$L_{\rho_k}(x^k, \lambda^k, \mu^k) \le L_{\rho_k}(z^k, \lambda^k, \mu^k) + \varepsilon_k$$

for all $k \in K_1$. This implies that, for all $k \in K_1$,

$$\frac{\rho_k}{2}\left[\left\|h(x^k)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(x^k)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le \frac{\rho_k}{2}\left[\left\|h(z^k)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(z^k)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right]+\varepsilon_k+f(z^k)-f(x^k).$$

Therefore,

$$\left\|h(x^k)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(x^k)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2 \le \left[\left\|h(z^k)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(z^k)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right]+\frac{2(\varepsilon_k + f(z^k) - f(x^k))}{\rho_k}.$$

Since $\{\varepsilon_k\}$, $\{\lambda^k\}$, $\{\mu^k\}$ are bounded, ρ_k tends to infinity, and Ω is compact, the inequality (6) follows, taking limits for $k \in K_1$, by the continuity of f, h, and g. $\square$

In the case that $\Omega \subseteq P_k$ for all k, Theorem 2.1 says that any limit point is a global minimizer of the infeasibility measure $\|h(x)\|^2 + \|g(x)_+\|^2$ onto Ω. It is interesting to observe that the tolerances ε_k do not necessarily tend to zero, in order to obtain the thesis of Theorem 2.1. Moreover, although in the algorithm we assume that λ^k and μ^k are bounded, in the proof we only need that the quotients λ^k/ρ_k and μ^k/ρ_k tend to zero as ρ_k tends to infinity.

In the following theorem we prove that infeasibility can be detected in finite time. Let us define, for all $k \in I\!N$, $c_k > 0$ by:

$$|f(z) - f(x^k)| \leq c_k \quad \text{for all} \quad z \in \Omega \cap P_k. \tag{7}$$

Note that c_k may be computed using interval calculations as in the αBB algorithm. Clearly, since f is continuous and Ω is bounded, the sequence $\{c_k\}$ may be assumed to be bounded. Observe that, as in the case of Theorem 2.1, for proving Theorem 2.2 we do not need that $\varepsilon_k \to 0$.

Theorem 2.2. *Assume that $\{x^k\}$ is an infinite sequence generated by Algorithm 2.1 and, for all $k \in I\!N$, the set $\Omega \cap P_k$ is non-empty. Then, the problem (1) is infeasible if and only if there exists $k \in I\!N$ such that*

$$\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] + \varepsilon_k < -c_k. \tag{8}$$

Proof. Suppose that the feasible region of (1) is non-empty. Then there exists a global minimizer z such that $z \in \Omega \cap P_k$ for all $k \in I\!N$. Therefore,

$$f(x^k)+\frac{\rho_k}{2}\left[\left\|h(x^k)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(x^k)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \leq f(z)+\frac{\rho_k}{2}\left[\left\|h(z)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(z)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right]+\varepsilon_k.$$

Thus,

$$\frac{\rho_k}{2}\left[\left\|h(z)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(z)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right]-\frac{\rho_k}{2}\left[\left\|h(x^k)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(x^k)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \geq f(x^k)-f(z)-\varepsilon_k. \tag{9}$$

Since $h(z) = 0$ and $g(z) \leq 0$, we have:

$$\left\|h(z) + \frac{\lambda^k}{\rho_k}\right\|^2 = \left\|\frac{\lambda^k}{\rho_k}\right\|^2 \text{ and } \left\|\left(g(z) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2 \leq \left\|\frac{\mu^k}{\rho_k}\right\|^2.$$

Then, by (9),

$$\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \geq f(x^k) - f(z) - \varepsilon_k.$$

Therefore, by (7),

$$\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] + \varepsilon_k \geq -c_k$$

for all $k \in I\!N$. This means that the infeasibility test (8) fails to be fulfilled for all $k \in I\!N$.

Reciprocally, suppose that problem (1) is infeasible. In this case ρ_k tends to infinity. This implies that the sequence $\{x^k\}$ admits an infeasible limit point $x^* \in \Omega$. So, for some subsequence, the quantity $\|h(x^k) + \lambda^k/\rho_k\|^2 + \|(g(x^k) + \mu^k/\rho_k)_+\|^2$ is bounded away from zero. Since

$$-\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{2(\varepsilon_k + c_k)}{\rho_k}$$

tends to zero, it turns out that, for k large enough, the test (8) is fulfilled. $\square$

In the following theorem we prove another asymptotic convergence result, this time connected with optimality, instead of feasibility. Strictly speaking, this result coincides with the one presented in Theorem 2 of [21]. However, we decided to include a different proof here because some of the intermediate steps will be evoked in forthcoming results.

Theorem 2.3. *Assume that $\{x^k\}$ is an infinite sequence generated by Algorithm 2.1, $\lim_{k\to\infty} \varepsilon_k = 0$, and problem (1) is feasible. Then, every limit point of $\{x^k\}$ is a global solution of (1).*

Proof. Let $K \underset{\infty}{\subset} I\!N$ and $x^* \in \Omega$ be such that $\lim_{k\in K} x^k = x^*$. Since the feasible set is non-empty and compact, problem (1) admits a global minimizer $z \in \Omega$. By Assumption A2, $z \in P_k$ for all $k \in I\!N$. We consider two cases: $\rho_k \to \infty$ and $\{\rho_k\}$ bounded.

Case 1 ($\rho_k \to \infty$): By the definition of the algorithm:

$$f(x^k)+\frac{\rho_k}{2}\left[\left\|h(x^k)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(x^k)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le f(z)+\frac{\rho_k}{2}\left[\left\|h(z)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(z)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right]+\varepsilon_k \tag{10}$$

for all $k \in I\!N$. Since $h(z) = 0$ and $g(z) \le 0$, we have:

$$\left\|h(z) + \frac{\lambda^k}{\rho_k}\right\|^2 = \left\|\frac{\lambda^k}{\rho_k}\right\|^2 \text{ and } \left\|\left(g(z) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2 \le \left\|\frac{\mu^k}{\rho_k}\right\|^2.$$

Therefore, by (10),

$$f(x^k) \le f(x^k) + \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le f(z) + \frac{\|\lambda^k\|^2}{2\rho_k} + \frac{\|\mu^k\|^2}{2\rho_k} + \varepsilon_k.$$

Taking limits for $k \in K$, using that $\lim_{k\in K} \|\lambda^k\|^2/\rho_k = \lim_{k\in K} \|\mu^k\|^2/\rho_k = 0$, and $\lim_{k\in K} \varepsilon_k = 0$, by the continuity of f and the convergence of x^k, we get:

$$f(x^*) \le f(z).$$

Since z is a global minimizer, it turns out that x^* is a global minimizer, as we wanted to prove.

Case 2 ($\{\rho_k\}$ bounded): In this case, we have that $\rho_k = \rho_{k_0}$ for all $k \ge k_0$. Therefore, by the definition of Algorithm 2.1, we have:

$$f(x^k)+\frac{\rho_{k_0}}{2}\left[\left\|h(x^k)+\frac{\lambda^k}{\rho_{k_0}}\right\|^2+\left\|\left(g(x^k)+\frac{\mu^k}{\rho_{k_0}}\right)_+\right\|^2\right] \le f(z)+\frac{\rho_{k_0}}{2}\left[\left\|h(z)+\frac{\lambda^k}{\rho_{k_0}}\right\|^2+\left\|\left(g(z)+\frac{\mu^k}{\rho_{k_0}}\right)_+\right\|^2\right]+\varepsilon_k$$

for all $k \geq k_0$. Since $g(z) \leq 0$ and $\mu^k/\rho_{k_0} \geq 0$,

$$\left\| \left(g(z) + \frac{\mu^k}{\rho_{k_0}} \right)_+ \right\|^2 \leq \left\| \frac{\mu^k}{\rho_{k_0}} \right\|^2.$$

Thus, since $h(z) = 0$,

$$f(x^k) + \frac{\rho_{k_0}}{2} \left[\left\| h(x^k) + \frac{\lambda^k}{\rho_{k_0}} \right\|^2 + \left\| \left(g(x^k) + \frac{\mu^k}{\rho_{k_0}} \right)_+ \right\|^2 \right] \leq f(z) + \frac{\rho_{k_0}}{2} \left[\left\| \frac{\lambda^k}{\rho_{k_0}} \right\|^2 + \left\| \frac{\mu^k}{\rho_{k_0}} \right\|^2 \right] + \varepsilon_k \quad (11)$$

for all $k \geq k_0$. Let us now take $\varepsilon > 0$ arbitrarily small. Suppose, for a moment, that $g_i(x^*) < 0$. Since $\lim_{k\to\infty} \min\{-g_i(x^k), \mu_i^k/\rho_{k_0}\} = 0$, we have that

$$\lim_{k\in K} \mu_i^k/\rho_{k_0} = 0. \quad (12)$$

This implies that $(g_i(x^k) + \mu_i^k/\rho_{k_0})_+ = 0$ for $k \in K$ large enough. Therefore, for $k \in K$ large enough, $\sum_{i=1}^p (g_i(x^k) + \mu_i^k/\rho_{k_0})_+^2 = \sum_{g_i(x^*)=0} (g_i(x^k) + \mu_i^k/\rho_{k_0})_+^2$. Thus, by (11), for $k \in K$ large enough we have:

$$f(x^k) + \frac{\rho_{k_0}}{2} \left[\sum_{i=1}^m \left(h_i(x^k) + \frac{\lambda_i^k}{\rho_{k_0}} \right)^2 + \sum_{g_i(x^*)=0} \left(g_i(x^k) + \frac{\mu_i^k}{\rho_{k_0}} \right)_+^2 \right]$$

$$\leq f(z) + \frac{\rho_{k_0}}{2} \left[\sum_{i=1}^m \left(\frac{\lambda_i^k}{\rho_{k_0}} \right)^2 + \sum_{g_i(x^*)=0} \left(\frac{\mu_i^k}{\rho_{k_0}} \right)^2 + \sum_{g_i(x^*)<0} \left(\frac{\mu_i^k}{\rho_{k_0}} \right)^2 \right] + \varepsilon_k.$$

By (12), we deduce that, for $k \in K$ large enough,

$$f(x^k) + \frac{\rho_{k_0}}{2} \left[\sum_{i=1}^m \left(h_i(x^k) + \frac{\lambda_i^k}{\rho_{k_0}} \right)^2 + \sum_{g_i(x^*)=0} \left(g_i(x^k) + \frac{\mu_i^k}{\rho_{k_0}} \right)_+^2 \right]$$

$$\leq f(z) + \frac{\rho_{k_0}}{2} \left[\sum_{i=1}^m \left(\frac{\lambda_i^k}{\rho_{k_0}} \right)^2 + \sum_{g_i(x^*)=0} \left(\frac{\mu_i^k}{\rho_{k_0}} \right)^2 \right] + \varepsilon_k + \varepsilon. \quad (13)$$

For $k \in K$ large enough, by the boundedness of λ_i^k/ρ_{k_0} and the fact that $h(x^k) \to 0$, we have that

$$\frac{\rho_{k_0}}{2} \sum_{i=1}^m \left[h_i(x^k)^2 + 2h_i(x^k) \frac{\lambda_i^k}{\rho_{k_0}} \right] \geq -\varepsilon.$$

Therefore, by (13),

$$f(x^k) + \frac{\rho_{k_0}}{2} \left[\sum_{i=1}^m \left(\frac{\lambda_i^k}{\rho_{k_0}} \right)^2 + \sum_{g_i(x^*)=0} \left(g_i(x^k) + \frac{\mu_i^k}{\rho_{k_0}} \right)_+^2 \right] \leq f(z) + \frac{\rho_{k_0}}{2} \left[\sum_{i=1}^m \left(\frac{\lambda_i^k}{\rho_{k_0}} \right)^2 + \sum_{g_i(x^*)=0} \left(\frac{\mu_i^k}{\rho_{k_0}} \right)^2 \right] + \varepsilon_k + 2\varepsilon.$$

Thus, there exists $k_1 \geq k_0$ such that for all $k \in K$ such that $k \geq k_1$, we have that

$$f(x^k) + \frac{\rho_{k_0}}{2} \left[\sum_{g_i(x^*)=0} \left(g_i(x^k) + \frac{\mu_i^k}{\rho_{k_0}} \right)_+^2 \right] \leq f(z) + \frac{\rho_{k_0}}{2} \left[\sum_{g_i(x^*)=0} \left(\frac{\mu_i^k}{\rho_{k_0}} \right)^2 \right] + \varepsilon_k + 2\varepsilon. \quad (14)$$

Define

$$I = \{i \in \{1, \dots, p\} \mid g_i(x^*) = 0\}$$

and

$$K_1 = \{k \in K \mid k \geq k_1\}.$$

For each $i \in I$, we define

$$K_+(i) = \{k \in K_1 \mid g_i(x^k) + \mu_i^k/\rho_{k_0} \geq 0\}$$

and

$$K_-(i) = \{k \in K_1 \mid g_i(x^k) + \mu_i^k/\rho_{k_0} < 0\}.$$

Obviously, for all $i \in I$, $K_1 = K_+(i) \cup K_-(i)$. Let us fix $i \in I$. For k large enough, since $g_i(x^*) = 0$, by the continuity of g_i and the boundedness of μ_i^k/ρ_{k_0}, we have that:

$$\frac{\rho_{k_0}}{2}\left(g_i(x^k)^2 + \frac{2g_i(x^k)\mu_i^k}{\rho_{k_0}}\right) \geq -\varepsilon.$$

Therefore,

$$\frac{\rho_{k_0}}{2}\left[g_i(x^k)^2 + \frac{2g_i(x^k)\mu_i^k}{\rho_{k_0}} + \left(\frac{\mu_i^k}{\rho_{k_0}}\right)^2\right] \geq \frac{\rho_{k_0}}{2}\left(\frac{\mu_i^k}{\rho_{k_0}}\right)^2 - \varepsilon.$$

Thus, for $k \in K_+(i)$ large enough,

$$\frac{\rho_{k_0}}{2}\left(g_i(x^k) + \frac{\mu_i^k}{\rho_{k_0}}\right)_+^2 \geq \frac{\rho_{k_0}}{2}\left(\frac{\mu_i^k}{\rho_{k_0}}\right)^2 - \varepsilon. \tag{15}$$

Now, if $k \in K_-(i)$, we have that $-g_i(x^k) > \mu_i^k/\rho_{k_0}$. So, since $g_i(x^k)$ tends to zero, for $k \in K_-(i)$ large enough we have that $(\rho_{k_0}/2)(\mu_i^k/\rho_{k_0})^2 \leq \varepsilon$. Therefore,

$$\frac{\rho_{k_0}}{2}\left(g_i(x^k) + \frac{\mu_i^k}{\rho_{k_0}}\right)_+^2 = 0 \geq \frac{\rho_{k_0}}{2}\left(\frac{\mu_i^k}{\rho_{k_0}}\right)^2 - \varepsilon. \tag{16}$$

Combining (15) and (16) and taking k large enough, we obtain:

$$f(x^k) + \frac{\rho_{k_0}}{2}\left[\sum_{g_i(x^*)=0}\left(g_i(x^k) + \frac{\mu_i^k}{\rho_{k_0}}\right)_+^2\right] \geq f(x^k) + \frac{\rho_{k_0}}{2}\left[\sum_{g_i(x^*)=0}\left(\frac{\mu_i^k}{\rho_{k_0}}\right)^2\right] - p\varepsilon. \tag{17}$$

Then, by (14) and (17), for $k \in K$ large enough we have that

$$f(x^k) \leq f(z) + \varepsilon_k + (2 + p)\varepsilon.$$

Since $\lim_{k\in K} \varepsilon_k = 0$ and ε is arbitrarily small, we have that $\lim_{k\in K} f(x^k) \leq f(z)$. By the continuity of f, it follows that $f(x^*) \leq f(z)$ and, since x^* is feasible, we have that x^* is a global minimizer as we wanted to prove. $\square$

The following theorem establishes a sufficient computable condition guaranteeing that $f(x^k)$ is close to (and perhaps smaller than) the best possible value of $f(z)$ in the feasible set. Again, $\varepsilon_k \to 0$ is not used in its proof.

Theorem 2.4. *Assume that $\{x^k\}$ is an infinite sequence generated by Algorithm 2.1. Let $\varepsilon \in I\!R$ (possibly negative) and $k \in I\!N$ such that*

$$\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le \varepsilon. \tag{18}$$

Then

$$f(x^k) \le f(z) + \varepsilon + \varepsilon_k, \tag{19}$$

for all global minimizer z.

Proof. Let $z \in \Omega$ be a global minimizer of (1). By Assumption A2, $z \in P_k$ for all $k \in I\!N$. By the definition of Algorithm 2.1, we have that

$$f(x^k)+\frac{\rho_k}{2}\left[\left\|h(x^k)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(x^k)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le f(z)+\frac{\rho_k}{2}\left[\left\|h(z)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(z)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right]+\varepsilon_k \tag{20}$$

for all $k \in I\!N$. Moreover, since

$$\left\|h(z) + \frac{\lambda^k}{\rho_k}\right\|^2 = \left\|\frac{\lambda^k}{\rho_k}\right\|^2 \text{ and } \left\|\left(g(z) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2 \le \left\|\frac{\mu^k}{\rho_k}\right\|^2, \tag{21}$$

we obtain:

$$f(x^k) + \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le f(z) + \frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] + \varepsilon_k. \tag{22}$$

Assuming that (18) takes place, we have

$$f(x^k) + \frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \varepsilon \le f(x^k) + \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right]. \tag{23}$$

Hence, by (22) and (23), we have

$$f(x^k) + \frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \varepsilon \le f(z) + \frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] + \varepsilon_k. \tag{24}$$

Simplifying the expression (24), we obtain:

$$f(x^k) \le f(z) + \varepsilon + \varepsilon_k,$$

as we wanted to prove. □

In the following theorem we prove that the inequality (18), employed in Theorem 2.4 as a sufficient condition, eventually holds for some iterate k, if we assume that $\varepsilon > 0$ and $\{\varepsilon_k\}$ tends to zero.

Theorem 2.5. *Assume that $\{x^k\}$ is an infinite sequence generated by Algorithm 2.1. Suppose that (1) is feasible and $\lim_{k\to\infty} \varepsilon_k = 0$. Let ε be an arbitrary positive number. Then, there exists $k \in I\!N$ such that*

$$\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le \varepsilon. \tag{25}$$

Proof. By the compactness of Ω, there exists $K \underset{\infty}{\subset} I\!N$ and $x^* \in \Omega$ such that $\lim_{k\in K} x^k = x^*$ and, by Theorem 2.1, x^* is feasible. Suppose that ρ_k tends to infinity. Note that the left-hand side of (25) is bounded by $(\|\lambda^k\|^2 + \|\mu^k\|^2)/(2\rho_k)$ which tends to zero, by the boundedness of $\{\lambda^k\}$ and $\{\mu^k\}$. Thus, we obtain (25) for k large enough.

Consider now the case in which $\{\rho_k\}$ is bounded. For all $i = 1, \ldots, m$ we have that $(\rho_k/2)[h_i(x^k) + \lambda_i^k/\rho_k]^2 = (\rho_k/2)[h_i(x^k)^2 + 2h_i(x^k)\lambda_i^k/\rho_k + (\lambda_i^k/\rho_k)^2]$. Since $\{\rho_k\}$ is bounded, $\{\lambda^k\}$ is bounded, and $h_i(x^k) \to 0$ there exists $k_0(i) \in K$ such that $(\rho_k/2)[h_i(x^k) + \lambda_i^k/\rho_k]^2 \geq (\rho_k/2)(\lambda_i^k/\rho_k)^2 - \varepsilon/(2m)$ for all $k \in K$, $k \geq k_0(i)$. Taking $k_0 = \max\{k_0(i)\}$ we obtain that, for all $k \in K, k \geq k_0$, $i = 1, \ldots, m$,

$$\frac{\rho_k}{2}\left(\frac{\lambda_i^k}{\rho_k}\right)^2 - \frac{\rho_k}{2}\left(h_i(x^k) + \frac{\lambda_i^k}{\rho_k}\right)^2 \leq \frac{\varepsilon}{2m}. \tag{26}$$

Assume that $g_i(x^*) < 0$. Then, as in Case 2 of the proof of Theorem 2.3, since

$$\lim_{k\to\infty} \min\{-g_i(x^k), \mu_i^k/\rho_k\} = 0,$$

we have that $\lim_{k\in K} \mu_i^k/\rho_k = 0$. Thus, there exists $k_1(i) \geq k_0$ such that $(g_i(x^k) + \mu_i^k/\rho_k)_+ = 0$ for all $k \in K, k \geq k_1(i)$. Therefore, since $\mu_i^k/\rho_k \to 0$, there exists $k_2(i) \geq k_1(i)$ such that

$$\frac{\rho_k}{2}\left(\frac{\mu_i^k}{\rho_k}\right)^2 - \frac{\rho_k}{2}\left(g_i(x^k) + \frac{\mu_i^k}{\rho_k}\right)_+^2 \leq \frac{\varepsilon}{2p} \tag{27}$$

for all $k \in K, k \geq k_2(i)$. Taking $k_2 = \max\{k_2(i)\}$, we obtain that (27) holds for all $k \in K, k \geq k_2$ whenever $g_i(x^*) < 0$.

Now, as in the proof of Theorem 2.3, define

$$I = \{i \in \{1, \ldots, p\} \mid g_i(x^*) = 0\}$$

and

$$K_1 = \{k \in K \mid k \geq k_2\}.$$

For each $i \in I$, we define

$$K_+(i) = \{k \in K_1 \mid g_i(x^k) + \mu_i^k/\rho_k \geq 0\}$$

and

$$K_-(i) = \{k \in K_1 \mid g_i(x^k) + \mu_i^k/\rho_k < 0\}.$$

Let us fix $i \in I$. For $k \in K_1$ large enough, since $g_i(x^*) = 0$, by the continuity of g_i and the boundedness of μ_i^k/ρ_k, we have that:

$$\frac{\rho_k}{2}\left(g_i(x^k)^2 + \frac{2g_i(x^k)\mu_i^k}{\rho_k}\right) \geq -\frac{\varepsilon}{2p}.$$

Therefore,

$$\frac{\rho_k}{2}\left[g_i(x^k)^2 + \frac{2g_i(x^k)\mu_i^k}{\rho_k} + \left(\frac{\mu_i^k}{\rho_k}\right)^2\right] \geq \frac{\rho_k}{2}\left(\frac{\mu_i^k}{\rho_k}\right)^2 - \frac{\varepsilon}{2p}.$$

Thus, for $k \in K_+(i)$ large enough,

$$\frac{\rho_k}{2}\left(\frac{\mu_i^k}{\rho_k}\right)^2 - \frac{\rho_k}{2}\left(g_i(x^k) + \frac{\mu_i^k}{\rho_k}\right)_+^2 \leq \frac{\varepsilon}{2p}. \tag{28}$$

Now, if $k \in K_-(i)$, we have that $-g_i(x^k) > \mu_i^k/\rho_k$. So, since $g_i(x^k)$ tends to zero, for $k \in K_-(i)$ large enough we have that $(\rho_k/2)(\mu_i^k/\rho_k)^2 \leq \varepsilon/(2p)$. Therefore,

$$\frac{\rho_k}{2}\left(\frac{\mu_i^k}{\rho_k}\right)^2 - \frac{\rho_k}{2}\left(g_i(x^k) + \frac{\mu_i^k}{\rho_k}\right)_+^2 \leq \frac{\varepsilon}{2p}. \tag{29}$$

.

By (27), (28), and (29),

$$\frac{\rho_k}{2}\left(\frac{\mu_i^k}{\rho_k}\right)^2 - \frac{\rho_k}{2}\left(g_i(x^k) + \frac{\mu_i^k}{\rho_k}\right)_+^2 \leq \frac{\varepsilon}{2p} \tag{30}$$

for all $i = 1, \ldots, p$.

Taking the summation for $i = 1, \ldots, m$ in (26) and for $i = 1, \ldots, p$ in (30) we obtain the desired result. $\square$

Theorem 2.6 is our final result in this section. We prove that, in a finite number of iterations, Algorithm 2.1 finishes with a certificate of infeasibility, or finds a feasible point with tolerance $\varepsilon_{\text{feas}}$ such that its objective function value is optimal with tolerance ε_{opt}. We will assume that $\lim_{k\to\infty} \varepsilon_k = 0$, a condition that can of course be guaranteed if the external tolerance sequence $\bar{\varepsilon}_k$ tends to zero.

Theorem 2.6. *Assume that Algorithm 2.1 is executed with the condition that* $\lim_{k\to\infty} \varepsilon_k = 0$. *Then, the execution finishes in a finite number of iterations with one of the following diagnostics:*

1. **Infeasibility**, *which means that, with guarantee, no feasible point of (1) exists;*

2. **Solution found**, *in the case that the final point x^k is guaranteed to satisfy*

$$\|h(x^k)\| + \|g(x^k)_+\| \leq \varepsilon_{\text{feas}}$$

and

$$f(x^k) \leq f(z) + \varepsilon_{\text{opt}}$$

for all $z \in \Omega$ such that $h(z) = 0$ and $g(z) \leq 0$.

Proof. The proof follows straightforwardly from Theorems 2.2, 2.4, and 2.5. $\square$

3 Adaptive precision variation of the main algorithm

The algorithm defined in this section is variation of Algorithm 2.1, where

$$\varepsilon_k = O(\|h(x^k)\| + \|g(x^k)_+\| + \sum_{i=1}^{p} |\min\{-g_i(x^k), \mu_i^k/\rho_k\}|).$$

From now on, we denote

$$W_k = \|h(x^k)\| + \|g(x^k)_+\| + \sum_{i=1}^{p} |\min\{-g_i(x^k), \mu_i^k/\rho_k\}| \tag{31}$$

and

$$W_{k,\ell} = \|h(x^{k,\ell})\| + \|g(x^{k,\ell})_+\| + \sum_{i=1}^{p} |\min\{-g_i(x^{k,\ell}), \mu_i^k/\rho_k\}|. \tag{32}$$

As a consequence, if W_k tends to zero, the new algorithms will find solutions up to an arbitrarily high precision. On the other hand, if the problem is infeasible, W_k will be bounded away from zero, no high precision solutions will be demanded from the subproblem solver, and infeasibility will be detected in finite time. Since the stopping tolerance ε_k at each subproblem depends on x^k, subproblems may exist at which infinitely many iterations could be necessary to obtain a solution. In this case, the subproblem solver will never stop. Fortunately, we will prove that, at such a subproblem, some inner iterate generated by the subproblem solver will necessarily be a solution with the required precision. Adaptive precision solution of subproblems have been already used by some authors in the context of local optimization and quadratic programming. See, among others, [33, 34, 45, 54].

Let $c > 0$ be a given constant. Algorithm 2.1, with $\varepsilon_k = cW_k$, will be called here Algorithm 3.1. We assume that the subproblem is solved by means of an iterative global optimization method that generates a sequence $\{x^{k,\ell}\}_{\ell \in I\!N}$ such that

$$L_{\rho_k}(x^{k,\ell}, \lambda^k, \mu^k) \le L_{\rho_k}(z, \lambda^k, \mu^k) + \varepsilon_{k,\ell}, \tag{33}$$

for all $z \in \Omega \cap P_k$, where $\lim_{\ell \to \infty} \varepsilon_{k,\ell} = 0$. If, for some ℓ, we obtain that

$$L_{\rho_k}(x^{k,\ell}, \lambda^k, \mu^k) \le L_{\rho_k}(z, \lambda^k, \mu^k) + cW_{k,\ell}, \tag{34}$$

we define $x^k = x^{k,\ell}$, $W_k = W_{k,\ell}$, and $\varepsilon_k = cW_k$, we stop the execution of the subproblem solver and we return to the main algorithm.

However, the possibility remains that, for all $\ell \in I\!N$, there exists $z \in \Omega \cap P_k$ such that

$$L_{\rho_k}(z, \lambda^k, \mu^k) + cW_{k,\ell} \le L_{\rho_k}(x^{k,\ell}, \lambda^k, \mu^k) \le L_{\rho_k}(z, \lambda^k, \mu^k) + \varepsilon_{k,\ell}. \tag{35}$$

This means that, although subproblems are solved up to arbitrarily small precisions, the precision attained is always bigger than cW_k. In principle, the subproblem solver will never stop in this case. Clearly, (35) implies that, for all $\ell \in I\!N$,

$$cW_{k,\ell} \le \varepsilon_{k,\ell}. \tag{36}$$

Therefore,

$$\lim_{\ell \to \infty} \|h(x^{k,\ell})\| + \|g(x^{k,\ell})_+\| = \lim_{\ell \to \infty} \sum_{i=1}^{p} \left| \min \left\{ -g_i(x^{k,\ell}), \frac{\mu_i^k}{\rho_k} \right\} \right| = 0. \tag{37}$$

At this point, it is convenient to give a formal definition of Algorithm 3.1. We will assume, from now on in this section, that Assumption A2 holds. We emphasize that Assumption A2 is assumed to hold, but no external optimality tolerances $\{\varepsilon_k\}$ exist at all. The algorithm will return with a message of guaranteed infeasibility, or with x^k that is feasible with a given tolerance $\varepsilon_{\rm feas} > 0$ and optimal in the sense that $f(x^k)$ is smaller than or equal to $f(z) + \varepsilon_{\rm opt}$ for all feasible z.

Algorithm 3.1

Let $c > 0$, $\lambda_{\min} < \lambda_{\max}$, $\mu_{\max} > 0$, $\gamma > 1$, $0 < \tau < 1$. Let $\lambda_i^1 \in [\lambda_{\min}, \lambda_{\max}], i = 1, \dots, m$, $\mu_i^1 \in [0, \mu_{\max}], i = 1, \dots, p$, and $\rho_1 > 0$. Initialize $k \leftarrow 1$.

Step 1 Solve the subproblem

By means of global optimization on the set $\Omega \cap P_k$ (typically, the αBB algorithm), address the subproblem

$$\text{Minimize } L_{\rho_k}(x, \lambda^k, \mu^k) \quad \text{subject to} \quad x \in \Omega \cap P_k. \tag{38}$$

If, in the process of solving (38), the set $\Omega \cap P_k$ is detected to be empty, stop the execution of Algorithm 3.2 declaring **Infeasibility**. Otherwise, we assume that the subproblem solver generates a sequence $\{x^{k,0}, x^{k,1}, x^{k,2}, \dots\}$ such that, for all $\ell \in I\!N$,

$$L_{\rho_k}(x^{k,\ell}, \lambda^k, \mu^k) \leq L_{\rho_k}(z, \lambda^k, \mu^k) + \varepsilon_{k,\ell} \;\text{ for all }\; z \in \Omega \cap P_k, \tag{39}$$

where

$$\lim_{\ell \to \infty} \varepsilon_{k,\ell} = 0. \tag{40}$$

At each iteration ℓ of the subproblem solver we compute $W_{k,\ell}$ as in (32) and we perform the test (34). Note that, by (39), it is enough to test whether

$$\varepsilon_{k,\ell} \leq c W_{k,\ell}. \tag{41}$$

If (41), and, hence, (34) holds, we define $x^k = x^{k,\ell}$, $W_k = W_{k,\ell}$, $\varepsilon_k = \varepsilon_{k,\ell}$ and we go to Step 2.

If (34) is not guaranteed to hold at iteration ℓ of the subproblem solver, define

$$\gamma_{k,\ell} = \frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^{k,\ell}) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^{k,\ell}) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right].$$

If

$$\|h(x^{k,\ell})\| + \|g(x^{k,\ell})_+\| \leq \varepsilon_{\text{feas}} \quad \text{and} \quad \gamma_{k,\ell} + \varepsilon_{k,\ell} \leq \varepsilon_{\text{opt}}, \tag{42}$$

define $x^k = x^{k,\ell}$ and stop the execution of Algorithm 3.2 declaring **Solution found**. Otherwise, the execution of the subproblem solver continues with iterate $\ell + 1$.

Step 2 Test Infeasibility

Compute $c_k > 0$ such that $|f(z) - f(x^k)| \leq c_k$ for all $z \in \Omega \cap P_k$ and define

$$\gamma_k = \frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right].$$

If

$$\gamma_k + \varepsilon_k < -c_k,$$

stop the execution of the algorithm declaring **Infeasibility**.

Step 3 Test Feasibility and optimality

If

$$\|h(x^k)\| + \|g(x^k)_+\| \leq \varepsilon_{\text{feas}} \quad \text{and} \quad \gamma_k + \varepsilon_k \leq \varepsilon_{\text{opt}},$$

stop the execution of the algorithm declaring **Solution found**.

Step 4 Update penalty parameter

Define
$$V_i^k = \min\left\{ -g_i(x^k), \frac{\mu_i^k}{\rho_k}\right\}, i = 1, \ldots, p.$$

If $k = 1$ or
$$\max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\} \leq \tau \ \max\{\|h(x^{k-1})\|_\infty, \|V^{k-1}\|_\infty\}, \tag{43}$$
define $\rho_{k+1} = \rho_k$. Otherwise, define $\rho_{k+1} = \gamma\rho_k$.

Step 5 Update multipliers

Compute $\lambda_i^{k+1} \in [\lambda_{\min}, \lambda_{\max}], i = 1, \ldots, m$ and $\mu_i^{k+1} \in [0, \mu_{\max}], i = 1, \ldots, p$. Set $k \leftarrow k + 1$ and go to Step 1.

In the following theorem, we deal with the behavior of Algorithm 3.1 when $\Omega \cap P_k$ is non-empty for all k and all the subproblems are solved, satisfying (34) for some finite value of ℓ. Similarly to what has been done in Section 2 for the Algorithm 2.1, we will prove some asymptotic properties related with the generated sequence by Algorithm 3.1. When we mention that $\{x^k\}$ is an infinite sequence generated by Algorithm 3.1, the reader may think that $\{x^k\}$ was generated by Algorithm 3.1 omitting Step 2 and Step 3. We show that, in this case, every limit point is feasible, or is a minimizer of infeasibility in the sense of Theorem 2.1.

Theorem 3.1. *Assume that $\{x^k\}$ is an infinite sequence generated by Algorithm 3.1. Let $K \underset{\infty}{\subset} I\!N$ and $x^* \in \Omega$ be such that $\lim_{k\in K} x^k = x^*$. Then, for all $z \in \Omega$ such that z is a limit point of $\{z^k\}_{k\in K}$, with $z^k \in \Omega \cap P_k$ for all $k \in K$, we have:*
$$\|h(x^*)\|^2 + \|g(x^*)_+\| \leq \|h(z)\|^2 + \|g(z)_+\|^2. \tag{44}$$
In particular, if problem (1) is feasible, every limit point of $\{x^k\}$ is feasible too.

Proof. Define, for all $k \in I\!N$,
$$\varepsilon_k = cW_k.$$
Since $x^k \in \Omega$ for all k, Ω is compact, $\{\lambda^k/\rho_k\}$ and $\{\mu^k/\rho_k\}$ are bounded, and the constraint functions are continuous, we have that the sequence $\{W_k\}$ is bounded. Therefore, $\{\varepsilon_k\}$ is bounded and Assumption A1 holds. Thus, the infinite sequence $\{x^k\}$ may be thought as being generated by Algorithm 2.1. So, (44) follows from Theorem 2.1. □

Theorem 3.2. *Assume that $\{x^k\}$ is a sequence generated by Algorithm 3.1 and, for some $k \in I\!N$, the subproblem solver does not stop. Then, every limit point of the sequence $\{x^{k,\ell}\}_{\ell\in I\!N}$ is feasible.*

Proof. By (37) the sequences $\{h(x^{k,\ell})\}$ and $\{g(x^{k,\ell})_+\}$ tend to zero as ℓ tends to infinity. This implies the desired result. □

Theorem 3.3 corresponds to Theorem 2.2 of Section 2 and establishes a sufficient and necessary computable condition for infeasibility.

Theorem 3.3. *Assume that $\{x^k\}$ is an infinite sequence generated by Algorithm 3.1. Then, the problem (1) is infeasible if and only if there exists $k \in I\!N$ such that*
$$\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] + cW_k < -c_k. \tag{45}$$

Proof. As in Theorem 3.1, defining $\varepsilon_k = cW_k$, the infinite sequence $\{x^k\}$ can be considered as being generated by Algorithm 2.1. Therefore, the thesis follows from Theorem 2.2. $\square$

In Theorem 3.4 we will prove that, when Algorithm 3.1 generates an infinite sequence $\{x^k\}$, its limit points are global solutions of (1).

Theorem 3.4. *Assume that $\{x^k\}$ is an infinite sequence generated by Algorithm 3.1 and the problem (1) is feasible. Then, $\lim_{k\to\infty} W_k = 0$ and every limit point of $\{x^k\}$ is a global solution of the problem.*

Proof. As in Theorem 2.3, Let $K \underset{\infty}{\subset} I\!N$ and $x^* \in \Omega$ be such that $\lim_{k\in K} x^k = x^*$. By Theorem 3.1, x^* is feasible.

Since the feasible set is non-empty and compact, problem (1) admits a global minimizer $z \in \Omega$. By Assumption A2, $z \in P_k$ for all $k \in I\!N$. Consider first the case in which $\rho_k \to \infty$. By Theorem 3.3, we have that

$$\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2 \le \left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2 + \frac{2(cW_k + c_k)}{\rho_k}$$

for all $k \in I\!N$, where c_k is defined in (7). Taking limits for $k \to \infty$ we get that $\|h(x^*)\| = \|g(x^*)_+\| = 0$ for all limit point x^*. This implies that

$$\lim_{k\to\infty} \|h(x^k)\| = \lim_{k\to\infty} \|g(x^k)_+\| = 0. \tag{46}$$

Now, since for all $i = 1, \dots, p$, μ_i^k/ρ_k tends to zero and $g_i(x^k)_+$ also tends to zero, we have that:

$$\lim_{k\to\infty} \min\{-g_i(x^k), \mu_i^k/\rho_k\} = 0. \tag{47}$$

By (46) and (47), it turns out that

$$\lim_{k\to\infty} W_k = 0. \tag{48}$$

As in (10), by the definition of Algorithm 3.1,

$$f(x^k)+\frac{\rho_k}{2}\left[\left\|h(x^k)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(x^k)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le f(z)+\frac{\rho_k}{2}\left[\left\|h(z)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(z)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right]+cW_k \tag{49}$$

for all $k \in I\!N$.

Since $h(z) = 0$ and $g(z) \le 0$, we have:

$$\left\|h(z) + \frac{\lambda^k}{\rho_k}\right\|^2 = \left\|\frac{\lambda^k}{\rho_k}\right\|^2 \text{ and } \left\|\left(g(z) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2 \le \left\|\frac{\mu^k}{\rho_k}\right\|^2.$$

Therefore, by (49),

$$f(x^k) \le f(x^k) + \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le f(z) + \frac{\|\lambda^k\|^2}{2\rho_k} + \frac{\|\mu^k\|^2}{2\rho_k} + cW_k.$$

By (48), taking limits for $k \in K$, using that $\lim_{k\in K} \|\lambda^k\|^2/\rho_k = \lim_{k\in K} \|\mu^k\|^2/\rho_k = 0$, by the continuity of f and the convergence of x^k, we get:

$$f(x^*) \le f(z).$$

Since z is a global minimizer, it turns out that x^* is a global minimizer in the case that $\rho_k \to \infty$.

Consider now the case in which ρ_k is bounded. By (31) and (43), we have that $\lim_{k\to\infty} W_k = 0$. Therefore, defining $\varepsilon_k = cW_k$ we may think that the infinite sequence is generated by Algorithm 2.1. Then, the thesis follows from Theorem 2.3. □

In order to complete the asymptotic convergence properties of Algorithm 3.1, we only need to consider the case in which, at some iteration k, the subproblem solver does not finish, thus generating a sequence $\{x^{k,\ell}\}$. This is done in the following theorem.

Theorem 3.5. *Assume that for some $k \in I\!N$ the subproblem solver used at Step 1 of Algorithm 3.1 does not finish (thus generating a sequence $\{x^{k,0}, x^{k,1}, x^{k,2}, \dots\}$). Then, every limit point of the infinite sequence $\{x^{k,\ell}\}$ is a global solution of (1).*

Proof. By (36) and (40), we have that $\lim_{\ell\to\infty} W_{k,\ell} = 0$. Then, by (32), if x^* is a limit point of $\{x^{k,\ell}\}$ (say, $\lim_{\ell\in K} x^{k,\ell} = x^*$) we obtain that $h(x^*) = 0$ and $g(x^*) \le 0$. Now, since $W_{k,\ell} \to 0$, we have that

$$\lim_{\ell\to\infty} \min\{-g_i(x^{k,\ell}), \mu_i^k/\rho_k\} = 0$$

for all $i = 1, \dots, p$. This implies that

$$\mu_i^k = 0 \quad \text{or} \quad g_i(x^*) = 0 \tag{50}$$

for all $i = 1, \dots, p$. The remaining steps of this proof evoke Case 2 of Theorem 2.3.

Let $z \in \Omega \cap P_k$ be a global minimizer of (1). By Step 1 of of Algorithm 3.1 and (39), we have:

$$f(x^{k,\ell})+\frac{\rho_k}{2}\left[\left\|h(x^{k,\ell})+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(x^{k,\ell})+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le f(z)+\frac{\rho_k}{2}\left[\left\|h(z)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(z)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right]+\varepsilon_{k,\ell}$$

for all $\ell \in I\!N$. Since $g(z) \le 0$ and $\mu^k/\rho_k \ge 0$,

$$\left\|\left(g(z)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2 \le \left\|\frac{\mu^k}{\rho_k}\right\|^2.$$

Thus, since $h(z) = 0$,

$$f(x^{k,\ell})+\frac{\rho_k}{2}\left[\left\|h(x^{k,\ell})+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(x^{k,\ell})+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \le f(z)+\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\frac{\mu^k}{\rho_k}\right\|^2\right]+\varepsilon_{k,\ell} \tag{51}$$

for all $\ell \in I\!N$.

By (50), if $g_i(x^*) < 0$ we have that $\mu_i^k = 0$. This implies that $(g_i(x^{k,\ell}) + \mu_i^k/\rho_k)_+ = 0$ for $\ell \in K$ large enough. Therefore, for $\ell \in K$ large enough, $\sum_{i=1}^p (g_i(x^{k,\ell}) + \mu_i^k/\rho_k)_+^2 = \sum_{g_i(x^*)=0} (g_i(x^{k,\ell}) + \mu_i^k/\rho_k)_+^2$.

Thus, by (51), for $\ell \in K$ large enough we have:

$$f(x^{k,\ell})+\frac{\rho_k}{2}\left[\sum_{i=1}^m\left(h_i(x^{k,\ell})+\frac{\lambda_i^k}{\rho_k}\right)^2+\sum_{g_i(x^*)=0}\left(g_i(x^{k,\ell})+\frac{\mu_i^k}{\rho_k}\right)_+^2\right]$$

$$\leq f(z) + \frac{\rho_k}{2}\left[\sum_{i=1}^{m}\left(\frac{\lambda_i^k}{\rho_k}\right)^2 + \sum_{g_i(x^*)=0}\left(\frac{\mu_i^k}{\rho_k}\right)^2\right] + \varepsilon_{k,\ell}. \tag{52}$$

Taking limits for $\ell \in K$ on both sides of (52) we obtain that $f(x^*) \leq f(z)$. Thus, the desired result is proved. □

As in the case of Theorem 2.4, the following theorem establishes a computable sufficient condition which guarantees that $f(x^k)$ is not much greater (and perhaps smaller) than the minimum of $f(z)$ in the feasible region.

Theorem 3.6. *Assume that $\{x^k\}$ is an infinite sequence generated by Algorithm 3.1 (thus, the subproblem solver always finishes satisfying (34)). Let $\varepsilon \in I\!R$ (perhaps negative) and $k \in I\!N$ be such that*

$$\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \leq \varepsilon.$$

Then

$$f(x^k) \leq f(z) + \varepsilon + cW_k,$$

for all global minimizer z.

Proof. As in Theorem 3.1, defining $\varepsilon_k = cW_k$, we may think the infinite sequence $\{x^k\}$ as being generated by Algorithm 2.1. Therefore, the desired results follow from Theorem 2.4. □

As in the case of Theorem 2.5, the following theorem shows that the sufficient condition stated in Theorem 3.6 eventually takes place at some x^k, when Algorithm 3.1 generates an infinite sequence.

Theorem 3.7. *Assume that $\{x^k\}$ is an infinite sequence generated by Algorithm 3.1. Suppose that the problem (1) is feasible. Let ε be an arbitrary positive number. Then, there exists $k \in I\!N$ such that*

$$\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \leq \varepsilon. \tag{53}$$

Proof. Define $\varepsilon_k = cW_k$. By Theorem 3.4, we have that $\lim_{k\to\infty} W_k = 0$. Therefore, the infinite sequence $\{x^k\}$ may be thought of as being generated by Algorithm 2.1, with $\varepsilon_k \to 0$. Therefore, the thesis follows from Theorem 2.5. □

Theorem 3.8 deals with the case in which the sequence $\{x^k\}$ is finite because, at some iteration, the stopping criterion for the subproblem never takes place. In this case, we will prove that a sufficient condition similar to (53) is eventually fulfilled.

Theorem 3.8. *Assume that for some $k \in I\!N$ the subproblem solver used in Step 1 of Algorithm 3.1 does not finish. Let $\{x^{k,0}, x^{k,1}, x^{k,2}, \ldots\}$ be the sequence generated by the subproblem solver. Let $\varepsilon > 0$ be arbitrarily small. Then, there exists $\ell \in I\!N$ such that*

$$\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^{k,\ell}) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^{k,\ell}) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \leq \varepsilon. \tag{54}$$

Proof. By the compactness of Ω, there exists $K \underset{\infty}{\subset} I\!N$ and $x^* \in \Omega$ such that $\lim_{\ell \in K} x^{k,\ell} = x^*$ and, by Theorem 3.2, x^* is feasible. The proof follows as a small variation of the arguments of Theorem 2.5 for the case in which $\{\rho_k\}$ is bounded plus the fact that $W_{k,\ell} \to 0$. □

Let us prove now that, again in the case in which the algorithm stays solving the subproblem k, the condition (54) guarantees a small value of $f(x^{k,\ell})$.

Theorem 3.9. *Assume that for some $k \in I\!N$ the subproblem solver used at Step 1 of Algorithm 3.1 does not finish. As in previous theorems, let $\{x^{k,0}, x^{k,1}, x^{k,2}, \ldots\}$ be the sequence generated by the subproblem solver. Let $\varepsilon \in I\!R$ (note that ε may be negative) and $\ell \in I\!N$ such that*

$$\frac{\rho_k}{2}\left[\left\|\frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\frac{\mu^k}{\rho_k}\right\|^2\right] - \frac{\rho_k}{2}\left[\left\|h(x^{k,\ell}) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^{k,\ell}) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \leq \varepsilon. \quad (55)$$

Then, for all global minimizer z of problem (1), we have that

$$f(x^{k,\ell}) \leq f(z) + \varepsilon + \varepsilon_{k,\ell}. \quad (56)$$

Proof. We have that, for all global minimizer z and for all $\ell \in I\!N$,

$$f(x^{k,\ell}) + \frac{\rho_k}{2}\left[\left\|h(x^{k,\ell}) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^{k,\ell}) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \leq f(z) + \frac{\rho_k}{2}\left[\left\|h(z) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(z) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] + \varepsilon_{k,\ell}. \quad (57)$$

The proof proceeds similarly to Theorem 2.4, replacing x^k by $x^{k,\ell}$ and ϵ_k by $\epsilon_{k,\ell}$. □

Theorem 3.10 is the final result of this section. As in the case of Theorem 2.6, we prove that Algorithm 3.1 stops in finite time with a certificate of infeasibility, or guaranteeing optimality up to arbitrarily small given precisions in terms of feasibility and optimality.

Theorem 3.10. *The execution of Algorithm 3.1, for solving problem (1), finishes in finite time with one of the following diagnostics:*

1. **Infeasibility**, *which means that, with guarantee, no feasible point of (1) exists;*
2. **Solution found**, *in the case that the final point x^k is guaranteed to satisfy*

   $$\|h(x^k)\| + \|g(x^k)_+\| \leq \varepsilon_{feas}$$

   and

   $$f(x^k) \leq f(z) + \varepsilon_{opt}$$

   for all $z \in \Omega$ such that $h(z) = 0$ and $g(z) \leq 0$.

Proof. The proof follows straightforwardly from the theorems proved in this section. □

4 Numerical experiments

We implemented Algorithms 2.1 and 3.1 as modifications of the method introduced in [21] (pp.141–142), freely available at *http://www.ime.usp.br/∼egbirgin/*. For solving the subproblems at Step 1 of

Algorithms 2.1 and 3.1, we considered a new implementation of the well-known spatial Branch-and-Bound [51, 71] αBB method [1, 2, 3, 14] described in [21] (pp.147–148). This new implementation is motivated by the necessity of having available, at each iteration ℓ of the subproblems solver, an approximate solution $x^{k,\ell}$ and a tolerance $\varepsilon_{k,\ell}$ such that (39) holds. Moreover, the new implementation incorporates mechanisms for possible detecting the infeasibility of the subproblem being solved. This new implementation of the αBB method is fully described in [27], where numerical experiments comparing it with the one considered in [21] are reported. From now on, the method introduced in [21] (pp.141–142) will be named "Original ALABB", while the implementations of Algorithms 2.1 and 3.1 will be named ALABB_{GP} and ALABB_{AP}, respectively. ALABB stands for "Augmented Lagrangian method that uses the αBB method for solving the subproblems". The subscripts, GP and AP stand for "Given Precision (for solving the subproblems)" and "Adaptive Precision (for solving the subproblems)", respectively.

For interval analysis calculations we use the Intlib library [48]. For solving linear programming problems we use subroutine `simplx` from the Numerical Recipes in Fortran [60]. To solve the linearly constrained optimization problems, we use Genlin [13], an active-set method for linearly constrained optimization based on a relaxed form of Spectral Projected Gradient iterations intercalated with internal iterations restricted to faces of the polytope. Genlin generalizes the box-constraint optimization method Gencan [23]. It should be noted that `simplx` and Genlin are dense solvers. Therefore, for problems with more that 50 variables or constraints, we used Minos [55] to solve linear programming problems and linearly constrained problems. Codes are written in Fortran 77 (double precision). All the experiments were run on a 3.2 GHz Intel(R) Pentium(R) with 4 processors, 1Gb of RAM and Linux Operating System.

Given a problem of the form (1), we consider that $\Omega = \Omega_1 \cap \Omega_2$, where $\Omega_1 = \{x \in I\!R^n \mid l \leq x \leq u\}$, $\Omega_2 = \{x \in I\!R^n \mid Ax = b,\ Cx \leq d\}$, and $l \leq x \leq u$, $Ax = b$, and $Cx \leq d$ represent all the bound constraints, linear equality constraints, and linear inequality constraints of problem (1), respectively. This means that only the nonlinear constraints will be penalized. In both algorithms, as suggested in [5] for the underlying local augmented Lagrangian method for Nonlinear Programming problems, we set $\gamma = 10$, $\tau = 0.5$, $\lambda_{\min} = -10^{20}$, $\mu_{\max} = \lambda_{\max} = 10^{20}$, $\lambda^1 = 0$, $\mu^1 = 0$, and

$$\rho_1 = \max\left\{10^{-6}, \min\left\{10, \frac{2|f(x^0)|}{\|h(x^0)\|^2 + \|g(x^0)_+\|^2}\right\}\right\},$$

where x^0 is an arbitrary initial point. In Algorithm ALABB_{AP}, we arbitrarily set $c = 1$.

4.1 Preliminaries

We start the numerical experiments by checking the practical behavior of Algorithm ALABB_{GP} in very simple problems. The constant c_k at Step 2 of Algorithm ALABB_{GP} is computed as follows. By interval arithmetic, it is computed (only once) the interval $[f^{\min}, f^{\max}]$ such that $f^{\min} \leq f(x) \leq f^{\max}$ for all $x \in \Omega_1$. Then c_k is given by

$$c_k = \max\{f(x^k) - f^{\min}, f^{\max} - f(x^k)\}. \tag{58}$$

We considered $\varepsilon_{\text{feas}} = \varepsilon_{\text{opt}} = 10^{-4}$ and $\bar{\varepsilon}_k = \max\{10^{-k}, \varepsilon_{\text{opt}}/2\}$.

In a first set of experiments, we considered the three simple problems given by:

Problem A: Min x subject to $x^2 + 1 \leq 0,\ -10 \leq x \leq 10$,

Problem B: Min x subject to $x^2 = 0,\ -10 \leq x \leq 10$,

Problem C: Min x subject to $x^2 \leq 1,\ -10 \leq x \leq 10$.

Problem A is infeasible, while Problems B and C are feasible problems, Problem C admits Lagrange multipliers and Problem B does not. In all cases we arbitrarily considered $x^0 = 1.5$. Table 1 shows the behavior of Algorithm ALABB$_{\rm GP}$ in Problems A, B, and C in detail. In the table, k represents the iteration of the algorithm, ρ_k and λ^k are the values of the penalty parameter and the Lagrange multiplier, respectively, that define the k-th augmented Lagrangian subproblem, $x^k \in I\!R$ is the ε_k-global minimizer of the k-th subproblem, $f(x^k)$ is the value of the objective function of the original problem at x^k, $\|h(x^k)\| + \|g(x^k)_+\|$ is the (Euclidean norm) infeasibility measurement at x^k, c_k is the value of the constant computed at Step 2 to perform the infeasibility test while γ_k is the quantity defined at Step 2. Finally, $\varepsilon_k \leq \bar{\varepsilon}_k$ is the actual tolerance returned by the inner solver αBB such that x^k is an ε_k-global minimizer of the augmented Lagrangian subproblem of iteration k.

Problem A							
k	ρ_k	λ^k	x^k and $f(x^k)$	$\|h(x^*)\| + \|g(x^*)_+\|$	c^k	γ_k	ε_k
0			1.5000E+00	3.3E+00			
1	2.8E+00	0.0E+00	-1.7104E−01	1.0E+00	1.0E+01	-1.5E+00	0.0E+00
2	2.8E+00	2.9E+00	-8.6434E−02	1.0E+00	1.0E+01	-4.4E+00	0.0E+00
3	2.8E+01	5.8E+00	-1.4623E−02	1.0E+00	1.0E+01	-2.0E+01	0.0E+00

Problem B							
k	ρ_k	λ^k	x^k and $f(x^k)$	$\|h(x^*)\| + \|g(x^*)_+\|$	c^k	γ_k	ε_k
0			1.5000E+00	2.3E+00			
1	5.9E+00	0.0E+00	-4.3861E−01	1.9E−01	1.0E+01	-1.1E−01	1.7E−10
2	5.9E+00	1.1E+00	-2.9927E−01	9.0E−02	1.0E+01	-1.3E−01	6.3E−11
3	5.9E+00	1.7E+00	-2.4628E−01	6.1E−02	1.0E+01	-1.1E−01	0.0E+00
4	5.9E+01	2.0E+00	-1.4925E−01	2.2E−02	1.0E+01	-6.0E−02	0.0E+00
5	5.9E+01	3.4E+00	-1.1925E−01	1.4E−02	1.0E+01	-5.4E−02	0.0E+00
6	5.9E+02	4.2E+00	-7.0250E−02	4.9E−03	1.0E+01	-2.8E−02	0.0E+00
7	5.9E+02	7.1E+00	-5.5791E−02	3.1E−03	1.0E+01	-2.5E−02	0.0E+00
8	5.9E+03	9.0E+00	-3.2691E−02	1.1E−03	1.0E+01	-1.3E−02	8.3E−16
9	5.9E+03	1.5E+01	-2.5933E−02	6.7E−04	1.0E+01	-1.2E−02	6.9E−18
10	5.9E+04	1.9E+01	-1.5181E−02	2.3E−04	1.0E+01	-6.0E−03	4.6E−16
11	5.9E+04	3.3E+01	-1.2040E−02	1.4E−04	1.0E+01	-5.4E−03	2.3E−17
12	5.9E+05	4.2E+01	-7.0468E−03	5.0E−05	1.0E+01	-2.8E−03	2.2E−16

Problem C							
k	ρ_k	λ^k	x^k and $f(x^k)$	$\|h(x^*)\| + \|g(x^*)_+\|$	c^k	γ_k	ε_k
0			1.5000E+00	1.3E+00			
1	1.0E+01	0.0E+00	-1.0241E+00	4.9E−02	1.1E+01	-1.2E−02	3.4E−12
2	1.0E+01	4.9E−01	-1.0006E+00	1.1E−03	1.1E+01	-5.7E−04	0.0E+00
3	1.0E+01	5.0E−01	-1.0000E+00	2.7E−05	1.1E+01	-1.4E−05	0.0E+00

Table 1: Detailed report of the quantities that characterize the behavior of Algorithm ALABB$_{\rm GP}$ on Problems A, B, and C.

The highlight of Table 1 is that Algorithm ALABB$_{\rm GP}$ detects very quickly that Problem A is infeasible and makes the method stop. Therefore, the contribution of Algorithm ALABB$_{\rm GP}$ with respect to the Original ALABB method is that, for Problem A, it rapidly detects that the problem is infeasible and stops with a *certificate of infeasibility*. In contrast, Original ALABB method applied to Problem A stops after nineteen iterations heuristically declaring "Constraints violation has not decreased substantially over 9 outer iterations. Problem possibly infeasible.". In Problems B and C Algorithm ALABB$_{\rm GP}$ and the Original ALABB method perform similarly in practice. Both methods exhibit, in Problem B, the typical behavior in the case in which Lagrange multipliers do not exist, taking many more iterations to solve it than in the case of Problem C. The difference between the methods lies in the fact that Algorithm ALABB$_{\rm GP}$ *provides the gap* (sum of γ_k plus ε_k displayed in the last two columns of Table 1) and *guarantees with finite termination* that the required gap $\varepsilon_{\rm opt} = 10^{-4}$ is achieved.

Figures in the table (last iteration of each one of the three problems) show that negative gaps are reported by Algorithm ALABB$_{\rm GP}$ for all three problems. This is only possible because the delivered approximations to global solutions have an infeasibility tolerance of $\varepsilon_{\rm feas} = 10^{-4}$. In fact, direct calculation shows that, for all k, we have

$$\gamma_k + \varepsilon_k \geq \epsilon_k - \frac{\rho_k}{2}\left[\|h(x^k)\|^2 + \|g(x^k)_+\|^2\right] - \max\{\|\lambda^k\|, \|\mu^k\|\}\left[\|h(x^k)\| + \|g(x^k)_+\|\right].$$

Therefore, at the final iterate, we have

$$\gamma_k + \varepsilon_k \;\geq\; \varepsilon_k - \frac{\rho_k}{2}\varepsilon_{\rm feas}^2 - \max\{\|\lambda^k\|, \|\mu^k\|\}\varepsilon_{\rm feas},$$

showing what is the best that can be expected for the global optimality gap $\gamma_k + \varepsilon_k$ with respect to the allowed infeasibility tolerance $\varepsilon_{\rm feas}$.

The reader may be surprised by the small values of ε_k reported in the last column of Table 1 that say that subproblems are being solved to global optimality with high accuracy. Consider Problem B. The first subproblem is solved at Step 1 requiring precision $\bar{\varepsilon}_1 = 0.1$, using $\lambda^0 = 0$ and $\rho_1 \approx 5.9259$. The inner solver αBB returns $x^1 \approx -4.3861 \times 10^{-1}$ guaranteeing that it satisfies (4) with $\varepsilon_1 \approx 1.6771 \times 10^{-10} \ll \bar{\varepsilon}_1$. It may be useful to mention that the subproblem solved by the αBB method was

$$\text{Min } x + \frac{\rho_1}{2}x^4 \text{ subject to } -10 \leq x \leq 10,$$

whose solution is, in fact, approximately x^1. The convexity of this subproblem explains the tight gap $\varepsilon_1 \ll \bar{\varepsilon}_k$ obtained by the αBB method.

To verify the influence of the choice of c_k at Step 2 of Algorithm ALABB$_{\rm GP}$, we checked the behavior of the method on Problem A when considering the naive choice $c_k = f^{\max} - f^{\min} = 20$ for all k. As expected, larger values of c_k (larger than the choice suggested in (58)), make the infeasibility test at Step 2 harder to be satisfied and, in this simple example, the method takes one more iteration to stop giving a certificate of infeasibility. As an alternative to strengthen the method, values for $f^{\min}$ and $f^{\max}$ may be found by computing a global solution to the two auxiliary problems

$$\text{Min / Max } f(x) \text{ subject to } x \in \Omega,$$

which, by the definition of Ω at the beginning of the present section, are linearly constrained problems. This task is not harder than twice the effort of Step 2 of Algorithm ALABB$_{\rm GP}$. This alternative way of computing c_k returns the same answer as the one given by interval analysis on Problem A, but might provide better bounds in harder problems.

4.2 Influence of the endogenous sequence $\{\varepsilon_k\}$

In the following experiments we considered the set of problems analysed in [21] whose precisely considered formulations can be found in [22]. The problems' AMPL formulations can also be found at *http://www.ime.usp.br/∼egbirgin/*.

With the purpose of evaluating the influence of the endogenous sequence $\{\varepsilon_k\}$ considered by Algorithm ALABB$_{\rm AP}$ to stop the subproblems' solver, we run Algorithms ALABB$_{\rm GP}$ and ALABB$_{\rm AP}$ with the same tolerances considered in [21]. It means that we considered $\varepsilon_{\rm feas} = \varepsilon_{\rm opt} = 10^{-4}$ for Problems 1–16 and $\varepsilon_{\rm feas} = \varepsilon_{\rm opt} = 10^{-1}$ for the larger problems prodpl0 and prodpl1. Table 2 shows the results. In the table, the first three columns identify the problem and the number of variables and constraints. "Time" is the CPU time in seconds, "It" is the number of augmented Lagrangian iterations, "#Nodes" is the total number of Branch-and-Bound nodes used by the inner solver αBB to solve all the subproblems of a given problem. #Nodes gives a measurement of the effort needed to solve the whole set of subproblems of a given problem, i.e. the overall effort needed to solve the original problem. Since #Nodes is a much more precise measurement than the very short CPU times, it will be used, from now on, to evaluate the performance of the methods. Still in the table, $f(x^*)$ is the value of the objective function at the final iterate x^*, $\|h(x^*)\| + \|g(x^*)_+\|$ is the (Euclidean norm) infeasibility measurement at x^*, and $\varepsilon \le \varepsilon_{\rm opt}$ is the reported gap for the ε-global optimality of x^*. The main contribution of Algorithm ALABB$_{\rm GP}$ with respect to the Original ALABB method is to provide the gap $\varepsilon \le \varepsilon_{\rm opt}$ such that $f(x^*) \le f(z) + \varepsilon$ for any feasible point z. Note that, since an $\varepsilon_{\rm feas}$ level of infeasibility is being accepted, the method is capable of providing negative values of ε for some instances. As expected, required optimality gaps were guaranteed in all the cases using a finite number of iterations.

The main *practical* differences between Algorithms ALABB$_{\rm GP}$ and ALABB$_{\rm AP}$ with respect to the Original ALABB method (when applied to feasible problems) is that Algorithms ALABB$_{\rm GP}$ and ALABB$_{\rm AP}$ provide the actual gap $\varepsilon \le \varepsilon_{\rm opt}$ such that $f(x^*) \le f(z) + \varepsilon$ for all feasible point z. Comparing the number of Branch-and-Bound nodes in Table 2 it is easy to see that the number of nodes generated by Algorithm ALABB$_{\rm AP}$ is, in *all* the considered problems, not greater than the number of nodes generated by Algorithm ALABB$_{\rm GP}$, while, on average, is almost 20% smaller.

We end this section solving Problems 1–16, prodpl0, and prodpl1 with tolerances $\varepsilon_{\rm feas} = \varepsilon_{\rm opt} = 10^{-8}$ to show that stricter tolerances can also be achieved by the current implementation of Algorithms ALABB$_{\rm GP}$ and ALABB$_{\rm AP}$. Table 3 shows the results. Figures in the table show that: (a) as expected, both algorithms achieved the desired feasibility and optimality tolerances in a finite number of iterations; (b) optimality gaps smaller than or equal to the required optimality gap are delivered by the methods.

4.3 Infeasible problems

In this subsection we consider the problem of packing a given set of N circles with radii $r_i, i = 1\dots,N$, within an ellipse with semi-axes $e_a \ge e_b > 0$, maximizing the sum of the squared distances between the circles' centers. By packing, we mean that the circles must be placed within the ellipse without overlapping. Considering continuous variables $u, v, s \in I\!R^N$, this problem can be modeled [17, 28] as a continuous and differentiable nonlinear programming problem as follows:

$$\begin{array}{lll}
\text{Maximize} & \sum_{i<j} \left\{ \left[(1+(s_i-1)(e_b^2/e_a^2))u_i - (1+(s_j-1)(e_b^2/e_a^2))u_j\right]^2 + [s_i v_i - s_j v_j]^2 \right\} & \\
\text{subject to} & (u_i/e_a)^2 + (v_i/e_b)^2 = 1, & i = 1,\dots,N, \\
& (s_i-1)^2 \left[(e_b^2/e_a^2)^2 u_i^2 + v_i^2\right] \ge r_i^2, & i = 1,\dots,N, \\
& \left[(1+(s_i-1)(e_b^2/e_a^2))u_i - (1+(s_j-1)(e_b^2/e_a^2))u_j\right]^2 + [s_i v_i - s_j v_j]^2 \ge (r_i+r_j)^2, & \forall\, i<j, \\
& 0 \le s_i \le 1, & i = 1,\dots,N.
\end{array}$$

The Cartesian coordinates of the circles' centers can be recovered using

$$x_i = \left[1 + (s_i - 1)(e_b^2/e_a^2)\right] u_i, \quad y_i = s_i v_i, \quad i = 1,\dots,N.$$

In order to apply a spatial Branch-and-Bound-based global optimization technique, redundant valid bounds $-e_a \le u_i \le e_a$ and $-e_b \le v_i \le e_b$, for $i = 1,\dots,N$, may be added. We considered a set of sixteen instances with $(e_a, e_b) \in \{(4,2),(3,2),(2,2),(2,1)\}$ and $N \in \{2,3,4,5\}$. In all cases, we arbitrarily considered identical unitary-radius circles.

In order to tackle a problem with the methods being introduced, some information is mandatory while some other that may be useful to improve the efficiency of the method is not. Mandatory

Algorithm ALABB_{GP}

Problem	n	q	Time	It	#Nodes	$f(x^*)$	$\|h(x^*)\| + \|g(x^*)_+\|$	ε
1	5	3	3.27	10	29134	2.9312786586066682E−02	5.5E−06	5.2E−05
2(a)	11	8(6)	0.02	8	17	-4.0000000024768008E+02	1.8E−06	-1.1E−07
2(b)	11	8(6)	0.08	13	66	-6.0000005928794553E+02	1.6E−05	-5.9E−05
2(c)	11	8(6)	0.04	8	17	-7.5000000001451747E+02	4.9E−08	-9.5E−09
2(d)	12	9(7)	0.00	2	2	-4.0000000000184355E+02	2.6E−09	-5.5E−09
3(a)	6	5	2.48	6	6250	-3.8880635743184450E−01	5.6E−06	5.5E−05
3(b)	2	1	0.45	4	3804	-3.8881143432360377E−01	4.4E−10	9.5E−05
4	2	1	0.00	4	36	-6.6666666666666670E+00	1.8E−15	5.4E−05
5	3	3	0.00	5	115	2.0115933406086481E+02	7.0E−05	1.0E−10
6	2	1	0.01	5	97	3.7629193233270610E+02	0.0E+00	5.0E−05
7	2	4(2)	0.00	4	190	-2.8284271288287419E+00	1.2E−08	1.1E−05
8	2	2(1)	0.08	5	2126	-1.1870486335082188E+02	1.0E−06	4.2E−05
9	6	6(6)	0.00	1	2	-1.3401903555050817E+01	0.0E+00	9.6E−06
10	2	2	0.00	4	82	7.4178195828323901E−01	5.2E−10	9.6E−05
11	2	1	0.00	4	46	-4.9999999999981415E−01	0.0E+00	1.0E−07
12	2	1	0.00	8	144	-1.6738975393040647E+01	2.0E−05	-3.6E−05
13	3	2(1)	0.00	8	118	1.8934657289312642E+02	9.5E−10	-1.0E−11
14	4	3(3)	0.00	1	1	-4.5142016513619279E+00	0.0E+00	0.0E+00
15	3	3(1)	0.01	4	89	0.0000000000000000E+00	1.6E−06	3.1E−05
16	5	3(1)	0.01	6	94	7.0492011812333732E−01	3.5E−05	3.4E−05
prodpl0	68	37(33)	1.41	2	7	5.9183361696283363E+01	9.2E−02	-9.8E−01
prodpl1	68	37(33)	1.73	2	11	5.2852602634604963E+01	2.7E−02	-1.4E−01

Algorithm ALABB_{AP}

Problem	n	q	Time	It	#Nodes	$f(x^*)$	$\|h(x^*)\| + \|g(x^*)_+\|$	ε
1	5	3	2.51	10	23025	2.9312786586066682E−02	5.5E−06	1.0E−04
2(a)	11	8(6)	0.02	8	17	-4.0000000024768008E+02	1.8E−06	-1.1E−07
2(b)	11	8(6)	0.08	13	66	-6.0000005928794553E+02	1.6E−05	-5.9E−05
2(c)	11	8(6)	0.04	8	17	-7.5000000001451747E+02	4.9E−08	-9.5E−09
2(d)	12	9(7)	0.00	2	2	-4.0000000000184355E+02	2.6E−09	-5.5E−09
3(a)	6	5	1.62	6	4525	-3.8880635674415054E−01	5.6E−06	1.0E−04
3(b)	2	1	0.26	2	2149	-3.8881366113419052E−01	2.9E−05	9.9E−05
4	2	1	0.00	2	18	-6.6666666666666670E+00	1.8E−15	5.4E−05
5	3	3	0.00	5	112	2.0115933406086481E+02	7.0E−05	1.0E−10
6	2	1	0.01	5	97	3.7629193233270610E+02	0.0E+00	9.4E−05
7	2	4(2)	0.00	3	142	-2.8284277850935133E+00	1.9E−06	1.0E−05
8	2	2(1)	0.08	5	2067	-1.1870486335101779E+02	1.0E−06	9.9E−05
9	6	6(6)	0.00	1	2	-1.3401903555050817E+01	0.0E+00	9.6E−06
10	2	2	0.00	2	36	7.4178849964562033E−01	0.0E+00	7.1E−05
11	2	1	0.00	2	20	-4.9999804943347009E−01	0.0E+00	2.1E−06
12	2	1	0.00	8	125	-1.6738975393040647E+01	2.0E−05	-2.3E−05
13	3	2(1)	0.02	8	116	1.8934657289312642E+02	9.5E−10	-1.0E−11
14	4	3(3)	0.00	1	1	-4.5142016513619279E+00	0.0E+00	0.0E+00
15	3	3(1)	0.00	1	28	0.0000000000000000E+00	1.5E−06	3.1E−05
16	5	3(1)	0.01	6	70	7.0492010918641423E−01	3.6E−05	6.6E−05
prodpl0	68	37(33)	1.41	2	7	5.9183361696283363E+01	9.2E−02	-9.8E−01
prodpl1	68	37(33)	1.75	2	11	5.2852602634604963E+01	2.7E−02	-1.4E−01

Table 2: Performance of Algorithms ALABB_{GP} and ALABB_{AP} in the set of problems considered in [21] and with the same tolerances considered in [21].

Algorithm ALABB$_{\text{GP}}$								
Problem	n	q	Time	It	#Nodes	$f(x^*)$	$\Vert h(x^*)\Vert + \Vert g(x^*)_+\Vert$	ε
1	5	3	15.74	12	117762	2.9310830860950987E−02	4.3E−10	5.1E−09
2(a)	11	8(6)	0.16	11	32	-4.0000000002195316E+02	5.5E−09	-2.2E−08
2(b)	11	8(6)	0.21	19	114	-6.0000000000000455E+02	8.8E−13	-3.5E−12
2(c)	11	8(6)	0.04	8	17	-7.5000000012666578E+02	9.4E−09	-9.8E−09
2(d)	12	9(7)	0.00	2	2	-4.0000000000184355E+02	2.6E−09	-5.5E−09
3(a)	6	5	14.73	9	29850	-3.8881143431953308E−01	6.1E−11	4.9E−09
3(b)	2	1	1.55	8	10091	-3.8881143430404086E−01	1.6E−10	9.7E−09
4	2	1	0.00	5	48	-6.6666666666666670E+00	1.8E−15	5.9E−16
5	3	3	0.00	8	226	2.0115933406086481E+02	2.5E−10	-1.6E−14
6	2	1	0.03	8	398	3.7629193233029866E+02	0.0E+00	8.3E−09
7	2	4(2)	0.01	5	247	-2.8284271247520278E+00	1.7E−11	-5.8E−12
8	2	2(1)	0.20	8	4472	-1.1870485977521858E+02	6.5E−11	9.8E−09
9	6	6(6)	0.00	1	3	-1.3401903555050817E+01	0.0E+00	3.4E−13
10	2	2	0.00	7	172	7.4178195825585458E−01	0.0E+00	8.8E−12
11	2	1	0.00	7	91	-4.9999999999515704E−01	0.0E+00	4.8E−12
12	2	1	0.01	12	258	-1.6738893206126292E+01	5.4E−09	-2.2E−08
13	3	2(1)	0.12	8	120	1.8934657289313745E+02	1.8E−11	7.8E−13
14	4	3(3)	0.00	1	1	-4.5142016513619279E+00	0.0E+00	0.0E+00
15	3	3(1)	0.30	8	315	0.0000000000000000E+00	2.8E−11	1.9E−09
16	5	3(1)	0.07	10	290	7.0492492643582416E−01	5.8E−09	1.8E−09
prodpl0	68	37(33)	16.41	13	94	6.0919236339470416E+01	5.8E−09	-1.4E−07
prodpl1	68	37(33)	7.96	7	54	5.3037015013663307E+01	7.7E−09	-3.3E−08

Algorithm ALABB$_{\text{AP}}$								
Problem	n	q	Time	It	#Nodes	$f(x^*)$	$\Vert h(x^*)\Vert + \Vert g(x^*)_+\Vert$	ε
1	5	3	11.77	12	91191	2.9310830860950987E−02	4.3E−10	1.0E−08
2(a)	11	8(6)	0.16	11	32	-4.0000000002195316E+02	5.5E−09	-2.2E−08
2(b)	11	8(6)	0.21	19	114	-6.0000000000000455E+02	8.8E−13	-3.5E−12
2(c)	11	8(6)	0.04	8	17	-7.5000000012666578E+02	9.4E−09	-9.8E−09
2(d)	12	9(7)	0.00	2	2	-4.0000000000184355E+02	2.6E−09	-5.5E−09
3(a)	6	5	9.69	8	20222	-3.8881143348144187E−01	8.9E−10	1.0E−08
3(b)	2	1	0.83	4	5467	-3.8881143431270038E−01	2.8E−10	9.7E−09
4	2	1	0.00	2	19	-6.6666666666666670E+00	1.8E−15	5.9E−16
5	3	3	0.01	8	223	2.0115933406086481E+02	2.5E−10	-1.6E−14
6	2	1	0.02	6	234	3.7629193229876364E+02	4.4E−10	6.0E−09
7	2	4(2)	0.01	5	257	-2.8284271247520278E+00	1.7E−11	-5.8E−12
8	2	2(1)	0.18	7	3815	-1.1870485977654887E+02	4.5E−10	9.8E−09
9	6	6(6)	0.00	1	3	-1.3401903555050817E+01	0.0E+00	3.4E−13
10	2	2	0.00	4	100	7.4178195825094195E−01	1.1E−13	3.9E−12
11	2	1	0.00	3	35	-5.0000000060447813E−01	2.4E−09	-6.0E−10
12	2	1	0.01	13	262	-1.6738893192901013E+01	2.1E−09	-8.5E−09
13	3	2(1)	0.12	8	118	1.8934657289313745E+02	1.8E−11	7.8E−13
14	4	3(3)	0.00	1	1	-4.5142016513619279E+00	0.0E+00	0.0E+00
15	3	3(1)	0.25	6	310	0.0000000000000000E+00	1.4E−09	1.9E−09
16	5	3(1)	0.05	10	230	7.0492492643582416E−01	5.8E−09	8.5E−09
prodpl0	68	37(33)	15.94	13	94	6.0919236353090284E+01	5.4E−09	-1.2E−07
prodpl1	68	37(33)	7.80	7	54	5.3037015019092010E+01	7.3E−09	-2.8E−08

Table 3: Performance of Algorithms ALABB$_{\text{GP}}$ and ALABB$_{\text{AP}}$ with $\varepsilon_{\text{feas}} = \varepsilon_{\text{opt}} = 10^{-8}$.

information includes and is limited to: (a) Fortran subroutines to compute the objective function, the constraints, and their first and second derivatives at a given point; and (b) Fortran subroutines to compute for a given box or subbox (using, for example, interval analysis), lower and upper bounds for all quantities listed in item (a) with the exception of the gradient of the objective function. The user must also indicate whenever a variable only appears linearly in the objective function and in linear constraints (those variables do not need to be spatially branched by the method). The optional information includes: (a) the best known value of the objective function at a feasible point; and (b) for a given subbox, a subroutine capable of computing linear underestimators, valid within the subbox, for the objective function, the inequality constraints, and/or the inequality constraints coming from interpreting each equality constraint $h_j(x) = 0$ as a double inequality constraint of the form $h_j(x) \leq 0$ and $-h_j(x) \leq 0$. It is important to notice that none of the optional information is being provided for the problem being analysed in the present subsection, basically, because coding those additional data is an extremely tedious task. Providing it automatically would be a great advantage of an improved implementation of the methods. Requirements regarding derivatives and interval arithmetic computations might also be automatically provided by the methods if we were using, for example, a different programming language with access to resources such as operators overloading and/or automatic differentiation tools.

Table 4 shows the performance of Algorithms ALABB$_{\mathrm{GP}}$ and ALABB$_{\mathrm{AP}}$ on the sixteen instances of the packing problem, while Figure 1 illustrate the "solutions". In the table, e_a, e_b, and N represent the elipses' axes and the number of considered identical unitary-radius circles; n and q represent the number of variables and the number of constraints, respectively. Note that $n = 3N$, $q = 2N + N(N-1)/2$, and that all constraints (as well as the objective function) are nonlinear. The remaining columns show the algorithms' performance and were already described before, the exception being the last column, that identifies whether the problem was detected to be infeasible or not.

Problem				Algorithm ALABB$_{\mathrm{GP}}$						
(e_a, e_b)	N	n	q	Time	It	#Nodes	$f(x^*)$	$\|h(x^*)\| + \|g(x^*)_+\|$	ε	SC
(4,2)	2	6	5	23.57	8	195260	3.6000632560807304E+01	1.9E−05	-5.8E−04	Solution found
(4,2)	3	9	9	42.01	11	25169	5.8521317671731637E+01	5.7E−05	1.4E−07	Solution found
(4,2)	4	12	14	661.38	13	218407	9.0072124538555983E+01	1.2E−05	-2.1E−06	Solution found
(4,2)	5	15	20	4923.75	13	1682495	1.1780270817412281E+02	1.4E−05	1.3E−08	Solution found
(3,2)	2	6	5	43.67	5	351735	1.6000808473480859E+01	6.3E−05	-7.6E−04	Solution found
(3,2)	3	9	9	49.91	11	40061	2.6239622360900952E+01	3.4E−05	-3.4E−04	Solution found
(3,2)	4	12	14	213.62	11	67749	4.0410065156787176E+01	4.1E−05	1.3E−07	Solution found
(3,2)	5	15	20	175.09	6	56634	6.9186201732988039E+01	5.9E−01	-1.1E+03	Infeasible
(2,2)	2	6	5	274.45	5	2368047	4.0000000632358033E+00	1.8E−07	5.0E−05	Solution found
(2,2)	3	9	9	127.25	6	242570	1.1935922304117483E+01	3.5E−01	-3.9E+02	Infeasible
(2,2)	4	12	14	1278.15	6	1782970	3.1185540474086196E+01	1.1E+00	-4.2E+02	Infeasible
(2,2)	5	15	20	11388.47	4	9962300	6.7360974622083830E+01	1.9E+00	-8.2E+02	Infeasible
(2,1)	2	6	5	0.00	1	1	8.8910020801615914E+00	1.9E+00	-9.3E+00	Infeasible
(2,1)	3	9	9	0.00	1	1	2.3633719665128936E+01	4.7E+00	-8.4E+01	Infeasible
(2,1)	4	12	14	0.00	1	1	2.7252194724984069E+01	7.3E+00	-2.2E+02	Infeasible
(2,1)	5	15	20	0.00	1	1	3.8314608806172572E+01	9.7E+00	-3.7E+02	Infeasible

Problem				Algorithm ALABB$_{\mathrm{AP}}$						
(e_a, e_b)	N	n	q	Time	It	#Nodes	$f(x^*)$	$\|h(x^*)\| + \|g(x^*)_+\|$	ε	SC
(4,2)	2	6	5	15.50	8	122228	3.6000632560807304E+01	1.9E−05	1.0E−04	Solution found
(4,2)	3	9	9	42.05	11	25169	5.8521317671731637E+01	5.7E−05	1.4E−07	Solution found
(4,2)	4	12	14	655.75	13	218391	9.0072124538555983E+01	1.2E−05	-2.1E−06	Solution found
(4,2)	5	15	20	4902.68	13	1682479	1.1780270817412281E+02	1.4E−05	1.3E−08	Solution found
(3,2)	2	6	5	39.00	5	313683	1.6000808541953081E+01	6.3E−05	9.6E−05	Solution found
(3,2)	3	9	9	49.89	11	40001	2.6239622360900952E+01	3.4E−05	-3.4E−04	Solution found
(3,2)	4	12	14	212.47	11	67749	4.0410065156787176E+01	4.1E−05	1.3E−07	Solution found
(3,2)	5	15	20	176.82	6	56634	6.9186201732988039E+01	5.9E−01	-1.1E+03	Infeasible
(2,2)	2	6	5	151.24	4	1263448	3.9999950333831871E+00	9.1E−06	1.0E−04	Solution found
(2,2)	3	9	9	126.62	6	242542	1.1935922304117483E+01	3.5E−01	-3.9E+02	Infeasible
(2,2)	4	12	14	1274.16	6	1782626	3.1185540474086196E+01	1.1E+00	-4.2E+02	Infeasible
(2,2)	5	15	20	11124.96	4	9962300	6.7360974622083830E+01	1.9E+00	-8.2E+02	Infeasible
(2,1)	2	6	5	0.00	1	1	8.8910020801615914E+00	1.9E+00	1.0E+20	Infeasible
(2,1)	3	9	9	0.00	1	1	2.3633719665128936E+01	4.7E+00	1.0E+20	Infeasible
(2,1)	4	12	14	0.00	1	1	2.7252194724984069E+01	7.3E+00	1.0E+20	Infeasible
(2,1)	5	15	20	0.00	1	1	3.8314608806172572E+01	9.7E+00	1.0E+20	Infeasible

Table 4: Performance of Algorithms ALABB$_{\mathrm{GP}}$ and ALABB$_{\mathrm{AP}}$ with $\varepsilon_{\mathrm{feas}} = \varepsilon_{\mathrm{opt}} = 10^{-4}$ for the sixteen instances of the packing problem.

Figures in Table 4 (as well as some of the graphics in Figure 1) show that eight out of the

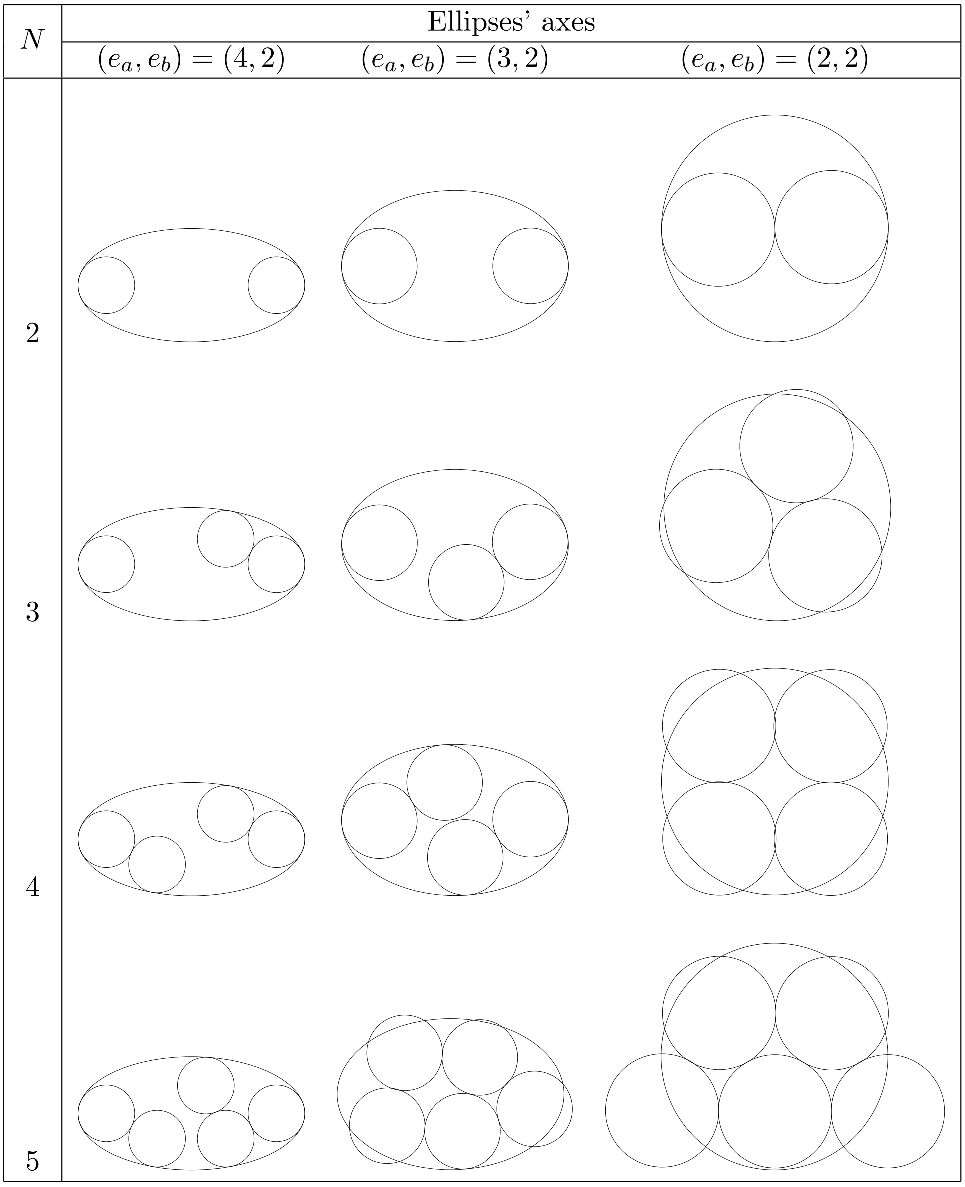

Figure 1: Graphical representation of twelve instances of the problem of packing circles within an ellipse.

sixteen considered instances were found to be infeasible. Among the infeasible instances, a different behaviour of the algorithms can be distinguished between instances with $(e_a, e_b) = (2, 1)$ and $N = 2, 3, 4, 5$ (last four lines in the table) and the other four infeasible instances ($(e_a, e_b) = (3, 2)$ with $N = 5$ and $(e_a, e_b) = (2, 2)$ with $N = 3, 4, 5$).

In the four instances with $(e_a, e_b) = (2, 1)$, infeasibility of the first augmented Lagrangian subproblem was detected by the αBB method at its first iteration, i.e. considering the original box constraints of the subproblem without further divisions. Due to the lack of the optional information regarding linear underestimators of the objective function and the constraints, infeasibility is being detected by interval analysis applied to the constraints within the inner solver αBB, that finds a constraint that can not be satisfied within the original box, i.e. that proves that the feasible set is empty. This is a very simple situation that would have been detected in a stage previous to the application of any global optimization algorithm. At least, these four examples show that the current implementation of the proposed methods performs as well as possible in these simple cases. Since no single minimization is done in those four instances, this is why there is nothing to be drawn to illustrate them in Figure 1.

In the other four instances detected to be infeasible, infeasibility was detected at Step 2 of Algorithms $\mathrm{ALABB}_{\mathrm{GP}}$ and $\mathrm{ALABB}_{\mathrm{AP}}$. In those four cases, the performances of both algorithms are mostly indistinguishable. Regarding the final infeasible point delivered by the methods, the nice symmetric pictures in Figure 1 show that these solutions are global minimizers of an infeasibility measure, as proved in Theorems 2.1 and 3.1. On the other hand, note that, as claimed, instances have been proven to be infeasible in a finite number of iterations. A short comment regarding the computation of c_k at Step 2 of Algorithms $\mathrm{ALABB}_{\mathrm{GP}}$ and $\mathrm{ALABB}_{\mathrm{AP}}$ is in order. c_k is computed with the sole purpose of detecting infeasibility, and the smaller its value the greater the chance of detecting infeasibility at the initial iterations of the methods is. As pointed out in Subsection 4.1, an interval $[f^{\min}, f^{\max}]$ such that $f^{\min} \leq f(x) \leq f^{\max}$ is computed by interval analysis and c_k is computed as defined in (58). The four instances in which infeasibility is being detected at Step 2 of Algorithms $\mathrm{ALABB}_{\mathrm{GP}}$ and $\mathrm{ALABB}_{\mathrm{AP}}$ are the ones with $(e_a, e_b) = (3, 2)$ and $N = 5$; and $(e_a, e_b) = (2, 2)$ and $N = 3, 4, 5$. For those instances, the interval $[f^{\min}, f^{\max}]$ computed by interval analysis for the original box Ω_1 is given by $[-520, 520]$, $[-96, 96]$, $[-192, 192]$, and $[-320, 320]$, respectively. However, since the objective function is the sum of squares, it is clear that $f^{\min} \geq 0$ (it is equally clear that this inequality is sharp). Moreover, maximizing the objective function over Ω_1 as suggested at the end of Subsection 4.1, we arrived at $f^{\max}$ equal to 312, 64, 128, and 192, respectively. Using these tighter intervals, the value of c_k computed as in (58) is strictly smaller than the one considered in the numerical experiments depicted in Table 4. A new numerical experiment was done considering those four infeasible instances and using the tighter intervals for computing c_k. Results were identical for three out of the four instances. For the instance given by $(e_a, e_b) = (2, 2)$ and $N = 4$ both algorithms stopped one iteration in advance (using 5 augmented Lagrangian iterations instead of 6). By saving the last augmented Lagrangian iteration, one less subproblem was solved and the total number of Branch-and-Bound nodes was reduced to $1,485,403$ (and 1043.18 seconds of CPU time) for Algorithm $\mathrm{ALABB}_{\mathrm{GP}}$ and to $1,485,059$ (and 1051.43 seconds of CPU time) for Algorithm $\mathrm{ALABB}_{\mathrm{AP}}$.

In the remaining eight feasible instances (that are not the main focus of the present subsection), both algorithms also presented a very similar behaviour, Algorithm $\mathrm{ALABB}_{\mathrm{AP}}$ being a little bit more efficient than Algorithm $\mathrm{ALABB}_{\mathrm{GP}}$. Algorithm $\mathrm{ALABB}_{\mathrm{AP}}$ uses 37%, 11%, and 47% fewer Branch-and-Bound nodes than Algorithm $\mathrm{ALABB}_{\mathrm{GP}}$ in three out of the eight instances, and they both use almost the same number of nodes in the remaining five instances. Last but not least, the performance of the methods presented all along the numerical results section should be taken as an illustration of the capabilities and drawbacks of the introduced methods, taking into account that they are highly dependent on the arbitrary problems' formulations being used and on the optional (additional) information accompanying each of them.

5 Conclusions and future research

The codes used to illustrate our theory and to solve the problems in the numerical sections of this paper are available in *http://www.ime.usp.br/~egbirgin/*. They probably represent a useful practical tool for solving global nonlinear programming problems employing the augmented Lagrangian technique. This software relies on the rigorous theory presented in Sections 2 and 3 of the present paper, by means of which we are able to compute solutions with guaranteed certificates of precision or, perhaps, infeasibility. As far as we know, this is the first paper in which this type of results are presented in the augmented Lagrangian context. Moreover, the results presented here complement those of [21] in the sense of broadening the scope of applicability of αBB in the direction of the general nonlinear programming field. As is usual in the nonlinear optimization world, we do not claim the universal effectiveness of our approach. The augmented Lagrangian approach enjoys some interesting features that are useful for problems with structures exhaustively studied in many other papers (see, for example, [25]). In particular, even local implementations of the augmented Lagrangian methods seem to provide global minimizers of constrained optimization problems more often than other optimization solvers [5]. This is due to the modular structure of the method, which allows one to employ opportunistic strategies for solving suproblems which are not necessarily linked to theory but are extremely useful in practice. In this sense, the results presented here, that are directly applicable to the field of global optimization, also help to enlighten the behavior of practical local PHR-like augmented Lagrangian algorithms.

In the recent book [32] and many papers on Mechanical Engineering applications (see [32] and the reference therein), Z. Dostál has shown the effectivity of the PHR augmented Lagrangian approach for solving convex quadratic programming problems. In the preface of the book, he emphasizes that the reliability and efficiency of augmented Lagrangian techniques is linked to problem conditioning characteristics that are present in its main branch of applications. A challenging problem is to combine Dostál convex techniques with the global techniques presented in the present paper for the effective solution of possibly large-scale nonconvex quadratic programming problems.

References

[1] C. S. Adjiman, I. P. Androulakis, C. D. Maranas, and C. A. Floudas, A global optimization method αBB for process design, *Computers and Chemical Engineering* 20, pp. S419–424, 1996.

[2] C. S. Adjiman, S. Dallwig, C. A. Floudas and A. Neumaier, A global optimization method, αBB, for general twice-differentiable constrained NLPs – I. Theoretical Advances, *Computers & Chemical Engineering* 22, pp. 1137–1158, 1998.

[3] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, A global optimization method, αBB, for general twice-differentiable constrained NLPs – II. Implementation and computational results, *Computers & Chemical Engineering* 22, pp. 1159–1179, 1998.

[4] F. A. Al-Khayyal and H. D. Sherali, On finitely terminating Branch-and-Bound algorithms for some global optimization problems, *SIAM Journal on Optimization* 10, pp. 1049–1057, 2000.

[5] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, On augmented Lagrangian methods with general lower-level constraints, *SIAM Journal on Optimization* 18, pp. 1286–1309, 2007.

[6] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification, *Mathematical Programming* 111, pp. 5–32, 2008.

[7] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. Second-order negative-curvature methods for box-constrained and general constrained optimization, *Computational Optimization and Applications* 45, pp. 209-236, 2010.

[8] R. Andreani, G. Haeser, and J. M. Martínez, On sequential optimality conditions for smooth constrained optimization, *Optimization* 60, pp. 627–641, 2011.

[9] R. Andreani, G. Haeser, M. L. Schuverdt, and P. J. S. Silva, A relaxed constant positive linear dependence constraint qualification and applications, to appear in *Mathematical Programming.*

[10] R. Andreani, G. Haeser, M. L. Schuverdt, and P. J. S. Silva, Two new weak constraint qualifications and applications, to appear in SIAM Journal on Optimization.

[11] R. Andreani, J. M. Martínez, and M. L. Schuverdt, On the relation between the Constant Positive Linear Dependence condition and quasinormality constraint qualification, *Journal of Optimization Theory and Applications* 125, pp. 473–485, 2005.

[12] R. Andreani, J. M. Martínez, and B. F. Svaiter. A new sequential optimality condition for constrained optimization and algorithmic consequences, *SIAM Journal on Optimization* 20, pp. 3533–3554, 2010.

[13] M. Andretta, E. G. Birgin, and J. M. Martínez, Partial Spectral Projected Gradient method with Active-Set Strategy for Linearly Constrained Optimization, *Numerical Algorithms* 53, pp. 23–52, 2010.

[14] I. P. Androulakis, C. D. Maranas, and C. A. Floudas, αBB: A global optimization method for general constrained nonconvex problems, *Journal of Global Optimization* 7, pp. 337–363, 1995.

[15] C. Audet and J. E. Dennis Jr., A progressive barrier for derivative-free nonlinear programming, *SIAM Journal on Optimization* 20, pp. 445–472, 2009.

[16] J. F. Bard, *Practical bilevel optimization. Algorithms and applications*, Kluwer Book Series: Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[17] E. G. Birgin, L. H. Bustamante, H. F. Callisaya, and J. M. Martínez, Packing Circles within Ellipses, *submitted.*

[18] E. G. Birgin, E. V. Castelani, A. L. Martinez, and J. M. Martínez, Outer trust-region method for constrained optimization, *Journal of Optimization Theory and Applications* 150, pp. 142–155, 2011.

[19] E. G. Birgin, R. Castillo, and J. M. Martínez, Numerical comparison of augmented Lagrangian algorithms for nonconvex problems, *Computational Optimization and Applications* 31, pp. 31–56, 2005.

[20] E. G. Birgin, D. Fernández, and J. M. Martínez, On the boundedness of penalty parameters in an augmented Lagrangian method with constrained subproblems, *Optimization Methods and Software* (DOI: 10.1080/10556788.2011.556634).

[21] E. G. Birgin, C. A. Floudas, and J. M. Martínez, Global minimization using an augmented Lagrangian method with variable lower-level constraints, *Mathematical Programming* 125, pp. 139–162, 2010.

[22] E. G. Birgin, C. A. Floudas, and J. M. Martínez, Global minimization using an augmented Lagrangian method with variable lower-level constraints, Technical Report MCDO220107 (see http://www.ime.usp.br/∼egbirgin/), Department of Applied Mathematics, Unicamp, Brazil, 2007.

[23] E. G. Birgin and J. M. Martínez, Large-scale active-set box-constrained optimization method with spectral projected gradients, *Computational Optimization and Applications* 23, pp. 101–125, 2002.

[24] E. G. Birgin and J. M. Martínez, Improving ultimate convergence of an augmented Lagrangian method, *Optimization Methods and Software 23*, pp. 177–195, 2008.

[25] E. G. Birgin and J. M. Martínez, Practical augmented Lagrangian methods, in *Encyclopedia of Optimization*, C. A. Floudas and P. M. Pardalos, editors. Second Edition. Springer, pp. 3013–3023, 2009.

[26] E. G. Birgin and J. M. Martínez, Augmented Lagrangian method with nonmonotone penalty parameters for constrained optimization, *Computational Optimization and Applications* 51, pp. 941–965, 2012.

[27] E. G. Birgin, J. M. Martínez, and L. Prudente, Global Nonlinear Programming with possible infeasibility and finite termination, Technical Report MCDO250612 (see http://www.ime.usp.br/∼egbirgin/), Department of Applied Mathematics, Unicamp, Brazil, 2012.

[28] H. F. Callisaya, *Empacotamento em quadráticas*, PhD Thesis, Institute of Mathematics, Statistics and Scientific Computing, University of Campinas, 2012.

[29] E. V. Castelani, A. L. Martinez, J. M. Martínez, and B. F. Svaiter, Addressing the greediness phenomenon in nonlinear programming by means of proximal augmented Lagrangians, *Computational Optimization and Applications* 46, pp. 229–245, 2010.

[30] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust Region Methods*, MPS/SIAM Series on Optimization, SIAM, Philadelphia, 2000.

[31] M. A. Diniz-Ehrhardt, J. M. Martínez, and L. G. Pedroso, Derivative-free methods for nonlinear programming with general lower-level constraints, *Computational & Applied Mathematics* 30, pp. 19–52, 2011.

[32] Z. Dostál, *Optimal Quadratic Programming Algorithms*, Springer, New York, 2009.

[33] Z. Dostál, A. Friedlander, and S. A. Santos, Augmented Lagrangians with adaptive precision control for quadratic programming with simple bounds and equality constraints, *SIAM Journal on Optimization* 13, pp. 1120–1140, 2003.

[34] J. Eckstein, A practical general approximation criterion for methods of multipliers based on Bregman distances, *Mathematical Programming* 96, pp. 61–86, 2003.

[35] C. A. Floudas, *Deterministic global optimization: theory, methods and application*, Kluwer Academic Publishers, DorDrecht, Boston, London, 1999.

[36] C. A. Floudas, I. G. Akrotirianakis, S. Caratzoulas, C. A. Meyer, and J. Kallrath, Global optimization in the 21st century: Advances and challenges, *Computers and Chemical Engineering* 29, pp. 1185–1202, 2005.

[37] C. A. Floudas and C. E. Gounaris, A review of recent advances in global optimization, *Journal of Global Optimization* 45, pp. 3–38, 2009.

[38] C. A. Floudas and V. Visweeswaran, A global optimization algorithm (GOP) for certain classes of nonconvex NLPs – I. Theory, *Computers & Chemical Engineering* 14, pp. 1397–1417, 1990.

[39] D.Y. Gao, *Duality Principles in Nonconvex Systems: Theory, Methods, and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[40] D. Y. Gao, Canonical dual transformation method and generalized triality theory in nonsmooth global optimization, *Journal of Global Optimization* 17, pp. 127–160, 2000.

[41] D. Y. Gao, Perfect duality theory and complete solutions to a class of global optimization problems, generalized triality theory in nonsmooth global optimization, *Optimization* 52, pp. 467–493, 2003.

[42] D. Y. Gao, Canonical duality theory and solutions to constrained nonconvex quadratic programming, *Journal of Global Optimization* 29, pp. 337–399, 2004.

[43] D. Y. Gao, Complete solutions and extremality criteria to polynomial optimization problems, *Journal of Global Optimization* 35, pp. 131–143, 2006.

[44] D. Y. Gao, Solutions and optimality to box constrained nonconvex minimization problems, *Journal of Industry and Management Optimization* 3, pp. 1–12, 2007.

[45] W. W. Hager, Analysis and implementation of a dual algorithm for constrained optimization, *Journal of Optimization Theory and Applications* 79, pp. 427–462, 1993.

[46] M. R. Hestenes, Multiplier and gradient methods, *Journal of Optimization Theory and Applications* 4, pp. 303–320, 1969.

[47] R. Horst, P. M. Pardalos, and M. V. Thoai, *Introduction to Global Optimization*, Kluwer Book Series: Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[48] R.B. Kearfott, M. Dawande, K. Du, and C. Hu, Algorithm 737: INTLIB: A portable Fortran 77 interval standard-function library, *ACM Transactions on Mathematical Software* 20, pp. 447–459, 1994.

[49] R. M. Lewis and V. Torczon, A direct search approach to nonlinear programming problems using an augmented Lagrangian method with explicit treatment of linear constraints, Technical Report WM-CS-2010-01, College of William & Mary, Department of Computer Sciences, 2010.

[50] L. Liberti, Reduction constraints for the global optimization of NLPs, *International Transactions in Operational Research* 11, pp. 33–41, 2004.

[51] L. Liberti, Writing global optimization software, in L. Liberti and N. Maculan (eds.), *Global Optimization: from Theory to Implementation*, pp. 211–262, Springer, Berlin, 2006.

[52] L. Liberti and C. C. Pantelides, An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms, *Journal of Global Optimization* 36, pp. 161–189, 2006.

[53] H. Z. Luo, X. L. Sun, and D. Li, On the convergence of augmented Lagrangian methods for constrained global optimization, *SIAM Journal on Optimization* 18, pp. 1209–1230, 2007.

[54] J. M. Martínez and L. F. Prudente, Handling infeasibility in a large-scale nonlinear optimization algorithm, *Numerical Algorithms* 60, pp. 263–277, 2012.

[55] B. A. Murtagh and M. A. Saunders, *MINOS 5.4 User's Guide*. System Optimization Laboratory, Department of Operations Research, Standford University, CA.

[56] A. Neumaier, Complete search in continuous global optimization and constraints satisfaction, *Acta Numerica* 13, pp. 271–369, 2004.

[57] A. Neumaier, O. Shcherbina, W. Huyer and T. Vinkó, A comparison of complete global optimization solvers, *Mathematical Programming* 103, pp. 335–356, 2005.

[58] G. Pacelli and M. C. Recchioni, An interior point algorithm for global optimal solutions and KKT points, *Optimization Methods & Software* 15, pp. 225–256, 2001.

[59] M. J. D. Powell, A method for nonlinear constraints in minimization problems, in *Optimization*, R. Fletcher (ed.), Academic Press, New York, NY, pp. 283–298, 1969.

[60] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran: The Art of Scientific Computing*, Cambridge University Press, New York, 1992.

[61] R. T. Rockafellar, Augmented Lagrange multiplier functions and duality in nonconvex programming, *SIAM Journal on Control and Optimization* 12, pp. 268–285, 1974.

[62] H. S. Ryoo and N. V. Sahinidis, A branch-and-reduce approach to global optimization *Journal of Global Optimization* 8, pp. 107–138, 1996.

[63] N. V. Sahinidis and M. Tawarmalani, *BARON 9.0.4: Global Optimization of Mixed-Integer Nonlinear Programs*, User's manual, 2010.

[64] H. D. Sherali and W. P. Adams, *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*, Kluwer Book Series: Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[65] H. D. Sherali, A. Alameddine, and T. S. Glickman, Biconvex models and algorithms for risk management problems, *American Journal of Mathematical and Management Science* 14, pp. 197–228, 1995.

[66] H. D. Sherali and J. Desai, A global optimization RLT-based approach for solving the hard clustering problem, *Journal of Global Optimization* 32, pp. 281–306, 2005.

[67] H. D. Sherali and J. Desai, A global optimization RLT-based approach for solving the fuzzy clustering problem, *Journal of Global Optimization* 33, pp. 597–615, 2005.

[68] H. D. Sherali and C. H. Tuncbilek, A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique, *Journal of Global Optimization* 2, pp. 101-112, 1992.

[69] H. D. Sherali and C. H. Tuncbilek, New reformulation-linearization/convexification relaxations for univariate and multivariate polynomial programming problems, *Operations Research Letters* 21, pp. 1–10, 1997.

[70] H. D. Sherali and H. Wang, Global optimization of nonconvex factorable programming problems, *Mathematical Programming* 89, pp. 459–478, 2001.

[71] E. M. B. Smith and C. C. Pantelides, A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs, *Computers & Chemical Engineering* 23, pp. 457–478, 1999.

[72] M. Tawarmalani and N. V. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming*, Kluwer Book Series: Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[73] M. Tawarmalani and N. V. Sahinidis, A polyhedral branch-and-cut approach to global optimization, *Mathematical Programming* 103, pp. 225–249, 2005.

[74] H. Tuy, *Convex Analysis and Global Optimization*, Kluwer Book Series: Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[75] V. Visweeswaran and C. A. Floudas, A global optimization algorithm (GOP) for certain classes of nonconvex NLPs – II. Application of theory and test problems, *Computers & Chemical Engineering* 14, pp. 1419–1434, 1990.

[76] Z. B. Zabinsky, *Stochastic Adaptive Search for Global Optimization*, Kluwer Book Series: Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.