Global minimization using an Augmented Lagrangian method with variable lower-level constraints

E. G. Birgin * C. A. Floudas † J. M. Martínez ‡

November 22, 2006

Abstract

A novel global optimization method based on an Augmented Lagrangian framework is introduced for continuous constrained nonlinear optimization problems. At each outer iteration the method requires the ε-global minimization of the Augmented Lagrangian with simple constraints. Global convergence to an ε-global minimizer of the original problem is proved. The subproblems are solved using the αBB method. Numerical experiments are presented.

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

1 Introduction

Global optimization has ubiquitous applications in all branches of engineering sciences and applied sciences. During the last decade several textbooks addressed different facets of global optimization theory and applications [17, 28, 33, 44, 77, 86, 87, 91]. Recent review papers have also appeared [29, 68].

The Augmented Lagrangian methodology based on the Powell-Hestenes-Rockafellar [43, 70, 72] formula has been successfully used for defining practical nonlinear programming algorithms [12, 13, 19, 23]. Convergence to KKT points was proved using the Constant Positive Linear Dependence constraint qualification [11], which strengthens the results based on the classical regularity condition [18, 23].

*Department of Computer Science IME-USP, University of São Paulo, Rua do Matão 1010, Cidade Universitária, 05508-090, São Paulo SP, Brazil. This author was supported by PRONEX-Optimization (PRONEX - CNPq / FAPERJ E-26 / 171.164/2003 - APQ1), FAPESP (Grants 06/53768-0 and 06/51827-9) and CNPq (PROSUL 490333/2004-4). e-mail: egbirgin@ime.usp.br

†Department of Chemical Engineering, Princeton University, Princeton, NJ 08544, USA. This author was supported by the National Science Foundation and the National Institute of Health (R01 GM52032). e-mail: floudas@titan.princeton.edu

‡Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. This author was supported by PRONEX-Optimization (PRONEX - CNPq / FAPERJ E-26 / 171.164/2003 - APQ1), FAPESP (Grant 06/53768-0) and CNPq. e-mail: martinez@ime.unicamp.br

In this work, we consider the Augmented Lagrangian method introduced in [13] and we modify it in such a way that, at each outer iteration, we find an ε-*global* minimizer of the subproblem. In the definition of the subproblem we introduce an important modification with respect to [13]: besides the lower level constraints we include constraints that incorporate information about the global solution of the nonlinear programming problem. A theorem of convergence to ε-global minimizers is presented.

We consider linear constraints on the lower-level set, and additional valid linear constraints which result from outer approximations of the feasible region and hence incorporate the global optimum information. This allows us to use the αBB [3, 4, 5, 15, 28, 50, 51, 57, 58, 59] method and its convex underestimation techniques [5, 6, 8, 9, 21, 42, 60, 63, 64, 66] for the subproblems in such a way that the underestimation techniques are applied just to the Augmented Lagrangian function and not to the constraints. The αBB global optimization approach has been applied to various problems that include molecular conformations in protein folding [16, 46, 47, 48, 49, 89], parameter estimation [24, 25, 26, 27], and phase equilibrium [39, 40, 41, 61, 62]. Mixed-integer nonlinear models arising in process synthesis, design and operations problems [1, 2, 7, 22, 30, 31, 52] represent additional important application areas.

It is important to emphasize that many global optimization techniques for nonlinear programming problems, e.g., [4, 5, 10, 15, 32, 34, 35, 36, 37, 38, 53, 54, 69, 74, 75, 78, 79, 80, 81, 82, 83, 88]. However, to our knowledge, none of them is based on Augmented Lagrangians. Moreover, as a consequence of using the Augmented Lagrangian approach combined with the αBB method and its convex α-underestimation techniques, the method introduced in this paper does not rely on the functional form of the functions involved in the problem definition (objective function and constraints), apart from their continuity and differentiability. More precisely, while the method can take advantage of well known underestimators and relaxations for several kinds of functional forms, it can also deal with functional forms for which underestimators and relaxations have not been developed yet.

This paper is organized as follows. In Section 2 we describe the Augmented Lagrangian deterministic global optimization algorithm. The convergence to ε-global minimizers is proved in Section 3. In Section 4 we describe the global optimization of the Augmented Lagrangian subproblems. Numerical results are given in Section 5. In Section 6 we draw some conclusions.

Notation.

If $v \in I\!R^n$, $v = (v_1, \ldots, v_n)$, we denote $v_+ = (\max\{0, v_1\}, \ldots, \max\{0, v_n\})$.

If $K = (k_1, k_2, \ldots) \subset I\!N$ (with $k_j < k_{j+1}$ for all j), we denote $K \underset{\infty}{\subset} I\!N$.

The symbol $\|\cdot\|$ will denote the Euclidian norm.

2 The overall algorithm

The problem to be addressed 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 closed. A typical set Ω consists of "easy" constraints such as linear constraints and box constraints. By easy we mean that a suitable algorithm for local minimization is available.

Assumption 1. From now on we will assume that there exists a global minimizer z of the problem.

We define the following Augmented Lagrangian function:

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

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

Algorithm 2.1

Let $\lambda_{\min} < \lambda_{\max}$, $\mu_{\max} > 0$, $\gamma > 1$, $0 < \tau < 1$. Let $\{\varepsilon_k\}$ be a sequence of nonnegative numbers such that $\lim_{k\to\infty} \varepsilon_k = \varepsilon \geq 0$. 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 $\rho_1 > 0$. Initialize $k \leftarrow 1$.

Step 1. Let $P_k \subset I\!R^n$ be a closed set such that a global minimizer z (the same for all k) belongs to P_k. Find an ε_k-global minimizer x^k of the problem Min $L_{\rho_k}(x, \lambda^k, \mu^k)$ subject to $x \in \Omega \cap P_k$. That is $x^k \in \Omega \cap P_k$ is such that:

$$L_{\rho_k}(x^k, \lambda^k, \mu^k) \leq L_{\rho_k}(x, \lambda^k, \mu^k) + \varepsilon_k \tag{3}$$

for all $x \in \Omega \cap P_k$. The ε_k-global minimum can be obtained using a deterministic global optimization approach, such as the αBB method.

Step 2. Define

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

If $k = 1$ or

$$\max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\} \leq \tau \ \max\{\|h(x^{k-1})\|_\infty, \|V^{k-1}\|_\infty\}, \tag{4}$$

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

Step 3. 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.

Remark. In the implementation, we will compute $\lambda_i^{k+1} = \max\{\lambda_{\min}, \min\{\lambda_i^k + \rho h_i(x^k), \lambda_{\max}\}\}$ and $\mu_i^{k+1} = \max\{0, \min\{\mu^k + \rho g_i(x^k), \mu_{\max}\}\}$.

We emphasize that the deterministic global optimization method αBB will not use the point x^{k-1} as "initial approximation" as most local optimization solvers do. In fact, the concept of "initial point" has no meaning at all in this case. The information used by the outer iteration k is the set of approximate Lagrange multipliers computed after iteration $k-1$, and nothing else.

3 Convergence to an ε-global minimum

Theorem 1. *Assume that the sequence $\{x^k\}$ is well defined and admits a limit point x^*. Then, x^* is feasible.*

Proof. Since Ω is closed and $x^k \in \Omega$, we have that $x^* \in \Omega$. We consider two cases: $\{\rho_k\}$ bounded and $\rho_k \to \infty$. If $\{\rho_k\}$ is bounded, there exists k_0 such that $\rho_k = \rho_{k_0}$ for all $k \geq k_0$. Therefore, for all $k \geq k_0$, (4) holds. This implies that $\|h(x^k)\| \to 0$ and $\|V^k\| \to 0$. So, $g_i(x^k)_+ \to 0$ for all $i = 1, \ldots, p$. So, the limit point is feasible.

Now, assume that $\rho_k \to \infty$. Let z be as in Step 1. Therefore, z is feasible. So, $\|h(z)\| = \|g(z)_+\| = 0$. Suppose, by contradiction, that x^* is not feasible. Therefore,

$$\|h(x^*)\|^2 + \|g(x^*)_+\|^2 > \|h(z)\|^2 + \|g(z)_+\|^2.$$

Let K be an infinite sequence of indices such that $\lim_{k\in K} x^k = x^*$. Since h and g are continuous, λ^k, μ^k are bounded and $\rho_k \to \infty$, there exists $c > 0$ such that for $k \in K$ large enough:

$$\left\|h(x^k) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2 > \left\|h(z) + \frac{\lambda^k}{\rho_k}\right\|^2 + \left\|\left(g(z) + \frac{\mu^k}{\rho_k}\right)_+\right\|^2 + c.$$

Therefore,

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

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

Since $\lim_{k\in K} x^k = x^*$ and f is continuous, for $k \in K$ large enough

$$\frac{\rho_k c}{2} + f(x^k) - f(z) > \varepsilon_k.$$

Therefore,

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

Now, since z is a global minimizer, we have that $z \in \Omega \cap P_k$ for all k. Therefore, the inequality above contradicts the definition of x^k. □

Theorem 2. *Under the same assumptions of Theorem 1, every limit point x^* of a sequence $\{x^k\}$ generated by Algorithm 1 is an ε-global minimizer of the problem.*

Proof. Let $K \underset{\infty}{\subset} I\!N$ be such that $\lim_{k\in K} x^k = x^*$. By Theorem 1, x^* is feasible. Let $z \in \Omega$ as as in Step 1. Then, $z \in P_k$ for all k.

We consider two cases: $\rho_k \to \infty$ and $\{\rho_k\}$ bounded.

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

$$f(x^k)+\frac{\rho_k}{2}\left[\left\|h(x^k)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(x^k)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right] \leq f(z)+\frac{\rho_k}{2}\left[\left\|h(z)+\frac{\lambda^k}{\rho_k}\right\|^2+\left\|\left(g(z)+\frac{\mu^k}{\rho_k}\right)_+\right\|^2\right]+\varepsilon_k \tag{5}$$

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

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

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

Therefore, by (5),

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

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

$$f(x^*) \leq f(z) + \varepsilon.$$

Since z is a global minimizer, it turns out that x^* is an ε-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 $k \geq k_0$. Therefore, by the definition of Algorithm 1, we have:

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

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 $h(z) = 0$,

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

for all $k \geq k_0$. Let $K_1 \underset{\infty}{\subseteq} K$ be such that

$$\lim_{k\in K_1} \lambda^k = \lambda^*, \lim_{k\in K_1} \mu^k = \mu^*.$$

By the feasibility of x^*, taking limits in the inequality above for $k \in K_1$, we get:

$$f(x^*)+\frac{\rho_{k_0}}{2}\left[\left\|\frac{\lambda^*}{\rho_{k_0}}\right\|^2+\left\|\left(g(x^*)+\frac{\mu^*}{\rho_{k_0}}\right)_+\right\|^2\right] \leq f(z)+\frac{\rho_{k_0}}{2}\left[\left\|\frac{\lambda^*}{\rho_{k_0}}\right\|^2+\left\|\frac{\mu^*}{\rho_{k_0}}\right\|^2\right]+\varepsilon.$$

Therefore,

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

Thus,

$$f(x^*) + \frac{\rho_{k_0}}{2}\sum_{i=1}^{p}\left(g_i(x^*) + \frac{\mu_i^*}{\rho_{k_0}}\right)_+^2 \leq f(z) + \frac{\rho_{k_0}}{2}\sum_{i=1}^{p}\left(\frac{\mu_i^*}{\rho_{k_0}}\right)^2 + \varepsilon. \tag{6}$$

Now, if $g_i(x^*) = 0$, since $\mu_i^*/\rho_{k_0} \geq 0$, we have that

$$\left(g_i(x^*) + \frac{\mu_i^*}{\rho_{k_0}}\right)_+ = \frac{\mu_i^*}{\rho_{k_0}}.$$

Therefore, by (6),

$$f(x^*) + \frac{\rho_{k_0}}{2}\sum_{g_i(x^*)<0}\left(g_i(x^*) + \frac{\mu_i^*}{\rho_{k_0}}\right)_+^2 \leq f(z) + \frac{\rho_{k_0}}{2}\sum_{g_i(x^*)<0}\left(\frac{\mu_i^*}{\rho_{k_0}}\right)^2 + \varepsilon. \tag{7}$$

But, by Step 2 of Algorithm 1 , $\lim_{k\in\infty}\max\{g_i(x^k), -\mu_i^k/\rho_{k_0}\} = 0$. Therefore, if $g_i(x^*) < 0$ we necessarily have that $\mu_i^* = 0$. Therefore, (7) implies that $f(x^*) \leq f(z) + \varepsilon$. Since z is a global minimizer, the proof is complete. □

4 Global optimization of subproblems

In this section, we address the problem of finding $x^k \in \Omega \cap P_k$ satisfying (3). This problem is equivalent to the problem of finding an ε_k-global solution of the problem:

$$\text{Minimize } L_{\rho_k}(x, \lambda^k, \mu^k) \text{ subject to } x \in \Omega \cap P_k, \tag{8}$$

where $\Omega = \{x \in I\!R^n \mid Ax = b,\ Cx \leq d,\ l \leq x \leq u\}$ and $Ax = b$, $Cx \leq d$ and $l \leq x \leq u$ represent the linear equality, linear inequality and bound constraints of problem (1), respectively. The remaining constraints of problem (1) will be $h(x) = 0$, $g(x) \leq 0$. The role of P_k will be elucidated soon.

To solve problem (8), we introduced and implemented the αBB algorithm [4, 5, 6, 28, 57, 58] for the particular case of linear constraints and bounds. The αBB method is a deterministic global optimization method for nonlinear programming problems based on Branch & Bound. For bounding purposes, it uses the convex α-underestimator of the function being minimized that coincides with the function at the bounds of the box and whose maximum separation (distance to the objective function) is proportional to the box dimensions. Therefore, the smaller the box, the tighter the convex α-underestimator.

Based on the last observation, the αBB method consists of splitting the box-constraints domain into smaller subdomains in order to reduce the gap between an upper and a lower bound on the minimum of the problem. The upper bound is given by the smallest functional value obtained through local minimizations within the subdomains, while the lower bound comes from

the global minimization of the convex α-underestimators subject to the problem constraints. If, within a subdomain, the lower bound plus the prescribed tolerance ε_k is above the upper bound, the subdomain can be discarded as it clearly does not contain the solution of the problem (considering the tolerance ε_k). The same argument applies if, via interval analysis, it is shown the subdomain does not contain any promising point.

The constraints of the problem, or any other valid constraint, can also be used to substitute a subdomain $[\bar{l}, \bar{u}]$ for any other proper subdomain $[\hat{l}, \hat{u}]$. In our implementation, we considered just linear constraints in the process of reducing a subdomain $[\bar{l}, \bar{u}]$. Three sources of linear constraints were used, namely (a) linear constraints of the original problem; (b) linear relaxations (valid within the subdomain $[\bar{l}, \bar{u}]$) of the nonlinear penalized constraints; and (c) linear "cuts" of the form $L^U_{\rho_k}(x, \lambda^k, \mu^k) \leq L_{\rm ub}$, where $L^U_{\rho_k}(\cdot, \lambda^k, \mu^k)$ is a linear relaxation of $L_{\rho_k}(\cdot, \lambda^k, \mu^k)$ within $[\bar{l}, \bar{u}]$. Constraints of type (a) and (b) can be used to discard regions that do not contain feasible points of problem (1) and play a role in the definition of P_k. Constraints of type (c) eliminate regions that do not contain the global solution of (8).

Let us call B the original box of the problem and L the original polytope defined by the linear constraints. As a result of the αBB process, the original box is covered by t "small" boxes $B_1, \ldots, B_t$. For each small box B_i a polytope Q_i (defined by the relaxations) is given (perhaps $Q_i = I\!R^n$) and a new small box $\hat{B}_i$ such that $(Q_i \cap B_i) \subset \hat{B}_i \subset B_i$ is constructed. By construction, the set $P_k = \cup_{i=1}^t \hat{B}_i$ contains the feasible set of the problem. So, P_k contains the global minimizers of the problem. The αBB algorithm guarantees an ε_k-global minimizer on $L \cap P_k$, as required by the theory.

The algorithm starts with a list S of unexplored subdomains that initially has as unique element the original box domain of the problem. Then, for each subdomain in the list it does the following tasks: (i) reduce the subdomain; (ii) try to discard the subdomain via interval analysis or computing the global solution of the underestimating problem; (iii) if the subdomain cannot be discarded; perform a local minimization within the subdomain; (iv) finally, split the subdomain and add the new subdomains to S. The method stops when the list S is empty.

The description of the αBB algorithm, that follows very closely the algorithm introduced in [5], is as follows.

Algorithm 4.1: αBB

Step 1. *Initialization*

Set $S = \{[l, u]\}$ and $L_{\rm ub} = +\infty$.

Step 2. *Stopping criterion*

If S is empty, stop.

Step 3. *Choose a subdomain*

Choose $[\bar{l}, \bar{u}] \in S$ and set $S \leftarrow S \setminus [\bar{l}, \bar{u}]$.

Step 4. *Subdomain reduction and possible discarding*

Let $W_{[\bar{l},\bar{u}]}$ be a set of linear constraints valid within the subdomain $[\bar{l},\bar{u}]$ plus linear constraints satisfied by the global solution of (8). For $i = 1,\ldots,n$, compute $\hat{l}_i$ and $\hat{u}_i$ as

$$arg\min \pm x_i \text{ subject to } Ax = b,\ Cx \leq d, \bar{l} \leq x \leq \bar{u},\ x \in W_{[\bar{l},\bar{u}]}, \tag{9}$$

respectively. If the feasible set of (9) is empty, discard the subdomain $[\bar{l},\bar{u}]$ and go to Step 2.

Step 5. *Reduced subdomain discarding*

Step 5.1. Using interval analysis, compute $[L^{\min}_{[\hat{l},\hat{u}]}, L^{\max}_{[\hat{l},\hat{u}]}]$ such that

$$L^{\min}_{[\hat{l},\hat{u}]} \leq L_{\rho_k}(x,\lambda^k,\mu^k) \leq L^{\max}_{[\hat{l},\hat{u}]},\ \forall\ x \in [\hat{l},\hat{u}].$$

If $L^{\min}_{[\hat{l},\hat{u}]} + \varepsilon_k \geq L_{\text{ub}}$ then discard the reduced subdomain $[\hat{l},\hat{u}]$ and go to Step 2.

Step 5.2. Compute a convex underestimator $U_{[\hat{l},\hat{u}]}(x)$ of $L_{\rho_k}(x,\lambda^k,\mu^k)$ and find

$$y_1 \leftarrow arg\min U_{[\hat{l},\hat{u}]}(x) \text{ subject to } Ax = b,\ Cx \leq d,\ \hat{l} \leq x \leq \hat{u}.$$

If $U_{[\hat{l},\hat{u}]}(y_1) + \varepsilon_k \geq L_{\text{ub}}$ then discard the reduced subdomain $[\hat{l},\hat{u}]$ and go to Step 2.

Step 6. *Local optimization within subdomain*

Using y_1 as initial guess, compute

$$y_2 \leftarrow arg\min L_{\rho_k}(x,\lambda^k,\mu^k) \text{ subject to } Ax = b,\ Cx \leq d,\ \hat{l} \leq x \leq \hat{u}.$$

If $L_{\rho_k}(y_2,\lambda^k,\rho^k) < L_{\text{ub}}$ then set $L_{\text{ub}} \leftarrow L_{\rho_k}(y_2,\lambda^k,\rho^k)$ and $x_{\text{ub}} \leftarrow y_2$.

Step 7. *Split subdomain*

Split $[\hat{l},\hat{u}]$ in at least 2 proper subdomains. Add the new subdomains to S and go to Step 3.

Remarks.

1. The set S is implemented as a queue (see Steps 3 and 7). It is a very simple and problem-independent strategy that, in contrast to the possibility of using a stack, provides to the method a diversification that could result in fast improvements of L_{ub}.

2. At Step 7, we divide the subdomain in two subdomains splitting the range of the variable selected by the least reduced axis rule [5] in its middle point. Namely, we choose x_i such that $i = arg\min_j\{(\hat{u}_j - \hat{l}_j)/(u_j - l_j)\}$. If a variable appears linearly in the objective function of the subproblem (8) then its range does not need to be split at all and it is excluded from the least reduced axis rule. The variables that appear linearly in the Augmented Lagrangian are the ones that appear linearly in the objective function of the original problem (1) and do not appear in the penalized nonlinear constraints.

3. Set $W_{[\bar{l},\bar{u}]}$ at Step 4 is composed by linear relaxations (valid within the subdomain $[\bar{l},\bar{u}]$) of the penalized nonlinear constraints. We considered the convex and concave envelopes for bilinear terms and the convex envelope for concave functions [28], as well as tangent hyperplanes to the convex constraints.

4. By the definition of the Augmented Lagrangian function (2), it is easy to see that $f(x) \leq L_\rho(x,\lambda,\mu)\ \forall x$. So, any linear relaxation of the objective function $f(x)$ is also a linear relaxation of the Augmented Lagrangian function $L_\rho(x,\lambda,\mu)$. Therefore, a constraint of the form $f^U(x) \leq L_{\text{ub}}$, where $f^U(\cdot)$ is a linear relaxation of $f(\cdot)$ for all $x \in [\bar{l},\bar{u}]$ was also included in the definition of $W_{[\bar{l},\bar{u}]}$ at Step 4.

5. As a consequence of the definition of Ω in (8) (associated to the choice of the lower-level constraints), the optimization subproblems at Steps 5.2 and 6 are linearly constrained optimization problems. Moreover, by the definition of Ω and the fact that $W_{[\bar{l},\bar{u}]}$ at Step 4 is described by linear constraints, the $2n$ optimization problems at Step 4 are linear programming problems.

6. At Step 5.2, it is not necessary to complete the minimization process. It would be enough to find y such that $U_{[\hat{l},\hat{u}]}(z)+\varepsilon_k < L_{\text{ub}}$ to realize that the subdomain can not be discarded.

There is an alternative that represents a trade-off between effectiveness and efficiency. Step 4 can be repeated, substituting $\bar{l}$ and $\bar{u}$ by $\hat{l}$ and $\hat{u}$ in (9), respectively, while $\hat{l} \neq \bar{l}$ or $\hat{u} \neq \bar{u}$. Moreover, within the same loop, $\bar{l}_i$ and $\bar{u}_i$ can be replaced by $\hat{l}_i$ and $\hat{u}_i$ as soon as they are computed. Doing that, the order in which the new bounds are computed might influence the final result, and a strategy to select the optimal sequence of bounds updates might be developed.

The computation of the convex underestimator $U_{[\hat{l},\hat{u}]}(x)$ uses the convex α-underestimator for general nonconvex terms introduced in [58] and defined as follows:

$$U_{[\hat{l},\hat{u}]}(x) = L_{\rho_k}(x,\lambda^k,\mu^k) - \sum_{i=1}^{n} \alpha_i(\hat{u}-x)(x-\hat{l}),$$

where $\alpha_i, i = 1,\ldots,n$, are positive scalars large enough to ensure the convexity of $U_{[\hat{l},\hat{u}]}(x)$.

We used the Scaled Gerschgorin method [4, 5] to compute the α's. Given $d \in I\!R_{++}$ and $[H_{[\hat{l},\hat{u}]}]$, the interval Hessian of $L_{\rho_k}(x,\lambda^k,\mu^k)$ for the interval $[\hat{l},\hat{u}]$, we have

$$\alpha_i = \max\{0, -\frac{1}{2}(h_{ii}^{\min} - \sum_{j\neq i} |h|_{ij}\frac{d_j}{d_i})\}, i = 1,\ldots,n, \tag{10}$$

where $|h|_{ij} = \max\{|h_{ij}^{\min}|, |h_{ij}^{\max}|\}$ and $(h_{ij}^{\min}, h_{ij}^{\max})$ denotes element at position (i,j) of $[H_{[\hat{l},\hat{u}]}]$. As suggested in [4], we choose $d = \hat{u} - \hat{l}$.

The piecewise convex α-underestimator [65] can also be used instead of the convex α-underestimator to compute the Augmented Lagrangian underestimator at Step 5.2. The latter one may be significantly tighter than the first one, while its computation is more time consuming. It is worth mentioning that, although the theory of both underestimators was developed for twice-continuously differentiable functions, the continuity of the second derivatives, which

does not hold in the Augmented Lagrangian (2), is not necessary at all. The computation of the piecewise convex α-underestimator deserves further explanations.

Considering, for each interval $[\hat{l}_i, \hat{u}_i], i = 1, \ldots, n$, a partitioning in N_i subintervals with endpoints $\hat{l}_i \equiv \hat{v}_i^0, \hat{v}_i^1, \ldots, \hat{v}_i^{N_i} \equiv \bar{u}_i$, a definition of the piecewise convex α-underestimator, equivalent to the one introduced in [65], follows:

$$\begin{aligned}
\Phi_{[\hat{l},\hat{u}]}(x) &= L_{\rho_k}(x, \lambda^k, \mu^k) - \textstyle\sum_{i=1}^n q_i(x), \\
q_i(x) &= q_i^j(x), \text{ if } x \in [\hat{v}_i^{j-1}, \hat{v}_i^j], i = 1, \ldots, n, j = 1, \ldots, N_i, \\
q_i^j(x) &= \alpha_i^j(\hat{v}_i^j - x_i)(x_i - \hat{v}_i^{j-1}) + \beta_i^j x_i + \gamma_i^j, i = 1, \ldots, n, j = 1, \ldots, N_i.
\end{aligned}$$

In the definition above, α's, β's and γ's are such that $q_i(x), i = 1, \ldots, n$, are continuous and smooth and $\Phi_{[\hat{l},\hat{u}]}(x)$ is a convex underestimator of $L_{\rho_k}(x, \lambda^k, \mu^k)$ within the box $[\hat{l}, \hat{u}]$.

One way to compute the α's is to compute one $\alpha \in I\!R^n$ for each one of the $\prod_{i=1}^n N_i$ subdomains and set $\alpha_i^j, i = 1, \ldots, n, j = 1, \ldots, N_i$, as the maximum over all the $[\alpha]_i$'s of the subdomains included in the "slice" $[(\hat{l}_1, \ldots, \hat{l}_{i-1}, \hat{v}_i^{j-1}, \hat{l}_{i+1}, \ldots, \hat{l}_n), (\hat{u}_1, \ldots, \hat{u}_{i-1}, \hat{v}_i^j, \hat{u}_{i+1}, \ldots, \hat{u}_n)]$. In this way, $\prod_{i=1}^n N_i$ α's must be computed.

After having computed the α's, and considering that each interval $[\hat{l}_i, \hat{u}_i]$ is partitioned in N_i identical subintervals of size $s_i = (\hat{u}_i - \hat{l}_i)/N_i$, β's and γ's can be computed as follows: for each $i = 1, \ldots, n$,

$$\begin{aligned}
\beta_i^1 &= -\tfrac{s_i}{N_i} \textstyle\sum_{j=2}^{N_i}(N_i - j + 1)(\alpha_i^{j-1} + \alpha_i^j), \\
\gamma_i^1 &= -\beta_i^1 \hat{l}_i, \\
\beta_i^j &= \beta_i^{j-1} - s_i(\alpha_i^{j-1} + \alpha_i^j), & j = 2, \ldots, N_i, \\
\gamma_i^j &= \gamma_i^{j-1} + (\hat{l}_i + (j-1)s_i)s_i(\alpha_i^{j-1} + \alpha_i^j), & j = 2, \ldots, N_i.
\end{aligned}$$

Note that all the β's and γ's can be computed in $O(\sum_{i=1}^n N_i)$ operations using the formulae above, while the direct application of formulae present in [65] would imply in $O(\sum_{i=1}^n N_i^2)$ operations.

5 Numerical experiments

For solving linear programming problems we use subroutine `simplx` from the Numerical Recipes in Fortran [71]. For interval analysis calculations we use the INTLIB library [45]. Finally, to solve the linearly constrained optimization problems, we use GENLIN [14], a straightforward extension of GENCAN [20], an active-set method for bound-constrainted minimization. The codes are free for download in *www.ime.usp.br/∼egbirgin/*. They are written in Fortran 77 (double precision).

The method requires from the user subroutines to compute the objective function, the constraints and their first and second derivatives. In addition, it also requires subroutines to compute the interval objective function and the interval Hessian of the objective function, as well as the interval constraints and the interval Jacobian and interval Hessians of the constraints. Finally, the user must also provide two extra subroutines: (i) a first subroutine to indicate which variables appear nonlinearly in the objective function or appear in a nonlinear constraint; and

(ii) a second subroutine that computes the linear relaxations of the nonlinear constraints. This second subroutine is not mandatory and even an empty subroutine is a valid option. Note that the implemented method does not perform any kind of algebraic manipulation on the problems which are solved as stated by the user. On the other hand, the implemented algorithm provides several subroutines to empirically verify the correctness of the derivatives, the interval calculations and the relaxations of the constraints.

For the practical implementation of Algorithms 2.1 and 4.1, we set $\tau = 0.5$, $\gamma = 10$, $\lambda_{\min} = -10^{20}$, $\mu_{\max} = \lambda_{\max} = 10^{20}$, $\lambda^1 = 0$, $\mu^1 = 0$ 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. As stopping criterion we used $\max(\|h(x^k)\|_\infty, \|V^k\|_\infty) \leq 10^{-4}$. Several tests were done in order to analise the behaviour of the method in relation to the choice of the optimality gaps ε_k for the ε_k-global solution of the subproblems and ε for the ε-global solution of the original problem. All the experiments were run on a 3.2 GHz Intel(R) Pentium(R) with 4 processors, 1Gb of RAM and Linux Operating System. Compiler option "-O" was adopted.

We selected a set of 18 problems from the literature (see Appendix). In a first experiment, we solved the problems using the convex α-underestimator and using a variable $\varepsilon_k = \max\{\varepsilon, 10^{-k}\}$ and a fixed $\varepsilon_k = \varepsilon$. In both cases we considered $\varepsilon \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. Table 1 shows the results. In Table 1, n is the number of variables and m is the number of constraints. The number within parentheses is the number of linear constraints. "Time" is the total CPU time in seconds, "It" is the number of iterations of Algorithm 2.1 (outer iterations) which is equal to the number of subproblems (8) that are solved to ε_k-global optimality using the αBB approach, and "#Nodes" is the total number of iterations of Algorithm 4.1 (inner iterations). The method with the eight combinations of ε_k and ε found the same global minimum in all the problems. As expected, smaller gaps require more computations. However, the effort increase is not the same in all the problems, as it strongly depends on the tightness of the convex α-underestimator in relation to the Augmented Lagrangian function and the difficulty in closing that gap for each particular problem.

In a second experiment, we set $\varepsilon_k = \max\{\varepsilon, 10^{-k}\}$ and $\varepsilon = 10^{-4}$ (which provides the tightest global optimality certification) and compare the performance of the method using the convex α-underestimator and the piecewise convex α-underestimator with $N_i \equiv N \equiv 2\ \forall i$. Table 2 shows the results. In Table 2, $f(x^*)$ is the global minimum. As expected, both versions of the method found the global minimum reported in the literature up to the prescribed tolerances. The version that uses the convex α-underestimator required less CPU time while the version that used the piecewise convex α-underestimator used a smaller number of inner iterations.

It is well known that the performance of a global optimization method for solving a problem strongly depends on the particular way the problem is modeled. So, it is worth mentioning that all the test problems, apart from Problems 2(a–d), were solved using the models shown in the Appendix without any kind of algebraic manipulation or reformulation. One of the key problem features that affects the behaviour of the present approach is the number of variables that appear nonlinearly in the objective function or in a nonlinear constraint. Those are the variables that need to be branched in the αBB method. In the formulation of Problems 2(a–c)

Variable $\varepsilon_k = \max\{\varepsilon, 10^{-k}\}$														
Problem	n	m	$\varepsilon = 10^{-1}$			$\varepsilon = 10^{-2}$			$\varepsilon = 10^{-3}$			$\varepsilon = 10^{-4}$		
			Time	It	#Nodes	Time	It	#Nodes	Time	It	#Nodes	Time	It	#Nodes
1	5	3	9.44	9	59861	12.82	9	82167	15.83	9	103919	18.86	9	124897
2(a)	11	8(6)	0.09	8	78	0.11	8	88	0.13	8	100	0.13	8	104
2(b)	11	8(6)	0.64	13	563	0.70	13	613	0.73	13	651	0.76	13	671
2(c)	11	8(6)	0.07	8	60	0.09	8	72	0.17	8	84	0.16	8	88
2(d)	12	9(7)	0.04	2	20	0.06	2	28	0.14	3	63	0.23	4	110
3(a)	6	5	1.18	6	1476	3.47	6	5078	7.78	6	12688	12.07	6	20366
3(b)	2	1	0.19	2	1294	0.47	2	3046	1.39	3	8719	2.90	4	18016
4	2	1	0.00	2	68	0.00	2	80	0.01	3	129	0.01	4	182
5	10	10	0.00	1	3	0.00	1	3	0.00	1	3	0.00	1	3
6	2	1	0.00	2	22	0.00	2	24	0.00	3	41	0.00	4	60
7	3	3	0.03	7	765	0.03	7	797	0.03	7	817	0.04	7	817
8	2	1	0.01	5	343	0.01	5	427	0.01	5	467	0.01	5	493
9	2	4(2)	0.01	3	149	0.01	3	247	0.01	3	283	0.02	4	446
10	2	2(1)	0.10	6	2340	0.13	6	3032	0.14	6	3496	0.15	6	3918
11	6	6(6)	0.00	1	3	0.00	1	3	0.00	1	3	0.00	1	5
12	2	2	0.00	3	95	0.00	3	123	0.01	3	139	0.01	4	206
13	2	1	0.00	2	22	0.00	2	38	0.00	3	105	0.01	4	194
14	2	1	0.01	8	240	0.01	8	314	0.01	8	334	0.01	8	370
15	3	2(1)	0.10	8	866	0.23	8	2048	0.43	8	3750	0.47	8	4178
16	4	3(3)	0.00	1	1	0.00	1	1	0.00	1	1	0.00	1	1
17(a)	3	3(1)	0.01	1	99	0.02	2	266	0.03	3	565	0.06	4	1080
17(b)	3	3(2)	0.00	1	23	0.00	2	48	0.00	3	75	0.01	4	108
18	5	3(1)	0.02	6	116	0.07	6	362	0.12	6	698	0.15	6	902
Fixed $\varepsilon_k = \varepsilon$														
Problem	n	m	$\varepsilon = 10^{-1}$			$\varepsilon = 10^{-2}$			$\varepsilon = 10^{-3}$			$\varepsilon = 10^{-4}$		
			Time	It	#Nodes	Time	It	#Nodes	Time	It	#Nodes	Time	It	#Nodes
1	5	3	9.46	9	59861	12.83	9	82167	15.88	9	103919	18.90	9	124897
2(a)	11	8(6)	0.09	8	78	0.10	8	88	0.12	8	100	0.13	8	104
2(b)	11	8(6)	0.64	13	563	0.69	13	613	0.73	13	651	0.76	13	671
2(c)	11	8(6)	0.07	8	60	0.09	8	72	0.17	8	84	0.17	8	88
2(d)	12	9(7)	0.04	2	20	0.06	2	28	0.08	2	38	0.10	2	50
3(a)	6	5	1.17	6	1476	3.46	6	5078	7.79	6	12694	12.19	6	20598
3(b)	2	1	0.19	2	1294	0.66	2	4382	1.52	2	9580	2.22	2	14012
4	2	1	0.01	2	68	0.00	2	90	0.01	2	98	0.01	2	106
5	10	10	0.00	1	3	0.00	1	3	0.00	1	3	0.00	1	3
6	2	1	0.00	2	22	0.00	2	26	0.00	2	34	0.00	2	38
7	3	3	0.04	7	765	0.03	7	797	0.03	7	819	0.03	7	825
8	2	1	0.01	5	343	0.01	5	443	0.01	5	515	0.01	5	557
9	2	4(2)	0.01	3	149	0.01	3	293	0.02	3	403	0.02	3	493
10	2	2(1)	0.10	6	2340	0.13	6	3100	0.15	6	3692	0.17	6	4304
11	6	6(6)	0.00	1	5	0.00	1	5	0.00	1	5	0.00	1	5
12	2	2	0.00	3	95	0.00	3	137	0.01	3	181	0.01	3	201
13	2	1	0.00	2	22	0.00	2	38	0.00	2	78	0.00	2	102
14	2	1	0.01	8	240	0.01	8	322	0.01	8	342	0.02	8	386
15	3	2(1)	0.10	8	866	0.36	8	3250	1.21	8	10608	1.86	8	15992
16	4	3(3)	0.00	1	1	0.00	1	1	0.00	1	1	0.00	1	1
17(a)	3	3(1)	0.01	1	99	0.01	1	167	0.02	1	299	0.03	1	515
17(b)	3	3(2)	0.00	1	23	0.00	1	25	0.00	1	27	0.00	1	33
18	5	3(1)	0.02	6	116	0.06	6	362	0.11	6	698	0.15	6	934

Table 1: Performance of the global Augmented Lagrangian algorithm using different gaps for the global optimality of the subproblems and the original problem. In all the cases the method found the same global minimum.

Problem	n	m	αBB			Piecewise αBB (N=2)			$f(x^*)$
			Time	It	#Nodes	Time	It	#Nodes	
1	5	3	18.86	9	124897	42.57	9	117679	2.9313E−02
2(a)	11	8(6)	0.13	8	104	1.30	8	102	−4.0000E+02
2(b)	11	8(6)	0.76	13	671	2.23	13	671	−6.0000E+02
2(c)	11	8(6)	0.16	8	88	1.09	8	88	−7.5000E+02
2(d)	12	9(7)	0.23	4	110	3.44	4	110	−4.0000E+02
3(a)	6	5	12.07	6	20366	72.83	6	15264	−3.8880E−01
3(b)	2	1	2.90	4	18016	5.62	4	13954	−3.8881E−01
4	2	1	0.01	4	182	0.02	4	156	−9.4773E+00
5	10	10	0.00	1	3	0.01	1	3	−1.0000E+01
6	2	1	0.00	4	60	0.00	4	58	−6.6666E+00
7	3	3	0.04	7	817	0.04	7	561	2.0116E+02
8	2	1	0.01	5	493	0.02	5	443	3.7629E+02
9	2	4(2)	0.02	4	446	0.02	4	372	−2.8284E+00
10	2	2(1)	0.15	6	3918	0.17	6	3522	−1.1870E+02
11	6	6(6)	0.00	1	5	0.01	1	5	−1.3402E+01
12	2	2	0.01	4	206	0.01	4	170	7.4178E−01
13	2	1	0.01	4	194	0.01	4	160	−5.0000E−01
14	2	1	0.01	8	370	0.01	8	350	−1.6739E+01
15	3	2(1)	0.47	8	4178	1.47	8	3820	1.8935E+02
16	4	3(3)	0.00	1	1	0.00	1	1	−4.5142E+00
17(a)	3	3(1)	0.06	4	1080	0.18	4	1558	0.0000E+00
17(b)	3	3(2)	0.01	4	108	0.01	4	106	0.0000E+00
18	5	3(1)	0.15	6	902	0.22	6	758	7.0492E−01

Table 2: Performance of the global Augmented Lagrangian algorithm. As expected, the global solutions were found (up to the prescribed tolerance) in all the problems. While using the piecewise convex α-underestimator reduces the number of nodes, it increases the CPU time when compared with using the convex α-underestimator.

presented in the Appendix, *all* the nine variables need to be branched. As a result, the direct application of Algorithm 2.1 to those problems takes several seconds of CPU time. On the other hand, the addition of new variables $w_{78} = x_7 x_8$ and $w_{79} = x_7 x_9$ reduces the number of variables that need to be branched from nine to five and the CPU time used by the method to a fraction of a second. Almost identical reasoning applies to Problem 2(d).

Finally, note that, as Problems 11 and 16 have just lower-level constraints, the Augmented Lagrangian framework is not activated at all and its resolution is automatically made through the direct application of the αBB method.

6 Final remarks

As already mentioned in [13], one of the advantages of the Augmented Lagrangian approach for solving nonlinear programming problems is its intrinsic adaptability to the global optimization problem. Namely, if one knows how to solve globally simple subproblems, the Augmented Lagrangian methodology allows one to globally solve the original constrained optimization problem. In this paper, we proved rigorously this fact, with an additional improvement of the Augmented Lagrangian method: the subproblems are redefined at each iteration using an auxiliary constraint set P_k that incorporates information obtained on the flight about the solution. Using the αBB algorithm for linearly constrained minimization subproblems, we showed that this approach is reliable. As a result, we have a practical Augmented Lagrangian method for constrained optimization that provably obtains ε-global minimizers. Moreover, the Augmented Lagrangian approach has a modular structure thanks to which one may easily replace subproblem global optimization solvers, which means that our method will be automatically improved as long as new global optimization solvers will be developed for the simple subproblems.

The challenge is improving efficiency. There are lots of unconstrained and constrained global optimization problems in Engineering, Physics, Chemistry, Economy, Computational Geometry and other areas that are not solvable with the present computer facilities. Our experiments seem to indicate that there is little to improve in the Augmented Lagrangian methodology, since the number of outer iterations is always moderate. The field for improvement is all concentrated in the global optimization of the subproblems. So, much research is expected in the following years in order to be able to efficiently solve more challenging practical problems.

The αBB global optimization algorithm is fully parallelizable in at least two levels: (i) subproblems with different subdomains can be solved in parallel; and (ii) the piecewise convex α-underestimator can be computed in parallel, reducing the number of nodes in the Branch & Bound algorithm without increasing the CPU time. Also the development of tighter underestimators for general nonconvex terms would be the subject of further research.

References

[1] C. S. Adjiman, I. P. Androulakis and C. A. Floudas, Global Optimization of MINLP Problems in Process Synthesis and Design, *Computers and Chemical Engineering* 21, pp. S445–S450, 1997.

[2] C. S. Adjiman, I. P. Androulakis and C. A. Floudas, Global Optimization of Mixed Integer Nonlinear Problems, *AIChE Journal* 46, pp. 1769–1797, 2000.

[3] 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.

[4] 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.

[5] 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.

[6] C. S. Adjiman and C. A. Floudas, Rigorous convex underestimators for general twice-differentiable problems, *Journal of Global Optimization* 9, pp. 23–40, 1996.

[7] A. Aggarwal and C. A. Floudas, Synthesis of General Separation Sequences – Nonsharp Separations, *Computers and Chemical Engineering* 14, pp. 631–653, 1990.

[8] I. G. Akrotirianakis and C. A. Floudas, A New Class of Improved Convex Underestimators for Twice Continuously Differentiable Constrained NLPs, *Journal of Global Optimization* 30, pp. 367–390, 2004.

[9] I. G. Akrotirianakis and C. A. Floudas, Computational Experience with a New Class of Convex Underestimators: Box Constrained NLP Problems, *Journal of Global Optimization* 29, pp. 249–264, 2004.

[10] 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, 1049–1057, 2000.

[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, vol. 125, pp. 473–485, 2005.

[12] 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*, to appear.

[13] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt, On Augmented Lagrangian methods with general lower-level constraints, Technical Report MCDO-050303, Department of Applied Mathematics, UNICAMP, Brazil, 2005 (Available in Optimization On Line and in www.ime.usp.br/∼egbirgin/).

[14] M. Andretta, *Large-scale linear constraint minimization*, Ph.D. Thesis, working project.

[15] 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.

[16] I. P. Androulakis, C. D. Maranas and C. A. Floudas, Prediction of Oligopeptide Conformations via Deterministic Global Optimization, *Journal of Global Optimization* 11, pp. 1–34, 1997.

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

[18] D. P. Bertsekas, *Nonlinear Programming*, 2nd edition, Athena Scientific, Belmont, Massachusetts, 1999.

[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 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.

[21] S. Caratzoulas and C. A. Floudas, Trigonometric Convex Underestimator for the Base Functions in Fourier Space, *Journal of Optimization Theory and Applications* 124, pp. 339–362, 2004.

[22] A. R. Ciric and C. A. Floudas, A Retrofit Approach of Heat Exchanger Networks, *Computers and Chemical Engineering* 13, pp. 703–715, 1989.

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

[24] W. R. Esposito and C. A. Floudas, Global Optimization in Parameter Estimation of Nonlinear Algebraic Models via the Error-In-Variables Approach, *Industrial and Engineering Chemistry Research* 37, pp. 1841–1858, 1998.

[25] W. R. Esposito and C. A. Floudas, Global Optimization for the Parameter Estimation of Differential-Algebraic Systems, *Industrial and Engineering Chemistry Research* 39, pp. 1291–1310, 2000.

[26] W. R. Esposito and C. A. Floudas, Deterministic Global Optimization in Nonlinear Optimal Control Problems, *Journal of Global Optimization* 17, pp. 97–126, 2000.

[27] W. R. Esposito and C. A. Floudas, Deterministic Global Optimization in Isothermal Reactor Network Synthesis, *Journal of Global Optimization* 22, pp. 59–95, 2002.

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

[29] 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.

[30] C. A. Floudas and S. H. Anastasiadis, Synthesis of General Distillation Sequences with Several Multicomponent Feeds and Products, *Chemical Engineering Science* 43, pp. 2407–2419, 1998.

[31] C. A. Floudas and I. E. Grossmann, Synthesis of Flexible Heat Exchanger Networks With Uncertain Flowrates and Temperatures, *Computers and Chemical Engineering* 11, pp. 319–336, 1987.

[32] 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.

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

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

[35] 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.

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

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

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

[39] S. T. Harding and C. A. Floudas, Phase Stability with Cubic Equations of State: A Global Optimization Approach, *AIChE Journal* 46, pp. 1422–1440, 2000.

[40] S. T. Harding and C. A. Floudas, Locating Heterogeneous and Reactive Azeotropes, *Industrial and Engineering Chemistry Research* 39, pp. 1576–1595, 2000.

[41] S. T. Harding, C. D. Maranas, C. M. McDonald, and C. A. Floudas, Locating All Homogeneous Azeotropes in Multicomponent Mixtures, *Industrial and Engineering Chemistry Research* 36, pp. 160–178, 1997.

[42] D. Hertz, C. S. Adjiman and C. A. Floudas, Two results on bounding the roots of interval polynomials, *Computers and Chemical Engineering* 23, pp. 1333–1339, 1999.

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

[44] 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.

[45] 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.

[46] J. L. Klepeis and C. A. Floudas, Free Energy Calculations for Peptides via Deterministic Global Optimization, *Journal of Chemical Physics* 110, pp. 7491–7512, 1999.

[47] J. L. Klepeis and C. A. Floudas, Ab Initio Tertiary Structure Prediction of Proteins, *Journal of Global Optimization* 25, pp. 113–140, 2003.

[48] J. L. Klepeis and C. A. Floudas, ASTRO-FOLD: a combinatorial and global optimization framework for ab initio prediction of three-dimensional structures of proteins from the amino acid sequence, *Biophysical Journal* 85, pp. 2119–2146, 2003.

[49] J. L. Klepeis, C. A. Floudas, D. Morikis and J. D. Lambris, Predicting Peptide Structures Using NMR Data and Deterministic Global Optimization, *Journal of Computational Chemistry* 20, pp. 1354–1370, 1999.

[50] J. L. Klepeis, M. Pieja and C. A. Floudas, A New Class of Hybrid Global Optimization Algorithms for Peptide Structure Prediction: Integrated Hybrids, *Computer Physics Communications* 151, pp. 121–140, 2003.

[51] J. L. Klepeis, M. Pieja and C. A. Floudas, A New Class of Hybrid Global Optimization Algorithms for Peptide Structure Prediction: Alternating Hybrids and Application fo Met-Enkephalin and Melittin, *Biophysical Journal* 84, pp. 869–882, 2003.

[52] A. C. Kokossis and C. A. Floudas, Optimization of Complex Reactor Networks-II: Nonisothermal Operation, *Chemical Engineering Science* 49, pp. 1037–1051, 1994.

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

[54] 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.

[55] J. Liebman, L. Lasdon, L. Schrage and A. Waren, *Modeling and Optimization with GINO*, The Scientific Press, Palo Alto, CA, 1986.

[56] M. Manousiouthakis and D. Sourlas, A global optimization approach to rationally constrained rational programming, *Chemical Engineering Communications* 115, pp. 127–147, 1992.

[57] C. D. Maranas and C. A. Floudas, A deterministic global optimization approach for molecular structure determination, *Journal of Chemical Physics* 100, pp. 1247–1261, 1994.

[58] C. D. Maranas and C. A. Floudas, Global minimum potencial energy conformations for small molecules, *Journal of Global Optimization* 4, pp. 135–170, 1994.

[59] C. D. Maranas and C. A. Floudas, Finding All Solutions of Nonlinearly Constrained Systems of Equations, *Journal of Global Optimization* 7, pp. 143–182, 1995.

[60] C. D. Maranas and C. A. Floudas, Global Optimization in Generalized Geometric Programming, *Computers and Chemical Engineering* 21, pp. 351–370, 1997.

[61] C. M. McDonald and C. A. Floudas, Global Optimization for the Phase Stability Problem, *AIChE Journal* 41, pp. 1798–1814, 1995.

[62] C. M. McDonald and C. A. Floudas, Global Optimization and Analysis for the Gibbs Free Energy Function for the UNIFAC, Wilson, and ASOG Equations, *Industrial and Engineering Chemistry Research* 34, pp. 1674–1687, 1995.

[63] C. A. Meyer and C. A. Floudas, Convex Hull of Trilinear Monomials with Positive or Negative Domains: Facets of the Convex and Concave Envelopes, in *Frontiers in Global Optimization*, C. A. Floudas and P. M. Pardalos (Eds.), Kluwer Academic Publishers, pp. 327–352, 2003.

[64] C. A. Meyer and C. A. Floudas, Convex Hull of Trilinear Monomials with Mixed Sign Domains, *Journal of Global Optimization* 29, pp. 125–155, 2004.

[65] C. A. Meyer and C. A. Floudas, Convex underestimation of twice continuously differentiable functions by piecewise quadratic perturbation: spline αBB underestimators, *Journal of Global Optimization* 32, pp. 221–258, 2005.

[66] C. A. Meyer and C. A. Floudas, Convex Envelopes for Edge-Concave Functions, *Mathematical Programming* 103, pp. 207–224, 2005.

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

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

[69] 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.

[70] 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.

[71] 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.

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

[73] H. S. Ryoo and N. V. Sahinidis, Global optimization of nonconvex NLPs and MINLPs with applications in process design, *Computers & Chemical Engineering* 19, pp. 551–566, 1995.

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

[75] N. V. Sahinidis, BARON: A general purpose global optimization software package, *Journal of Global Optimization* 8, pp. 201–205, 1996.

[76] N. V. Sahindis and I. E. Grossmann, Reformulation of the multi-period MILP model for capacity expansion of chemical processes, *Operations Research* 40, S127–S140, 1992.

[77] 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.

[78] 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, 197–228, 1995.

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

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

[81] 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, 101-112, 1992.

[82] 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.

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

[84] G. Stephanopoulos and A. W. Westerberg, The use of Hestenes' method of multipliers to resolve dual gaps in engineering system optimization, *Journal of Optimization Theory and Applications* 15, pp. 285–309, 1975.

[85] R. E. Swaney, Global solution of algebraic nonlinear programs, *AIChE Annual Meeting*, Chicago, IL, 1990.

[86] 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.

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

[88] 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.

[89] K. M. Westerberg and C. A. Floudas, Locating All Transition States and Studying Reaction Pathways of Potential Energy Surfaces, *Journal of Chemical Physics* 110, pp. 9259–9296, 1999.

[90] A. W. Westerberg and J. V. Shah, Assuring a global optimum by the use of an upper bound on the lower (dual) bound, *Computers & Chemical Engineering* 2, pp. 83–92, 1978.

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

7 Appendix

In this Appendix we describe the global optimization test problems considered in the numerical experiments.

Problem 1. [67]

$$\begin{array}{ll} \text{Minimize} & (x_1-1)^2+(x_1-x_2)^2+(x_2-x_3)^3+(x_3-x_4)^4+(x_4-x_5)^4 \\ \text{subject to} & x_1+x_2^2+x_3^3=3\sqrt{2}+2 \\ & x_2-x_3^2+x_4=2\sqrt{2}-2 \\ & x_1x_5=2 \\ & -5\le x_i\le 5,\ i=1,\ldots,5 \end{array}$$

Problem 2. *Haverly's pooling problem [5].*

$$\begin{array}{ll} \text{Minimize} & -9x_1-15x_2+6x_3+c_1x_4+10(x_5+x_6) \\ \text{subject to} & x_7x_8+2x_5-2.5x_1\le 0 \\ & x_7x_9+2x_6-1.5x_2\le 0 \\ & 3x_3+x_4-x_7(x_8+x_9)=0 \\ & x_8+x_9-x_3-x_4=0 \\ & x_1-x_8-x_5=0 \\ & x_2-x_9-x_6=0 \\ & (0,\ldots,0)\le x\le(c_2,200,500,\ldots,500) \end{array}$$

(a) $c_1 = 16$ and $c_2 = 100$; **(b)** $c_1 = 16$ and $c_2 = 600$; **(c)** $c_1 = 13$ and $c_2 = 100$.

Problem 2(d). A very similar version of the problem above but with an additional variable and different bounds [73].

$$\begin{array}{ll} \text{Minimize} & -9x_5 - 15x_9 + 6x_1 + 16x_2 + 10x_6 \\ \text{subject to} & x_{10}x_3 + 2x_7 - 2.5x_5 \leq 0 \\ & x_{10}x_4 + 2x_8 - 1.5x_9 \leq 0 \\ & 3x_1 + x_2 - x_{10}(x_3 + x_4) = 0 \\ & x_1 + x_2 - x_3 - x_4 = 0 \\ & x_3 + x_7 - x_5 = 0 \\ & x_4 + x_8 - x_9 = 0 \\ & x_7 + x_8 - x_6 = 0 \\ & (0, \ldots, 0, 1) \leq x \leq (300, 300, 100, 200, 100, 300, 100, 200, 200, 3) \end{array}$$

Problem 3. *Reactor network design [56].*

$$\begin{array}{ll} \text{Minimize} & -x_4 \\ \text{subject to} & x_1 + k_1 x_1 x_5 = 1 \\ & x_2 - x_1 + k_2 x_2 x_6 = 0 \\ & x_3 + x_1 + k_3 x_3 x_5 = 1 \\ & x_4 - x_3 + x_2 - x_1 + k_4 x_4 x_6 = 0 \\ & \sqrt{x_5} + \sqrt{x_6} \leq 4 \\ & (0, 0, 0, 0, 10^{-5}, 10^{-5}) \leq x \leq (1, 1, 1, 1, 16, 16) \end{array}$$

$k_1 = 9.755988\ 10^{-2}$; $k_2 = 0.99\ k_1$; $k_3 = 3.919080\ 10^{-2}$; $k_4 = 0.90\ k_3$.

Problem 3(b). A reformulation of Problem 3 eliminating some variables [28].

$$\begin{array}{ll} \text{Minimize} & -\left(\frac{k_1 x_1}{(1+k_1 x_1)(1+k_3 x_1)(1+k_4 x_2)} + \frac{k_2 x_2}{(1+k_1 x_1)(1+k_2 x_2)(1+k_4 x_2)} \right) \\ \text{subject to} & \sqrt{x_1} + \sqrt{x_2} \leq 4 \\ & (10^{-5}, 10^{-5}) \leq x \leq (16, 16) \end{array}$$

$k_1 = 9.755988\ 10^{-2}$; $k_2 = 0.99\ k_1$; $k_3 = 3.919080\ 10^{-2}$; $k_4 = 0.90\ k_3$.

Problem 4.

$$\begin{array}{ll} \text{Minimize} & x_2 \\ \text{subject to} & x_1 \cos(x_1) - x_2 \leq 0 \\ & (-10, -10) \leq x \leq (10, 10) \end{array}$$

Problem 5.

$$\begin{array}{ll} \text{Minimize} & \sum_{i=1}^{10} x_i \\ \text{subject to} & x_i^2 = 1,\ i = 1, \ldots, n \\ & -1 \leq x_i \leq 1,\ i = 1, \ldots, n \end{array}$$

Problem 6. [76]

$$\begin{array}{ll} \text{Minimize} & -x_1 - x_2 \\ \text{subject to} & x_1 x_2 \leq 4 \\ & (0,0) \leq x \leq (6,4) \end{array}$$

Problem 7. *Water pumping system [73].*

$$\begin{array}{ll} \text{Minimize} & x_3 \\ \text{subject to} & 30x_1 - 6x_1^2 - x_3 = -250 \\ & 20x_2 - 12x_2^2 - x_3 = -300 \\ & 0.5(x_1 + x_2)^2 - x_3 = -150 \\ & (0,0,0) \leq x \leq (9.422, 5.903, 267.42) \end{array}$$

Problem 8. *Design of a reinforced concrete beam [55].*

$$\begin{array}{ll} \text{Minimize} & 29.4x_1 + 18x_2 \\ \text{subject to} & -x_1 + 0.2458x_1^2/x_2 \leq -6 \\ & (0, 10^{-5}) \leq x \leq (115.8, 30) \end{array}$$

Problem 9. [88]

$$\begin{array}{ll} \text{Minimize} & x_1 + x_2 \\ \text{subject to} & x_1^2 + x_2^2 \leq 4 \\ & x_1^2 - x_2^2 \leq -1 \\ & x_1 - x_2 \leq 1 \\ & -x_1 + x_2 \leq 1 \\ & (-2,-2) \leq x \leq (2,2) \end{array}$$

Problem 10. [56]

$$\begin{array}{ll} \text{Minimize} & x_1^4 - 14x_1^2 + 24x_1 - x_2^2 \\ \text{subject to} & x_2 - x_1^2 - 2x_1 \leq -2 \\ & -x_1 + x_2 \leq 8 \\ & (-8,0) \leq x \leq (10,10) \end{array}$$

Problem 11. *Design of three-stage process system with recycle [84].*

$$\begin{array}{ll} \text{Minimize} & x_1^{0.6} + x_2^{0.6} + x_3^{0.4} - 4x_3 + 2x_4 + 5x_5 - x_6 \\ \text{subject to} & -3x_1 + x_2 - 3x_4 = 0 \\ & -2x_2 + x_3 - 2x_5 = 0 \\ & 4x_4 - x_6 = 0 \\ & x_1 + 2x_4 \leq 4 \\ & x_2 + x_5 \leq 4 \\ & x_3 + x_6 \leq 6 \\ & (10^{-5}, 10^{-5}, 10^{-5}, 0, 0, 0) \leq x \leq (3, 4, 4, 2, 2, 6) \end{array}$$

Problem 12. [85]

$$\begin{array}{ll} \text{Minimize} & 2x_1 + x_2 \\ \text{subject to} & -16x_1x_2 \le -1 \\ & -4x_1^2 - 4x_2^2 \le -1 \\ & (0,0) \le x \le (1,1) \end{array}$$

Problem 13. [85]

$$\begin{array}{ll} \text{Minimize} & -2x_1x_2 \\ \text{subject to} & 4x_1x_2 + 2x_1 + 2x_2 \le 3 \\ & (0,0) \le x \le (1,1) \end{array}$$

Problem 14. [84]

$$\begin{array}{ll} \text{Minimize} & -12x_1 - 7x_2 + x_2^2 \\ \text{subject to} & -2x_1^4 - x_2 = -2 \\ & (0,0) \le x \le (2,3) \end{array}$$

Problem 15. [90]

$$\begin{array}{ll} \text{Minimize} & 35x_1^{0.6} + 35x_2^{0.6} \\ \text{subject to} & 600x_1 - 50x_3 - x_1x_3 = -5000 \\ & 600x_2 + 50x_3 = 15000 \\ & (10^{-5}, 10^{-5}, 100) \le x \le (34, 17, 300) \end{array}$$

Problem 16. *Design of two-stage process system [84].*

$$\begin{array}{ll} \text{Minimize} & x_1^{0.6} + x_2^{0.6} - 6x_1 - 4x_3 + 3x_4 \\ \text{subject to} & -3x_1 + x_2 - 3x_3 = 0 \\ & x_1 + 2x_3 \le 4 \\ & x_2 + 2x_4 \le 4 \\ & (10^{-5}, 10^{-5}, 0, 0) \le x \le (3, 4, 2, 1) \end{array}$$

Problem 17. *Chemical equilibrium problem [73].*

$$\begin{array}{ll} \text{Minimize} & 0 \\ \text{subject to} & x_3^2/(x_1x_2^3) = 0.000169 \\ & x_2/x_1 = 3 \\ & x_1 + x_2 + x_3 = 50 \\ & (10^{-5}, 10^{-5}, 0) \le x \le (12.5, 37.5, 50) \end{array}$$

Problem 17(b). A trivial reformulation of Problem 17.

$$\begin{array}{ll} \text{Minimize} & 0 \\ \text{subject to} & x_3^2 - 0.000169x_1x_2^3 = 0 \\ & x_2 - 3x_1 = 0 \\ & x_1 + x_2 + x_3 = 50 \\ & (0,0,0) \le x \le (12.5, 37.5, 50) \end{array}$$

Problem 18. *Heat exchanger network design [55].*

$$
\begin{array}{ll}
\text{Minimize} & x_1 + x_2 + x_3 \\
\text{subject to} & (x_4 - 1) - 12x_1(3 - x_4) = 0 \\
& (x_5 - x_4) - 8x_2(4 - x_5) = 0 \\
& (5 - x_5) - 4x_3 = 0 \\
& (0, 0, 0, 1, 1) \leq x \leq (1.5834, 3.6250, 1, 3, 4)
\end{array}
$$