Journal of Optimization Theory and Applications (2026) 209:38
https://doi.org/10.1007/s10957-026-02981-9

An Augmented Lagrangian-Based Method Using Primitive Directions for Mixed-Integer Nonlinear Problems

Andrea Cristofari[1] · **Gianni Di Pillo**[2] · **Giampaolo Liuzzi**[2] · **Stefano Lucidi**[2]

Received: 30 July 2025 / Accepted: 11 March 2026

Abstract

In this paper, we consider mixed-integer nonlinear constrained optimization problems. Specifically, we assume that the integrality constraints are non-relaxable, that is, the functions appearing in the problem cannot be computed when the integrality constraints are violated. To solve this class of problems, we propose an augmented Lagrangian-type algorithm which is able to handle integer variables by means of primitive directions. A theoretical analysis of the convergence properties of the proposed algorithm is carried out. Finally, some numerical experimentation is reported.

Keywords Constrained optimization · MINLP · Augmented Lagrangian methods · Primitive directions

Mathematics Subject Classification 90C30

Communicated by Stefan M. Wild.

✉ Giampaolo Liuzzi
liuzzi@diag.uniroma1.it

Andrea Cristofari
andrea.cristofari@uniroma2.it

Gianni Di Pillo
dipillo@diag.uniroma1.it

Stefano Lucidi
lucidi@diag.uniroma1.it

1 Department of Civil Engineering and Computer Science Engineering, University of Rome "Tor Vergata", Via del Politecnico, 1, 00133 Rome, Italy

2 Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto, 25, 00185 Rome, Italy

Published online: 16 April 2026

1 Introduction

In this paper, we are concerned with combining an *augmented Lagrangian* approach [9], commonly used to solve continuous problems, with the use of particular directions, named *primitive directions*, which naturally allow us to handle integer variables. We also assume that integrality constraints are non-relaxable, i.e., they must always be satisfied in order to obtain meaningful results from the oracle computing problem functions and their partial derivatives. Hence, we limit ourselves to considering problems where integer variables are non-binary.

Then, we consider constrained optimization problems of the following form:

$$\begin{array}{ll} \min & f(x,z) \\ \text{s.t.} & h_i(x,z) = 0, \quad i = 1,\dots,p, \\ & x_i \in [l_i, u_i] \subset \mathbb{R}, \quad i = 1,\dots,n_x, \\ & z_i \in [a_i, b_i] \cap \mathbb{Z}, \quad i = 1,\dots,n_z, \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $h_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1,\dots,p$, $x \in \mathbb{R}^{n_x}$, $z \in \mathbb{Z}^{n_z}$, $n_x, n_z \geq 1$, $n = n_x + n_z$ and x_i, z_i denote the i-th component of vectors x and z. In the following, we denote by Ω, X and $\mathcal{Z}$ the sets:

$$\begin{aligned} \Omega &= \{(x,z) \in \mathbb{R}^{n_x} \times \mathbb{Z}^{n_z} : h_i(x,z) = 0,\ i = 1,\dots,p\}, \\ X &= \{x \in \mathbb{R}^{n_x} : l_i \leq x_i \leq u_i,\ i = 1,\dots,n_x\}, \\ \mathcal{Z} &= \{z \in \mathbb{Z}^{n_z} : a_i \leq z_i \leq b_i, i = 1,\dots,n_z\}, \end{aligned}$$

where $l_i, u_i \in \mathbb{R}$, $l_i < u_i$ for all $i = 1,\dots,n_x$, and $a_i, b_i \in \mathbb{Z}$, $a_i < b_i$ for all $i = 1,\dots,n_z$. Hence, the feasible set of Problem (1) is indicated by $\mathcal{F}$, that is,

$$\mathcal{F} = \Omega \cap (X \times \mathcal{Z}).$$

Note that, by definition, $X \times \mathcal{Z} \subset \mathbb{R}^n$ is a compact set (since so are both X and $\mathcal{Z}$). Then, $\mathcal{F}$ is compact as well.

We assume that $n \geq 2$ to simplify the description. The two particular cases where one among n_x and n_z is zero are covered by our theoretical analysis. In particular, if $n_z = 0$, the analysis boils down to the analysis reported in [16]. On the contrary, if $n_x = 0$, i.e., when all the variables are discrete, the MIX-ALM algorithm can still be used to solve integer constrained problems. In this particular case, MIX-ALM is different from the method proposed in [31] (which is where the use of primitive directions was proposed for integer nonlinear optimization problems). In particular, we note that the method proposed in [31] employs a different merit function to handle the constraints.

Note that the structure of Problem (1) is sufficiently general to capture, through reformulation by adding slack variables, also problems with inequality constraints. We require f and h_i, $i = 1,\dots,p$, to be (at least) twice continuously differentiable with respect to x in an open set containing the set $\{x \in \mathbb{R}^{n_x} : x_i \in [l_i, u_i],\ i = 1,\dots,n_x\} \times \{z \in \mathbb{Z}^{n_z} : z_i \in [a_i, b_i],\ i = 1,\dots,n_z\}$. This situation is very common in real world problems, especially in multidisciplinary optimization problems where some function values might be the output of a computer simulation that cannot be performed if some variables take on non-integer values (see, e.g., [23]). More in particular, given a black-box mixed integer constrained problem, we can come up with a problem that fits in our setting if we use a surrogate function (for which we can access first and second derivatives) to model the input-output relation of the objective function and the nonlinear constraints. Moreover, the proposed method could tackle the acquisition (infill) subproblem without relaxations within integer–continuous Bayesian optimization [11].

Mixed-integer problems have been widely studied for a long time and several methods have been proposed in the literature for their solution (see, e.g., [6, 12]). A detailed overview on the subject was given in the survey [7], illustrating both examples of application in different fields and methodological aspects. In particular, as described in [7], when the objective function is convex and only convex inequality constraints are considered, most algorithms for mixed-integer nonlinear programming use tree search schemes, including branch-and-bound and branch-and-cut methods, outer approximation and Benders decomposition. In a bit more detail, branch-and-bound methods [17, 27] divide the original problem into several subproblems which are iteratively created when the solution of a relaxation does not satisfy integrality constraints (branching phase), so that such subproblems are solved or discarded according to specific rules (pruning phase). Schemes of that form can be extended to include the generation of cutting planes in order to cut off fractional solutions, giving rise to branch-and-cut methods [34, 35]. Further classes of methods also exist which use outer approximation cuts [22] and Benders cuts [24], involving a linearization of the objective function and the constraints.

In a nonconvex setting, i.e., when non-convexity appears in the objective function and/or the constraints, solving the problem becomes even more challenging. Common approaches use a piecewise linear approximation of the nonconvex functions and apply tools from mixed-integer linear programming. However, most of these approaches are based on branch-and-bound or branch-and-cut techniques, assuming that integrality constraints can be relaxed to compute lower bounds on the optimal solution. In our setting, relaxing the integrality constraints is not applicable since, as already said above, the objective and/or constraint functions are not defined when some integer variables assume non-integer values. To the best of our knowledge, the approach we are proposing is the first attempt to address problems with these characteristics, i.e. with functions which are twice differentiable with respect to a group of variables and that present integer non-relaxable variables. More precisely, we integrate previous algorithms into a new method and establish novel convergence results.

To handle the continuous variables, among the various augmented Lagrangian schemes proposed in the literature (see, e.g., [2, 8]), here we consider a primal-dual augmented Lagrangian method using Newton-type directions and an active-set strategy, recently proposed by the authors in [16]. However, we note that optimization with respect to the continuous variables can be carried out using other approaches. Among the recent ones we cite the projected interior point method proposed in [25]; the block coordinate descent algorithm recently proposed in [10].

Our goal is to extend common approaches from continuous optimization in order to design an algorithm seeking local solutions.

The paper is organized as follows. Section 1 is ended by reporting some notations and definitions used in the paper. In Section 2, we describe the proposed algorithm to solve Problem (1). In Section 3, we analyze the convergence of the method. In Section 4, we report some computational examples on problems with non-relaxable integrality constraints. Finally, in Section 5, we draw some conclusions.

1.1 Notation

In the current paper, we adopt the following notation. Given a vector $v \in \mathbb{R}^n$, we denote by v_i its ith entry, $i = 1, \dots, n$. By v^k we denote the kth element of an infinite sequence $\{v^k\}$. Then, the ith component of the kth element of a sequence v^k is denoted as v_i^k. Given an index set $I \subseteq \{1, \dots, n\}$, we denote by v_I the subvector obtained from v by discarding the components not belonging to I. A vector $v \in \mathbb{R}^n$ is assumed to be a column vector so that its transpose, v^T, is a row vector. The Euclidean norm of a vector v is indicated by $\|v\|$, while $\|v\|_\infty$ denotes the sup-norm of v. Given a matrix M, we indicate by $\|M\|$ the matrix norm induced by the Euclidean vector norm. Given $a \in \mathbb{R}$, $\lfloor a \rceil$ denotes the integer nearest to a with smallest absolute value. The projection of a vector v onto a box $[l, u]$ is denoted by $\mathcal{P}_{[l,u]}(v)$. The ith column of the identity matrix is indicated by $e_i \in \mathbb{R}^n$, $i = 1, \dots, n$, while $e \in \mathbb{R}^n$ denotes the vector made of all unit entries. The gradient of $f(x, z)$ with respect to x is denoted by $\nabla_x f(x, z)$, while the Hessian matrix is denoted by $\nabla^2_{xx} f(x, z)$. We indicate by $\nabla_{x_i} f(x, z)$ the ith entry of $\nabla_x f(x, z)$.

1.2 Definitions

To define local solutions in a mixed integer framework, we have to specify what a neighborhood with respect to the integer variables is. To this aim, we use the notion of *primitive directions* [31].

Definition 1.1 *(Primitive direction)* A direction $d \in \mathbb{Z}^{n_z}$ is a primitive direction when the greatest common divisor of its components is 1 (note that the vector of all zeros is not primitive according to the definition).

For instance, when $n_z = 3$, $d^1 = (1,\ 0,\ -1)^\top$ is a primitive direction whereas $d^2 = (2,\ 0,\ -2)^\top$ is not. In particular, note that $d^2 = 2d^1$. Thus, an example of a set of primitive directions is the following

$$D = \left\{ \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ -2 \end{pmatrix} \right\}.$$

Then, we define the set of feasible primitive directions (with respect to the bound constraints on the integer variables, i.e., $\mathcal{Z}$) at a point $z \in \mathcal{Z}$, denoted by $D(z)$.

Definition 1.2 *(Set of feasible primitive directions)* Given $z \in \mathcal{Z}$,

$$D(z) := \{d \in \mathbb{Z}^{n_z} : d \text{ is primitive}, z + d \in \mathcal{Z}\}.$$

Remark 1.1 Since $\mathcal{Z}$ is a finite set, it follows that $D(z)$ is finite for all z. Then, the set of directions

$$\bar{D} := \bigcup_{z \in \mathcal{Z}} D(z)$$

is finite as well.

Definition 1.3 *(Neighborhoods)* Given $(x, z) \in \mathbb{R}^{n_x} \times \mathcal{Z}$ and $\rho > 0$,

$$\mathcal{B}(x; \rho) := \{y \in \mathbb{R}^{n_x} : \|y - x\| < \rho\},$$
$$\mathcal{B}^z(z) := \left\{w \in \mathbb{Z}^{n_z} : \exists\, d \in D(z) \text{ such that } w = z + d\right\} \cup \{z\}.$$

We refer to $\mathcal{B}(x; \rho)$ and $\mathcal{B}^z(z)$ as, respectively, open continuous neighborhood of x of radius ρ and integer neighborhood of z. Now we are ready to formally state the definitions of global and local solutions for Problem (1).

Definition 1.4 *(Global minimum point)* A point $(x^*, z^*) \in \mathcal{F}$ is a global solution for Problem (1), when

$$f(x^*, z^*) \le f(x, z), \quad \text{for all } (x, z) \in \mathcal{F}.$$

Definition 1.5 *(Local minimum point)* A point $(\bar{x}, \bar{z}) \in \mathcal{F}$ is a local solution for Problem (1), when a $\rho > 0$ exists such that

$$f(\bar{x}, \bar{z}) \le f(x, z) \quad \text{for all } (x, z) \in \mathcal{F} \cap (\mathcal{B}(\bar{x}; \rho) \times \mathcal{B}^z(\bar{z})).$$

In order to give necessary optimality conditions with respect to the continuous variables, we introduce a suitable Lagrangian function which depends only on the objective function and the equality constraint functions.

Definition 1.6 *(Lagrangian function)* The Lagrangian function $L(x, z, \mu)$ for Problem (1) with respect to the equality constraints is defined as follows:

$$L(x, z, \mu) := f(x, z) + \mu^T h(x, z),$$

where $\mu \in \mathbb{R}^p$ is the Lagrange multiplier and $h : \mathbb{R}^n \to \mathbb{R}^p$ is the mapping such that $h(x, z) = (h_1(x, z), \ldots, h_p(x, z))^\top$.

From the continuous optimization context, we also recall the following definition of a KKT point with respect to the continuous variables x.

Definition 1.7 *(KKT point with respect to x)* We say that (x, z) is a KKT point for Problem (1) with respect to x if there exists $(\mu, \sigma, \rho) \in \mathbb{R}^{p+2n_x}$ such that

$$\begin{aligned} \nabla_x L(x, z, \mu) &= \sigma - \rho, \\ h(x, z) &= 0, \\ \sigma^T (l - x) &= 0, \\ \rho^T (x - u) &= 0, \\ l - x &\leq 0, \quad \sigma \geq 0, \\ x - u &\leq 0, \quad \rho \geq 0, \end{aligned} \tag{2}$$

where $\nabla_x L(x, z, \mu) = \nabla_x f(x, z) + \nabla_x h(x, z)\mu$ is the gradient of $L(x, z, \mu)$ with respect to x.

We note that the KKT conditions (2) can be rewritten as follows:

$$\begin{aligned} \nabla_{x_i} L(x, z, \mu) &\begin{cases} = 0, & \text{if } l_i < x_i < u_i, \\ \geq 0, & \text{if } x_i = l_i, \\ \leq 0, & \text{if } x_i = u_i, \end{cases} \\ h(x, z) &= 0. \end{aligned}$$

2 The Mixed-integer Augmented Lagrangian Method (MIX-ALM)

In this section, we describe the proposed Mixed-integer Augmented Lagrangian Method (MIX-ALM) to solve Problem (1), which is reported in Algorithm 1. The proposed approach combines a primal-dual augmented Lagrangian scheme, employing an active-set strategy, with the use of primitive directions. So, each iteration relies on the following main steps:

1. updating of the continuous variables, i.e., definition of x^{k+1};
2. updating of the integer variables, i.e., definition of z^{k+1};
3. updating of the penalty parameter, i.e., definition of ϵ^{k+1};
4. computation of new multiplier estimates, i.e., definition of $\bar{\mu}^{k+1}$.

In the following, we describe the above phases in more detail.

2.1 Updating of the Continuous Variables

At the beginning of any iteration k of the algorithm, we are given a tuple $(x^k, z^k, \bar{\mu}^k)$ and first try to compute (x^{k+1}, μ^{k+1}). This is done by using the same strategy recently proposed in [16] (named PD-ALM), combining an active-set strategy with a Newton scheme and augmented Lagrangian functions. For the sake of completeness, we briefly report here the main steps of this process.

Let us first define, for the current iteration k, the following augmented Lagrangian function for Problem (1) with respect to the equality constraints:

$$L_a(x^k, z^k, \bar{\mu}^k; \epsilon^k) := L(x^k, z^k, \bar{\mu}^k) + \frac{1}{\epsilon^k}\|h(x^k, z^k)\|^2,$$

where $\epsilon^k > 0$ is a penalty parameter and L is the Lagrangian function (see Definition 1.6).

Then, we carry out an active-set estimate to identify those variables that are likely to lie at the lower or the upper bound in the final solution, using appropriate multiplier functions described in [15, 18, 20]. In particular, we denote by $\mathcal{L}^k$ and $\mathcal{U}^k$ the sets of variables estimated to be active at the lower and the upper bound, respectively defined as

$$\begin{aligned} \mathcal{L}^k &:= \{i : \nabla_{x_i} L(x^k, z^k, \bar{\mu}^k) > 0,\ l_i \le x_i^k \le l_i + \nu \cdot \sigma_i^k\}, \\ \mathcal{U}^k &:= \{i : \nabla_{x_i} L(x^k, z^k, \bar{\mu}^k) < 0,\ u_i - \nu \cdot \rho_i^k \le x_i^k \le u_i\}, \end{aligned} \tag{3}$$

where $\nu > 0$ and

$$\begin{aligned} \sigma_i^k &:= \frac{(u_i - x_i^k)^2}{(l_i - x_i^k)^2 + (u_i - x_i^k)^2} \nabla_{x_i} L(x^k, z^k, \bar{\mu}^k), \quad i = 1, \dots, n_x, \\ \rho_i^k &:= -\frac{(l_i - x_i^k)^2}{(l_i - x_i^k)^2 + (u_i - x_i^k)^2} \nabla_{x_i} L(x^k, z^k, \bar{\mu}^k), \quad i = 1, \dots, n_x. \end{aligned}$$

Then, denoting $h^k := h(x^k, z^k)$, $L^k := L(x^k, z^k, \bar{\mu}^k)$, $\nabla_x h^k := \nabla_x h(x^k, z^k, \bar{\mu}^k)$, $\nabla_x L^k := \nabla_x L(x^k, z^k, \bar{\mu}^k)$, $\nabla^2_{xx} L^k := \nabla^2_{xx} L(x^k, z^k, \bar{\mu}^k)$, $\mathcal{A}^k := \mathcal{L}^k \cup \mathcal{U}^k$ and $\mathcal{N}^k := \{1, \dots, n_x\} \setminus \mathcal{A}^k$, we consider the following Newton system for the KKT conditions with respect to $d_{x_{\mathcal{N}^k}}$ and d_μ:

$$\begin{pmatrix} \nabla^2_{\mathcal{N}^k} L^k & \nabla_{\mathcal{N}^k} h^k \\ \nabla_{\mathcal{N}^k} (h^k)^T & 0 \end{pmatrix} \begin{pmatrix} d_{x_{\mathcal{N}^k}} \\ d_\mu \end{pmatrix} = - \begin{pmatrix} \nabla_{\mathcal{N}^k} L^k \\ h^k \end{pmatrix}, \tag{4}$$

where $\nabla^2_{\mathcal{N}^k} L^k$ denotes the submatrix obtained from $\nabla^2_{xx} L^k$ by discarding rows and columns not belonging to $\mathcal{N}^k$, and $\nabla_{\mathcal{N}^k} h^k$ (respectively, $\nabla_{\mathcal{N}^k} L^k$) denotes the submatrix obtained from $\nabla_x h^k$ (respectively, $\nabla_x L^k$) discarding rows not belonging to $\mathcal{N}^k$. Hence, we compute a solution $d^k = (d_{x_{\mathcal{N}^k}}, d_\mu)$ of system (4), provided it exists, in order to define

$$\tilde{x}^k_{\mathcal{N}^k} = \mathcal{P}_{[l_{\mathcal{N}^k}, u_{\mathcal{N}^k}]}(x^k_{\mathcal{N}^k} + d_{x_{\mathcal{N}^k}}). \tag{5}$$

Moreover, the variables estimated as active, i.e., those variables that should be at the bounds, are set to the bounds by defining

$$\tilde{x}_i^k = \begin{cases} l_i, & \text{if } i \in \mathcal{L}^k, \\ u_i, & \text{if } i \in \mathcal{U}^k. \end{cases} \tag{6}$$

Then, we check whether

$$L_a(\tilde{x}^k, z^k, \bar{\mu}^k; \epsilon^k) \le L_a(x^k, z^k, \bar{\mu}^k; \epsilon^k) \tag{7}$$

and

$$\|(d^k, (\tilde{x}^k - x^k)_{\mathcal{A}^k})\| \le \Delta^k, \tag{8}$$

where d^k is the solution of system (4) and $\tilde{x}_i^k$ for $i \in \mathcal{A}^k$ are given by (6), while Δ^k is decreased during the iterations by a factor $\beta \in (0, 1)$. If (7)–(8) hold, then we set $\Delta^{k+1} = \beta\Delta^k$, $x^{k+1} = \tilde{x}^k$ and

$$\mu^{k+1} = \bar{\mu}^k + d_\mu.$$

Otherwise (i.e., if a solution of system (4) does not exist or if at least one between (7) and (8) is not satisfied), we switch to a classical augmented Lagrangian iteration [2], using ALGENCAN [3] to compute x^{k+1} as an approximate minimizer of the following subproblem:

$$\begin{aligned} \min\ & L_a(x, z^k, \bar{\mu}^k; \epsilon^k) \\ & l \le x \le u. \end{aligned}$$

In fact, it is known that the projected Newton direction can be a very bad search direction, not even guaranteeing descent of the objective function. So, conditions (7)–(8) can be considered a safeguard which allows us to trigger those cases where the projected Newton direction is behaving poorly. In these situations, backing up to the traditional augmented Lagrangian could be beneficial as it is also evidenced in [14].

In particular, we require that

$$L_a(x^{k+1}, z^k, \bar{\mu}^k; \epsilon^k) \le L_a(x^k, z^k, \bar{\mu}^k; \epsilon^k) \tag{9}$$

and

$$\|x^{k+1} - \mathcal{P}_{[l,u]}(x^{k+1} - \nabla_x L_a(x^{k+1}, z^k, \bar{\mu}^k; \epsilon^k))\|_\infty \le \tau^k, \tag{10}$$

with $\{\tau^k\} \to 0$. Finally, we set

$$\mu^{k+1} = \bar{\mu}^k + \frac{2}{\epsilon^k} h(x^{k+1}, z^k),$$

2.2 Updating of the Discrete Variables

Once x^{k+1} has been computed, we try to compute z^{k+1} by moving along primitive directions (see Definition 1.2). As pointed out in [31], where primitive directions have been introduced in the context of nonlinear black-box optimization, given two integer vectors $a, b \in \mathbb{Z}^{n_z}$, $a \ne b$, then $a - b = \alpha d$ with d being a primitive direction and $\alpha \in \mathbb{Z}$. In the algorithm, given the current point (x^{k+1}, z^k), we search for a direction $d \in D(z^k)$ such that

$$L_a(x^{k+1}, z^k + d, \bar{\mu}^k; \epsilon^k) \le L_a(x^{k+1}, z^k, \bar{\mu}^k; \epsilon^k) - \frac{\xi^k}{\epsilon^k}, \tag{11}$$

where $\xi^k > 0$ is a control parameter. Thus, the required reduction is directly proportional to the control parameter ξ^k and inversely proportional to the penalty parameter ϵ^k. This is equivalent to requiring a sufficient reduction with respect to ξ^k of the augmented Lagrangian function times ϵ^k.

If a primitive direction d satisfying (11) is found, then we set $z^{k+1} = z^k + d$ and $\xi^{k+1} = \xi^k$, otherwise we set $z^{k+1} = z^k$ and $\xi^{k+1} = \beta\xi^k$, with $\beta \in (0, 1)$.

2.3 Updating of Penalty Parameter and Multiplier Estimates

To update the penalty parameter ϵ^k, given a parameter $\eta \in (0, 1)$ (see Algorithm 1), we check whether

$$\|h(x^{k+1}, z^{k+1})\|_\infty \le \eta \|h(x^k, z^k)\|_\infty \tag{12}$$

and

$$z^{k+1} = z^k. \tag{13}$$

If (12)–(13) are satisfied, then we set $\epsilon^{k+1} = \epsilon^k$, otherwise we set $\epsilon^{k+1} = \theta\epsilon^k$, with $\theta \in (0, 1)$.

As last operation of any iteration, we project the estimates of the Lagrange multipliers onto a prefixed box. Namely, we set

$$(\bar{\mu}^{k+1})_i = \max\{\bar{\mu}_{\min}, \min\{(\mu^{k+1})_i, \bar{\mu}_{\max}\}\}, \quad i = 1, \dots, p, \tag{14}$$

with given constants $\bar{\mu}_{\min}$ and $\bar{\mu}_{\max}$.

Combining all the aspects herein described, we formalize the description of the MIX-ALM method in Algorithm 1.

3 Convergence Results

In this section we report the convergence analysis of MIX-ALM. In order to extend the results given in [16] to the considered mixed-integer setting, our analysis can be roughly divided into two parts: first, we obtain results for any convergent subsequence and prove some optimality conditions with respect to the continuous variables; then, we give results concerning the optimality with respect to the integer variables.

Before going into details, we state a fundamental assumption which allows us to get stationarity results of augmented Lagrangian-type algorithms. In particular, some constraint qualification conditions are usually required. Among those proposed in the literature, here we use the Constant Positive Linear Dependence (CPLD) constraint qualification (see [3, 4, 33] for more details), which is given below.

Definition 3.1 *(CPLD)* Let $\mathcal{L}(x) := \{i : x_i = l_i\}$, $\mathcal{U}(x) := \{i : x_i = u_i\}$ and $\mathcal{N}(x) := \{1, \dots, n_x\} \setminus (\mathcal{L}(x) \cup \mathcal{U}(x))$. A feasible point (x, z) is said to satisfy CPLD

Algorithm 1 Mixed-integer Augmented Lagrangian Method (MIX-ALM)

Given scalars $\bar{\mu}_{\min} < \bar{\mu}_{\max}$, $\beta \in (0, 1)$, $\eta \in (0, 1)$, $\theta \in (0, 1)$, $\Delta_0 > 0$, $\epsilon_0 > 0$, $\xi_0 > 0$, a sequence $\{\tau^k\} \searrow 0$, a starting point $x_0 \in [l, u]$, $z_0 \in [a, b] \cap \mathbb{Z}^{n_z}$ and estimates of multipliers $(\bar{\mu}_0)_i = (\mu_0)_i \in [\bar{\mu}_{\min}, \bar{\mu}_{\max}]$, $i = 1, \dots, p$
For $k = 0, 1, \dots$
• *Updating of the continuous variables*
compute the active set estimates $\mathcal{L}^k, \mathcal{U}^k$ as in (3)
set $\mathcal{A}^k = \mathcal{L}^k \cup \mathcal{U}^k$ and $\mathcal{N}^k = \{1, \dots, n_x\} \setminus \mathcal{A}^k$
compute $d^k = (d_{x_\mathcal{N}}, d_\mu)$ by solving (4), if possible, and set $\tilde{x}^k$ as in (5)–(6)
If d^k has been computed and (7)–(8) are satisfied
set $x^{k+1} = \tilde{x}^k$, $\mu^{k+1} = \bar{\mu}^k + d_\mu$ and $\Delta^{k+1} = \beta \Delta^k$
Else
compute x^{k+1} satisfying (9)-(10)
Set $\mu^{k+1} = \bar{\mu}^k + \dfrac{2}{\epsilon^k} h(x^{k+1}, z^k)$ and $\Delta^{k+1} = \Delta^k$
End If
• *Updating of the integer variables*
If $d \in D(z^k)$ exists such that (11) is satisfied
set $z^{k+1} = z^k + d$ and $\xi^{k+1} = \xi^k$
Else
set $z^{k+1} = z^k$ and $\xi^{k+1} = \beta \xi^k$
End If
• *Updating of the penalty parameter*
If (12)–(13) are satisfied
set $\epsilon^{k+1} = \epsilon^k$
Else
set $\epsilon^{k+1} = \theta \epsilon^k$
End If
• *Updating of the multiplier estimates*
set $\bar{\mu}^{k+1}$ as in (14)
End For

for Problem (1) if, for any $\bar{I} \subseteq \{1, \dots, p\}$, $\bar{\mathcal{L}} \subseteq \mathcal{L}(x)$ and $\bar{\mathcal{U}} \subseteq \mathcal{U}(x)$, the existence of scalars λ_t, $t \in \bar{I}$, $\pi_i \geq 0$, $i \in \bar{\mathcal{L}}$, $\varphi_j \geq 0$, $j \in \bar{\mathcal{U}}$, not all zero and such that $\sum_{t \in \bar{I}} \lambda_t \nabla_x h_t(x, z) - \sum_{i \in \bar{\mathcal{L}}} \pi_i e_i + \sum_{j \in \bar{\mathcal{U}}} \varphi_j e_j = 0$ implies that, for all y in a neighborhood of x, the vectors $\nabla_x h_t(y, z)$, $t \in \bar{I}$, $-e_i$, $i \in \bar{\mathcal{L}}$, e_j, $j \in \bar{\mathcal{U}}$ are linearly dependent.

The CPLD constraint qualification has been introduced in [33]. It is weaker than other well known constraint qualifications like e.g. the Linear Independence Constraint Qualification (LICQ) and the Mangasarian-Fromowitz Constraint Qualification (MFCQ). For example, CPLD is always satisfied in the case of linear constraints whereas LICQ might not be.

We start the analysis of Algorithm MIX-ALM by showing that for some subsequence of the sequence of iterates the results of PD-ALM are retained. In particular, we consider convergent subsequences. Hence, on these subsequences, the integer variables are not updated by MIX-ALM from a certain iteration on. In such a case, MIX-ALM reduces (from a certain iteration) to PD-ALM described in [16].

In the next theorem, we prove that at least a convergent subsequence of iterates exists. Furthermore, any convergent subsequence enjoys some optimality property. In particular, we give results on the feasibility of any limit point (x^*, z^*) and its optimality violation with respect to the continuous variables. In particular, note that (x^*, z^*) is KKT with respect to x whenever the Newton direction is accepted for infinitely many iterations over the considered subsequence or when it is feasible and the CPLD constraint qualification holds.

Theorem 3.1 *Let $\{(x^k, z^k)\}$ be a sequence generated by MIX-ALM. There exists a subsequence $\{(x^k, z^k)\}_K$ such that*

$$\lim_{k\to\infty,\, k\in K} x^{k+1} = x^*, \qquad \lim_{k\to\infty,\, k\in K} z^k = z^*. \tag{15}$$

For any convergent subsequence $\{(x^k, z^k)\}_K$ such that (15) holds, the following are satisfied:

(a) if $\lim_{k\to\infty} \epsilon^k > 0$, then (x^*, z^*) is feasible;
(b) if $\tilde{x}^k$ is accepted (i.e., d^k is computed and $\|(d^k, (\tilde{x}^k - x^k)_{\mathcal{A}^k})\| \le \Delta^k$) for infinitely many iterations $k \in K$, then (x^*, z^*) is a KKT point with respect to x;
(c) if none of the above cases holds, then x^* is a KKT point of the problem $\min_{x\in X} \|h(x, z^*)\|^2$;
(d) if (x^*, z^*) is feasible and satisfies the CPLD constraint qualification, then (x^*, z^*) is a KKT point with respect to x.

Proof By definition of Algorithm MIX-ALM, the points x^k and z^k belong to X and $\mathcal{Z}$, respectively, which are compact sets. Then, at least an index set K exists such that (15) is satisfied.

Since $\{z^k\}$ is a sequence of integer vectors, it follows that there is a large enough $\bar{k}$ such that $z^k = z^*$ for all $k \in K$, $k \ge \bar{k}$ (i.e., $z^k = z^*$ definitely on the subsequence K). So, points (a) and (c) follow from [16, Proposition 4.2], point (b) follows from [16, Proposition 4.1] and point (d) follows from [16, Theorem 4.1]. □

Now, we prove stronger properties for the integer variables than those given in Theorem 3.1. First, we can identify an appropriate subsequence where MIX-ALM fails to update the integer variables for infinitely many iterations. As to be shown below, this can be achieved by analyzing the sequence $\{\xi^k\}$ of control parameters, which is proved to converge to zero in the next proposition.

Proposition 3.1 *The sequence $\{\xi^k\}$ of control parameters produced by MIX-ALM is such that*

$$\lim_{k\to\infty} \xi^k = 0.$$

Proof From the updating rule of ξ^k, it holds that $\{\xi^k\}$ is a monotone non-increasing sequence of positive numbers. Hence,

$$\lim_{k\to\infty} \xi^k = \bar{\xi} \ge 0.$$

Let us assume by contradiction that $\bar{\xi} > 0$. Then, an iteration index $\bar{k}$ exists such that, for all $k \geq \bar{k}$, we have

$$\xi^{k+1} = \xi^k = \xi_{\bar{k}} = \bar{\xi},$$

since the updating rule of ξ requires a decrease by a fixed fraction β. By the instructions at line 13 of MIX-ALM, we have that for $k \geq \bar{k}$, it holds $z^{k+1} \neq z^k$ so that (13) is not satisfied and $\epsilon^{k+1} = \theta\epsilon^k$ (see line 20 in Algorithm 1). Hence,

$$\lim_{k \to \infty} \epsilon^k = 0.$$

From the updating of the integer variables at lines 12-13, we have that $d \in D(z^k)$ satisfying (11) exists such that

$$\begin{aligned} L_a(x^{k+1}, z^{k+1}, \bar{\mu}^k; \epsilon^k) &= L_a(x^{k+1}, z^k + d, \bar{\mu}^k; \epsilon^k) \\ &\leq L_a(x^{k+1}, z^k, \bar{\mu}^k; \epsilon^k) - \bar{\xi}/\epsilon^k. \end{aligned} \tag{16}$$

Furthermore, from the instructions of the algorithm, either $x^{k+1} = \tilde{x}^k$, with $\tilde{x}^k$ satisfying (7)–(8), or x^{k+1} is such that (9)–(10) hold. Thus, in both cases, we have

$$L_a(x^{k+1}, z^k, \bar{\mu}^k, \epsilon^k) \leq L_a(x^k, z^k, \bar{\mu}^k, \epsilon^k).$$

Then, by (16), we have

$$L_a(x^{k+1}, z^{k+1}, \bar{\mu}^k, \epsilon^k) \leq L_a(x^k, z^k, \bar{\mu}^k, \epsilon^k) - \bar{\xi}/\epsilon^k. \tag{17}$$

By multiplying all terms by ϵ^k, we get

$$\begin{aligned} \|h(x^{k+1}, z^{k+1})\|^2 - \|h(x^k, z^k)\|^2 \leq \epsilon^k \big(f(x^k, z^k) - f(x^{k+1}, z^{k+1}) + \\ + (\bar{\mu}^k)^\top h(x^k, z^k) - (\bar{\mu}^k)^\top h(x^{k+1}, z^{k+1}) \big) - \bar{\xi}. \end{aligned}$$

Hence

$$\begin{aligned} \|h(x^{k+1}, z^{k+1})\|^2 &\leq \|h(x^k, z^k)\|^2 \\ &\quad + \epsilon^k | f(x^k, z^k) - f(x^{k+1}, z^{k+1}) + (\bar{\mu}^k)^\top h(x^k, z^k) + \\ &\quad - (\bar{\mu}^k)^\top h(x^{k+1}, z^{k+1}) | - \bar{\xi} \\ &\leq \|h(x^k, z^k)\|^2 + \epsilon^k \bar{\delta} - \bar{\xi}, \end{aligned} \tag{18}$$

where

$$\bar{\delta} = |f_{\max} - f_{\min}| + p|\mu_{\max}| \max_{i=1,\ldots,p} |h_{i,\max} - h_{i,\min}|,$$

and $f_{\max}, h_{i,\max}, i = 1, \ldots, p$, $(f_{\min}, h_{i,\min}, i = 1, \ldots, p)$ are maximum (minimum) values of f and h_i, $i = 1, \ldots, p$, respectively, on the compact set $X \times \mathcal{Z}$, and $\mu_{\max} = \max\{|\bar{\mu}_{\min}|, |\bar{\mu}_{\max}|\}$.

Since $\{\epsilon^k\} \to 0$ then, for sufficiently large k, we have that

$$\|h(x^{k+1}, z^{k+1})\|^2 \le \|h(x^k, z^k)\|^2 - \bar{\xi}/2, \tag{19}$$

leading to a contradiction with the non-negativity of the norm. □

Remark 3.1 In the proof of Proposition 3.1, we use $\lim_{k\to\infty} \epsilon^k = 0$. It is worth noting that the fact that the sequence of penalty parameters $\{\epsilon^k\}$ goes to zero is a consequence of the contradiction hypothesis, namely that $\lim_{k\to\infty} \xi^k = \bar{\xi} > 0$. Indeed, as it has been shown in Theorem 3.1, the penalty parameter need not tend to zero for the algorithm to have theoretical properties.

Corollary 3.1 *For MIX-ALM, there exists an infinite set H of iterations k such that $z^{k+1} = z^k$ for all $k \in H$.*

Proof Note that ξ^k is updated only when $k \in H$. Since $\{\xi^k\}$ is a monotone non-increasing sequence of positive numbers, which converges to zero (Proposition 3.1), then H must be infinite. □

Leveraging Corollary 3.1, in the next theorem we can establish some optimality properties for the integer variables when $\{\epsilon^k\}$ does not converge to zero. Furthermore, if $\{\epsilon^k\} \to 0$, in (x^*, z^*) we have stationarity over the continuous variables for $\|h(x, z^*)\|^2$ and local optimality over the integer variables for $\|h(x^*, z)\|^2$.

Theorem 3.2 *Let $\{(x^k, z^k)\}$ be a sequence generated by MIX-ALM. There exists a subsequence $\{(x^k, z^k)\}_K$ with $K \subseteq H$ (where H is defined in Corollary 3.1) such that*

$$\lim_{k\to\infty,\, k\in K} x^{k+1} = x^*, \qquad \lim_{k\to\infty,\, k\in K} z^k = z^*. \tag{20}$$

For all such limit points (x^*, z^*), the following holds:

(a) if $\lim_{k\to\infty} \epsilon^k > 0$, then (x^*, z^*) is feasible and z^* is a local minimum point of $f(x^*, z)$ on the set $\{z \in \mathcal{Z} : (x^*, z) \in \mathcal{F}\}$
(b) if $\lim_{k\to\infty} \epsilon^k = 0$ and $\tilde{x}^k$ is not accepted for infinitely many iterations in K, then

- *x^* is a KKT point for the problem $\min_{x\in X} \|h(x, z^*)\|^2$;*
- *z^* is a local minimum point for the problem $\min_{z\in\mathcal{Z}} \|h(x^*, z)\|^2$.*

Proof First let us recall that, by Corollary 3.1, H is an infinite index set and again that the points x^k and z^k belong to X and $\mathcal{Z}$, respectively, which are compact sets. Then, at least a subset $K \subseteq H$ exists such that (20) is satisfied.

Let us first consider point (a). If $\lim_{k\to\infty} \epsilon^k > 0$, by Theorem 3.1(a) we know that (x^*, z^*) is feasible. Since $\{\epsilon^k\}$ is a monotone non-increasing sequence of positive numbers, there exist $\bar{k}$ and $\bar{\epsilon} > 0$ such that, for all $k \ge \bar{k}$, we have

$$\epsilon^k = \bar{\epsilon}.$$

Recalling that $\{z^k\}_K \to z^*$ for $k \to \infty$ and that $K \subseteq H$ in this specific case, from the integrality of z we can assume that $z^{k+1} = z^k = z^*$ for all $k \in K$ (passing into a further subsequence if necessary). Then, since the integer variables are never updated by the algorithm at iteration $k \in K$, for sufficiently large $k \in K$ we have

$$L_a(x^{k+1}, z^* + d, \bar{\mu}^k; \bar{\epsilon}) > L_a(x^{k+1}, z^*, \bar{\mu}^k; \bar{\epsilon}) - \frac{\xi^k}{\bar{\epsilon}} \quad \forall d \in D(z^*). \tag{21}$$

Then, by using that $\{x^{k+1}\}_K \to x^*$, $\{\xi^k\} \to 0$ and $(x^*, z^*) \in \mathcal{F}$, we get

$$f(x^*, z^*+d) + \sum_{i=1}^{p} \hat{\mu}|h_i(x^*, z^*+d)| + \frac{1}{\bar{\epsilon}}\|h(x^*, z^*+d)\|^2 \geq f(x^*, z^*) \;\; \forall d \in D(z^*),$$

where $\hat{\mu} = \max\{|\bar{\mu}_{\min}|, |\bar{\mu}_{\max}|\}$. From the above relation, we can conclude

$$f(x^*, z^*) \leq f(x^*, z^* + d)$$

for all $d \in D(z^*)$ such that $(x^*, z^* + d) \in \mathcal{F}$, which proves point (c).

Let us now consider point (b). If $\lim_{k\to\infty} \epsilon^k = 0$, by Theorem 3.1(c) we already have that x^* is a KKT point for the problem $\min_{x\in X} \|h(x, z^*)\|^2$. We want to prove that

$$\|h(x^*, z^*)\|^2 \leq \|h(x^*, z)\|^2, \quad \forall\, z \in \mathcal{B}^z(z^*).$$

Recalling that $\{z^k\}_K \to z^*$ for $k \to \infty$ and that $K \subseteq H$ in this specific case, again we assume that $z^{k+1} = z^k = z^*$ for all $k \in K$ (passing into a further subsequence if necessary). Since the integer variables are never updated by the algorithm at iteration $k \in K$, for sufficiently large $k \in K$ we have

$$L_a(x^{k+1}, z^* + d, \bar{\mu}^k; \epsilon^k) > L_a(x^{k+1}, z^*, \bar{\mu}^k; \epsilon^k) - \frac{\xi^k}{\epsilon^k} \quad \forall d \in D(z^*). \tag{22}$$

Then, recalling that $\bar{\mu}^k$ is bounded, $\{x^{k+1}\}_K \to x^*$ and $\{\xi^k\} \to 0$, we can multiply (22) by ϵ^k and take the limit for $k \to \infty$, thus obtaining

$$\|h(x^*, z^* + d)\|^2 \geq \|h(x^*, z^*)\|^2 \quad \forall d \in D(z^*),$$

concluding the proof. □

Remark 3.2 Concerning point (c) of Theorem 3.1 and point (b) of the Theorem 3.2, they analyze the situation in which $\epsilon^k \to 0$. This can happen, for instance, when the problem is infeasible. In these cases, the given properties are very important since they guarantee that the limit point produced by the algorithm is "optimal" (stationary) with respect to the measure of feasibility violation. That is to say that the limit point is as close to feasibility as possible.

Remark 3.3 A quadratic convergence rate for MIX-ALM can be straightforwardly obtained when $\{z^k\} \to z^*$ since, in this case, MIX-ALM reduces to [16, Algorithm 2 (PD-ALM)] and the same analysis of [16, Section 5] can be given. In particular, under standard conditions on the limit (x^*, z^*) (i.e., linear independence constraint qualification and strong second order sufficient condition), $\tilde{x}^k$ is always accepted if the starting point x_0 is sufficiently close to x^*.

Let us conclude this section by giving some comments on the usefulness of Theorems 3.1 and 3.2. Although it is quite evident that the results of Theorem 3.1 follow from results in [16], the results of Theorem 3.2 can be useful especially when linked with point (b) and (d) of Theorem 3.1. In more detail, if the sequence $\{\epsilon^k\}$ of penalty parameters is bounded away from zero, we know from Theorem 3.2(a) that the limit point (x^*, z^*) is feasible and that z^* is a local minimum of $f(x^*, z)$. Moreover, if (x^*, z^*) satisfies the CPLD constraint qualification, by Theorem 3.1(d) we also know that x^* is a KKT point with respect to x. Alternatively, if $\tilde{x}^k$ is accepted infinitely many times, by Theorem 3.1(b), again we can say that x^* is a KKT point with respect to x. Finally, as pointed out, for instance, in [4], we note that the CPLD constraint qualification required in Theorem 3.1 is known to be a weak condition.

4 Numerical Examples

In this section, we report a preliminary study on the role of primitive directions in MIX-ALM and how they can be employed in practice. To conduct such an experimentation, we use the MINLPLIB library of mixed-integer nonlinear problems available at https://www.minlplib.org/. From the entire collection, we extracted the problems with

$$n_z > 0, \quad n_x > 0, \quad n_z + n_x \leq 30, \quad n_{binary} = 0, \quad n_{cons} > 0,$$

where n_{binary} is the number of binary variables and n_{cons} is the number of general constraints (i.e., constraints other than simple bounds on the variables). As it can be noted, we choose to consider only problems with $n_{binary} = 0$ for the following reason. Indeed, even if primitive directions can be used in principle to handle also binary variables, their use is not the most efficient one to handle binary variables. Indeed, definition and generation of primitive directions (as we shall see) does not take into account that some variables are binary. As a consequence, many of the generated primitive directions would then be discarded, because of the presence of binary variables, thus wasting a considerable amount of computing resources.

With these selection criteria, we get the 30 problems[1] reported in Table 1.

4.1 Implementation Details

The proposed MIX-ALM has been implemented in double precision Fortran 90. In particular, we modified the code of PD-ALM (defined in [16]) by inserting primitive

[1] In order to make all test problems meet the form of (1), with all functions defined over $\mathbb{R}^n$, we had to properly reformulate four test problems, that is, *nvs05*, *nvs22*, *portfol_roundlot* and *st_e32*. Specifically, for these problems, we reformulated the constraints or introduced appropriate lower/upper bounds on the variables so as to exclude points where functions are undefined. This adjustment was necessary because the original problems include ratios without restrictions on the denominator or square roots without sign restrictions on the argument.

Table 1 Set of problems selected for the experiments. The column labeled f^* reports the objective function value on the best known feasible solution

Problem name	#	n_x	n_z	n_{cons}	f^*
du-opt	1	7	13	8	3.5563
du-opt5	2	7	11	6	8.0737
eg_all_s	3	1	7	28	7.6578
eg_disc2_s	4	5	3	28	5.6421
eg_disc_s	5	4	4	28	5.7605
eg_int_s	6	5	3	28	6.4531
gear3	7	4	4	4	0
gear4	8	2	4	1	1.6434
jit1	9	21	4	31	173983.33
nvs01	10	1	2	2	12.4697
nvs02	11	3	5	3	5.9642
nvs05	12	6	2	9	5.4709
nvs08	13	1	2	3	23.4497
nvs14	14	3	5	3	-40358.1548
nvs20	15	11	5	8	230.9222
nvs21	16	1	2	2	-5.6848
nvs22	17	4	4	9	6.0582
portfol_roundlot	18	9	8	11	0.0283
prob10	19	1	1	2	3.4455
st_e32	20	16	18	18	-1.4304
st_e36	21	1	1	2	-246
st_e38	22	2	2	3	7197.7271
st_e40	23	1	3	8	30.4142
windfac	24	11	3	13	0.2545
cvxnonsep_normcon20	25	10	10	1	-21.7491
cvxnonsep_normcon30	26	15	15	1	-34.244
cvxnonsep_nsig20	27	10	10	1	80.9493
cvxnonsep_nsig30	28	15	15	1	130.6287
cvxnonsep_pcon20	29	10	10	1	-21.5123
cvxnonsep_pcon30	30	15	15	1	-35.9868

directions to handle the integer variables. In particular, the continuous subproblem is solved using the ASA-BCP method proposed in [15]; when ASA-BCP is not able to solve the subproblem, we resort to ALGENCAN [3]. Drawing inspiration from [31], since the set of all feasible primitive directions $D(z^k)$ used for the integer variables at each iteration k might have a huge cardinality, then we use a subset $\tilde{D}^k \subseteq D(z^k)$ such that $\tilde{D}^{k-1} \subseteq \tilde{D}^k$. In particular, if $z^{k+1} \neq z^k$, then $\tilde{D}^{k+1}$ for the next iteration is defined such that $\tilde{D}^{k+1} = \tilde{D}^k$. Otherwise, $\tilde{D}^{k+1}$ is such that $\tilde{D}^k \subset \tilde{D}^{k+1}$. More in particular, drawing inspiration from [1, 31], given a parameter $\eta > 1$, let $\{q_t(\eta)\}$ be a sequence of tentative directions where the element $q_t(\eta)$ depends on the t-th element in the n_z-dimensional Sobol sequence [5] (as implemented in [13]), that is

$$q_t(\eta) = \left\lfloor \eta \frac{2u_t - e}{\|2u_t - e\|} \right\rceil \in \mathbb{Z}^n \cap \left[-\eta - \frac{1}{2}, \eta + \frac{1}{2}\right]^n,$$

where u_t is the t-th element of the pseudo-random Sobol sequence. Then, given the set of primitive directions $\tilde{D}^k$, we define $\tilde{D}^{k+1} = \tilde{D}^k \cup \{q_{t^k}(\eta)\}$, where $t^k \geq 1000$ is the smallest integer such that $q_{t^k}(\eta)$ is primitive and $q_{t^k}(\eta) \notin \tilde{D}^k$.

As concerns the parameters of MIX-ALM, they were set as follows:

$$\beta = 0.9, \quad \Delta_0 = 10^3, \quad \xi_0 = 10^{-12}.$$

Note that parameters $\eta, \theta, \epsilon_0, \bar{\mu}_0, \mu_0$ and the sequence $\{\tau^k\}$ are the same as those defined in ALGENCAN [3].

In the numerical experiments, we stop MIX-ALM when the current primal-dual point $(x^k, z^k, \bar{\mu}^k)$ satisfies all the following three conditions:

1. $\|x^k - \mathcal{P}_{[l,u]}(x^k - \nabla_x L(x^k, z^k, \bar{\mu}^k))\|_\infty \leq 10^{-6} \max\{1, \|\nabla_x f(x^k, z^k)\|_\infty\}$,
2. $\|h(x^k, z^k)\|_\infty \leq 10^{-6} \max\{1, \|h(x_0, z_0)\|_\infty\}$,
3. $|\tilde{D}^k| \geq$ `max_dir` and $z^{k-1} = z^k$,

for a given choice of the parameter `max_dir`. Additionally, we stopped the algorithm when the penalty parameter ϵ^k falls below 10^{-20}.

4.2 Comparison on the Number of Primitive Directions

In this section, we compare two versions of MIX-ALM obtained by choosing `max_dir` $\in \{10n_z, 1000n_z\}$. Tests have been run on an Intel 13th Generation Core i9-13900 at 5.8Ghz with Ubuntu 22.04 and 32Gb RAM. The results are reported in Table 2 where, denoting the final point returned by the algorithms by $(\bar{x}, \bar{z})$, the column labels have the following meaning:

- $f = f(\bar{x}, \bar{z})$. i.e., the objective function value;
- `feaserr`$= \|h(\bar{x}, \bar{z})\|_\infty$, i.e., the feasibility error;
- `feastol`$= 10^{-6}\|h(x_0, z_0)\|_\infty$, i.e., the feasibility tolerance;
- `opterr`$= \|\bar{x} - \mathcal{P}_{[l,u]}(\bar{x} - \nabla_x L(\bar{x}, \bar{z}, \bar{\mu}))\|_\infty$, i.e., the optimality error with respect to x;
- `opttol`$= 10^{-6} \max\{1, \|\nabla_x f(\bar{x}, \bar{z})\|_\infty\}$, i.e., the optimality tolerance;
- `it` is the number of iterations;
- `nf` is the number of function evaluations.

We observe from Table 2 that, using `max_dir` $= 10n_z$, MIX-ALM is able to solve 20 problems, while 24 problems are solved by using `max_dir` $= 1000n_z$. Moreover, all failures are due to the penalty parameter ϵ^k falling below the threshold of 10^{-20}, except for problem 23 using `max_dir` $= 10n_z$, where the algorithm returns a point satisfying a stationarity condition with respect to the feasibility violation (see item (c) of Theorem 3.1, item (b) of Theorem 3.2 and Remark 3.2). We observe that problems `nvs05` (problem 12), `nvs22` (problem 17) and `portfol_roundlot` (problem 18)

Table 2 Results obtained by using a number of primitive directions max_dir $\in \{10n_z, 1000n_z\}$

#	max_dir	f	feaserr	feastol	opterr	opttol	it	nf
1	$10n_z$	3.56E+03	0.00E+00	3.80E-05	3.82E-11	7.66E-02	6	1039
	$1000n_z$	3.56E+03	0.00E+00	3.80E-05	3.82E-11	7.66E-02	619	268141
2	$10n_z$	Failure due to $\epsilon < 10^{-20}$						
	$1000n_z$	3.99E+03	2.67E-15	3.40E-05	1.93E-02	8.17E-02	524	230774
3	$10n_z$	Failure due to $\epsilon < 10^{-20}$						
	$1000n_z$	8.31E+00	2.22E-15	7.37E-05	1.02E-07	1.00E-06	334	23705
4	$10n_z$	5.68E+00	4.30E-08	2.91E-05	2.61E-07	1.00E-06	5	506
	$1000n_z$	5.68E+00	1.94E-07	2.91E-05	5.33E-07	1.00E-06	145	71839
5	$10n_z$	5.76E+00	5.79E-07	3.57E-05	8.91E-09	1.00E-06	4	385
	$1000n_z$	5.76E+00	1.39E-07	3.57E-05	1.38E-07	1.00E-06	192	93188
6	$10n_z$	8.03E+00	1.13E-06	1.24E-04	5.46E-09	1.00E-06	4	324
	$1000n_z$	8.03E+00	8.38E-11	1.24E-04	9.96E-07	1.00E-06	145	2500
7	$10n_z$	7.32E-01	8.87E-08	1.00E-06	1.29E-10	1.00E-06	3	80
	$1000n_z$	7.32E-01	8.87E-08	1.00E-06	8.87E-08	1.00E-06	192	71479
8	$10n_z$	8.56E+05	2.91E-11	8.56E-01	2.91E-12	1.00E-06	5	123
	$1000n_z$	8.56E+05	8.73E-11	8.56E-01	9.06E-07	1.00E-06	193	69833
9	$10n_z$	2.73E+05	4.11E-07	1.00E-06	2.31E+00	1.77E+02	8	428
	$1000n_z$	2.73E+05	2.67E-12	1.00E-06	5.86E-06	1.77E+02	193	72976
10	$10n_z$	4.92E+02	1.46E-11	9.56E-01	5.27E-07	1.00E-06	12	2046
	$1000n_z$	4.92E+02	1.82E-12	9.56E-01	5.28E-07	1.00E-06	101	69008
11	$10n_z$	7.88E+00	0.00E+00	1.26E-04	1.88E-11	1.00E-06	5	905
	$1000n_z$	7.88E+00	0.00E+00	1.26E-04	1.95E-08	1.00E-06	239	1194971

Table 2 continued

#	max_dir	f	feaserr	feastol	opterr	opttol	it	nf
12	$10n_z$	Failure due to $\epsilon < 10^{-20}$						
	$1000n_z$	Failure due to $\epsilon < 10^{-20}$						
13	$10n_z$	2.60E+01	2.67E-07	6.00E-06	9.80E-11	1.00E-05	5	414
	$1000n_z$	2.60E+01	3.44E-08	6.00E-06	8.95E-06	1.00E-05	101	35214
14	$10n_z$	Failure due to $\epsilon < 10^{-20}$						
	$1000n_z$	-1.26E+04	1.78E-15	1.26E-04	4.11E-07	1.00E-06	239	718323
15	$10n_z$	4.38E+02	1.02E-09	1.39E-06	9.75E-05	1.37E-04	4	297
	$1000n_z$	4.38E+02	3.07E-13	1.39E-06	9.99E-05	1.37E-04	239	288857
16	$10n_z$	0.00E+00	1.14E-11	1.00E-06	9.33E-08	1.00E-06	33	5742
	$1000n_z$	0.00E+00	1.14E-11	1.00E-06	4.74E-10	1.00E-06	34	5745
17	$10n_z$	Failure due to $\epsilon < 10^{-20}$						
	$1000n_z$	Failure due to $\epsilon < 10^{-20}$						
18	$10n_z$	Failure due to $\epsilon < 10^{-20}$						
	$1000n_z$	Failure due to $\epsilon < 10^{-20}$						
19	$10n_z$	2.74E+01	4.44E-16	1.00E-06	1.22E-13	1.00E-06	2	56
	$1000n_z$	2.74E+01	4.44E-16	1.00E-06	1.22E-13	1.00E-06	2	56
20	$10n_z$	Failure due to $\epsilon < 10^{-20}$						

Table 2 continued

#	max_dir	f	feaserr	feastol	opterr	opttol	it	nf
	$1000n_z$	Failure due to $\epsilon < 10^{-20}$						
21	$10n_z$	Failure due to $\epsilon < 10^{-20}$						
	$1000n_z$	Failure due to $\epsilon < 10^{-20}$						
22	$10n_z$	7.20E+03	3.42E-05	9.27E-01	1.51E-04	1.85E-04	340	182451
	$1000n_z$	7.20E+03	3.42E-05	9.27E-01	1.51E-04	1.85E-04	340	182451
23	$10n_z$	4.41E+00	1.79E+00	3.46E-06	1.19E-14	1.00E-06	2	47
	$1000n_z$	3.34E+01	4.44E-16	3.46E-06	1.42E-11	1.00E-06	144	46807
24	$10n_z$	Failure due to $\epsilon < 10^{-20}$						
	$1000n_z$	Failure due to $\epsilon < 10^{-20}$						
25	$10n_z$	-1.77E+01	2.30E-12	1.00E-06	2.82E-13	1.00E-06	5	453
	$1000n_z$	-1.77E+01	2.30E-12	1.00E-06	8.54E-07	1.00E-06	476	1472856
26	$10n_z$	-2.98E+01	2.94E-07	1.00E-06	1.86E-07	1.00E-06	4	309
	$1000n_z$	-2.98E+01	2.94E-07	1.00E-06	1.86E-07	1.00E-06	4	309
27	$10n_z$	8.39E+01	5.19E-12	1.00E-06	4.60E-07	1.86E-06	8	1355
	$1000n_z$	8.15E+01	2.26E-12	1.00E-06	1.35E-06	1.86E-06	476	2249371
28	$10n_z$	1.31E+02	1.83E-13	1.00E-06	7.36E-08	1.86E-06	7	1613
	$1000n_z$	1.31E+02	1.83E-13	1.00E-06	6.27E-07	1.86E-06	714	3354395
29	$10n_z$	-1.28E+01	1.46E-11	1.00E-06	7.64E-07	1.00E-06	5	35515
	$1000n_z$	-1.28E+01	0.00E+00	1.00E-06	7.64E-07	1.00E-06	476	95338
30	$10n_z$	-2.28E+01	6.69E-08	1.00E-06	3.29E-07	1.00E-06	14	54289
	$1000n_z$	-2.28E+01	0.00E+00	1.00E-06	3.29E-07	1.00E-06	714	81968

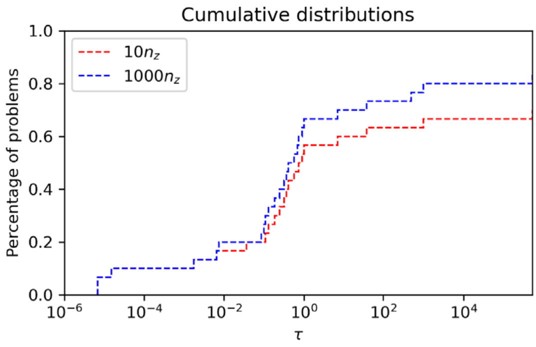

Fig. 1 Comparison between versions of MIX-ALM with max_dir $= 10n_z$ and max_dir $= 1000n_z$. For each value of τ, the curves represent the percentage of problem solved with accuracy τ and feasibility error below 10^{-4}, according to (23), using a maximum number of primitive directions equal to $10n_z$ and $1000n_z$

cannot be solved by Algorithm 1 even when the integrality constraints on the variables are relaxed with the required accuracy. About the results reported in Table 2 it is also worth mentioning that MIX-ALM using $10n_z$ primitive directions completed the run on all the problems in slightly more than one minute CPU time, whereas MIX-ALM using $1000n_z$ primitive directions took slightly more than ten minutes CPU time.

In order to make more evident the benefit of increasing the maximum number of primitive directions, we also depict the results of Table 2 in Figure 1. In particular, as proposed in [29], using a maximum number of primitive directions $s \in \{10n_z, 1000n_z\}$ and for each problem $p \in \mathcal{P}$, we compute the values

$$d_{sp} = \frac{|f_{sp} - f_p^*|}{\max\{1, |f_p^*|\}},$$

where f_{sp} is the final function value for given s and p, while f_p^* is the best known function value of problem p. Then, drawing inspiration from performance profiles introduced in [19], for each value of s, we draw the cumulative distribution $c_s(\tau)$, where

$$c_s(\tau) = \frac{1}{|\mathcal{P}|} \left| \{p \in \mathcal{P} : d_{sp} \le \tau \ \wedge \ \texttt{feaserr}_{sp} \le 10^{-4}\} \right|, \tag{23}$$

with $\texttt{feaserr}_{sp}$ denoting the feasibility error obtained for any given s and p.

From Figure 1, it is evident that increasing the maximum number of primitive directions improves the algorithm performances. Namely, the version of MIX-ALM using max_dir $= 1000n_z$ is better than the version using max_dir $= 10n_z$ with respect to the percentage of problems solved with a given accuracy. This is in line with what one can reasonably expect since the higher the number of primitive directions the larger the neighborhoods of integer variables explored by the algorithm.

4.3 Comparison with a derivative-free algorithm

In this section we present a comparison between MIX-ALM and the derivative-free method for constrained mixed integer problems DFLvinc freely available as part of the GitHub repository https://github.com/DerivativeFreeLibrary and proposed in [30].

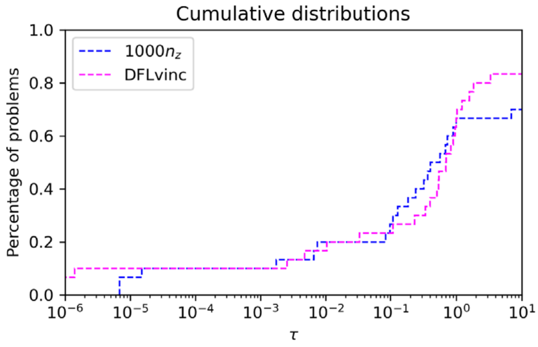

Fig. 2 Comparison between MIX-ALM (using $1000n_z$ primitive directions) and `DFLvinc` on the set of problems reported in Table 1

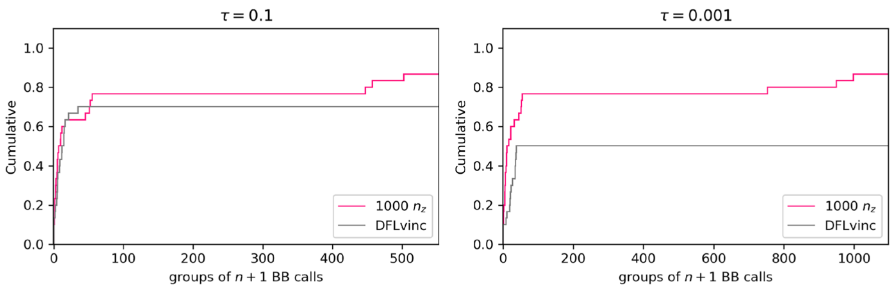

Fig. 3 Data profiles relative to the comparison between MIX-ALM (using $1000n_z$ primitive directions) and `DFLvinc` on the set of problems reported in Table 1

We run `DFLvinc` adopting a tolerance 10^{-6} for the maximum steplength in the stopping condition. Furthermore, we stop `DFLvinc` when the total number of function evaluations reaches $\max\{1000n, 10000\}$ (where n is the number of variables).

In Figure 2 we report the comparison between MIX-ALM (using $1000n_z$ primitive directions) and `DFLvinc` by means of the cumulative distributions defined in (23). Figure 3, reports the comparison between MIX-ALM (using $1000n_z$ primitive directions) and DFLvinc by means of data profiles (see [32] for their definition and [21] for their adaptation in the constrained context) for $\tau = 10^{-1}$ and 10^{-3}. Note that, the data profiles asses the ability of a given solver to find a feasible point whose objective function value is close to the best function value found by any solver. Hence, considering Figure 2, we can say that MIX-ALM and DFLvinc have almost comparable performances with respect to the ability to locate a good approximation of the known global minimum point, even though for low precisions (τ large) DFLvinc is better than MIX-ALM. On the other hand, from Figure 3, we can see that MIX-ALM completely dominates DFLvinc which is totally reasonable since MIX-ALM uses second order derivatives.

From Figure 2, `DFLvinc` might seem to outperform MIX-ALM on the test problems. This better performance can be explained considering the known ability of derivative-free methods to filter out local minima (which is the idea at the basis of the implicit filtering approach [28]). The considered set of problems (which are of relatively small dimensions) does not allow to evidence the superiority of the augmented

Table 3 Set of problems with $n_x \geq 100$ and $n_z \geq 1$ and $m > 0$

Problem name	n_x	n_z	n_{cons}	f^*
space960	4577	960	6497	$8.255 \cdot 10^6$
gastrans582_cold13	1936	250	3732	0
gastrans582_cold13_95	1936	250	3732	0
gastrans582_cold17	1936	250	3732	0
gastrans582_cold17_95	1936	250	3732	0
gastrans582_cool12	1936	250	3732	0
gastrans582_cool12_95	1936	250	3732	0
gastrans582_cool14	1936	250	3732	0
gastrans582_cool14_95	1936	250	3732	0
gastrans582_freezing27	1936	250	3732	0
gastrans582_freezing27_95	1936	250	3732	0
gastrans582_freezing30	1936	250	3732	0
gastrans582_freezing30_95	1936	250	3732	0
gastrans582_mild10	1936	250	3732	0
gastrans582_mild10_95	1936	250	3732	0
gastrans582_mild11	1936	250	3732	0
gastrans582_mild11_95	1936	250	3732	0
gastrans582_warm15	1936	250	3732	0
gastrans582_warm15_95	1936	250	3732	0
gastrans582_warm31	1936	250	3732	0
gastrans582_warm31_95	1936	250	3732	0
gastrans135	961	232	2472	0
gastrans040	231	48	553	0
o9_ar4_1	108	72	435	236.1385
fo9_ar25_1	108	72	435	32.1864
fo9_ar2_1	108	72	435	32.6250
fo9_ar3_1	108	72	435	24.8155
fo9_ar4_1	108	72	435	23.4643
fo9_ar5_1	108	72	435	23.4643

Lagrangian approach in dealing the continuous constrained subproblem with respect to the derivative-free approach of `DFLvinc`. In order to try and assess the superiority of MIX-ALM, we selected a second set of test problems from MINLPLIB [26], namely those with $n_x \geq 100$ and $n_z \geq 1$ and $m > 0$. These problems are reported in Table 3.

Then, in Table 4, we report the results on these larger problems obtained with MIX-ALM and `DFLvinc`. As it can be noted, MIX-ALM is always able to find feasible solutions (at least up to the required accuracy) whereas `DFLvinc` finds a feasible solution only for problem `space960`.

Table 4 Results on problems from Table 3

Problem name	MIX-ALM f	feaserr	DFLvinc f	feaserr
space960	7.61E+06	7.44E-06	1.75E+07	0.00E+00
gastrans582_cold13	0.00E+00	4.01E-04	0.00E+00	1.62E+04
gastrans582_cold13_95	0.00E+00	5.39E-04	0.00E+00	1.53E+04
gastrans582_cold17	0.00E+00	6.40E-04	0.00E+00	1.70E+04
gastrans582_cold17_95	0.00E+00	2.13E-03	0.00E+00	1.73E+04
gastrans582_cool12	0.00E+00	3.98E-04	0.00E+00	1.76E+04
gastrans582_cool12_95	0.00E+00	8.34E-04	0.00E+00	1.70E+04
gastrans582_cool14	0.00E+00	1.59E-04	0.00E+00	1.75E+04
gastrans582_cool14_95	0.00E+00	1.02E-03	0.00E+00	1.71E+04
gastrans582_freezing27	0.00E+00	1.84E-03	0.00E+00	2.02E+04
gastrans582_freezing27_95	0.00E+00	4.38E-04	0.00E+00	2.00E+04
gastrans582_freezing30	0.00E+00	9.97E-04	0.00E+00	1.97E+04
gastrans582_freezing30_95	0.00E+00	3.55E-04	0.00E+00	2.00E+04
gastrans582_mild10	0.00E+00	3.55E-04	0.00E+00	1.71E+04
gastrans582_mild10_95	0.00E+00	1.21E-03	0.00E+00	1.72E+04
gastrans582_mild11	0.00E+00	3.51E-03	0.00E+00	1.66E+04
gastrans582_mild11_95	0.00E+00	1.37E-03	0.00E+00	1.78E+04
gastrans582_warm15	0.00E+00	6.56E-04	0.00E+00	1.63E+04
gastrans582_warm15_95	0.00E+00	3.10E-04	0.00E+00	1.60E+04
gastrans582_warm31	0.00E+00	4.37E-06	0.00E+00	1.58E+04
gastrans582_warm31_95	0.00E+00	2.25E-05	0.00E+00	1.64E+04
gastrans135	0.00E+00	1.119E+02	0.00E+00	5.05E+03
gastrans040	0.00E+00	4.291E+02	0.00E+00	5.33E+01
o9_ar4_1	1.86E+02	6.923E-01	5.023E+02	3.01E+01
fo9_ar25_1	2.06E+01	1.485E+00	7.39E+01	2.37E+01
fo9_ar2_1	2.40E+01	2.104E+00	7.57E+01	2.15E+01
fo9_ar3_1	1.88E+01	1.086E+00	8.25E+01	2.31E+01
fo9_ar4_1	1.62E+01	6.923E-01	8.66E+01	2.34E+01
fo9_ar5_1	1.47E+01	4.929E-01	8.15E+01	1.78E+01

5 Conclusions

In this paper, we have presented a new method for mixed-integer nonlinear constrained optimization, particularly well suited for those problems where integrality constraints cannot be relaxed. In such a context, the proposed algorithm alternates two main operations at each iteration. First, it updates the continuous variables by means of an augmented Lagrangian scheme using Newton-type directions and an active-set strategy. Then, integer variables are updated by means of primitive directions. Under appropriate assumptions, we have established convergence to points satisfying KKT conditions with respect to the continuous variables, with the integer variables guaranteeing local optimality of the objective function. Finally, we performed a preliminary numerical experience to show the effect of increasing the size of the set of primitive directions. The numerical results confirmed that, by increasing the number of primitive directions, better results are obtained, as one can reasonably expect.

Acknowledgements The Authors would like to thank three anonymous Reviewers and the Associated Editor for their comments and suggestions which helped to improve the paper.

Funding Open access funding provided by Università degli Studi di Roma La Sapienza within the CRUI-CARE Agreement.

Data Availability The problems used in this study are available as part of the MINLPLIB collection of problems and can be found at the URL: https://www.minlplib.org/. For four problems, listed in the footnote at page , a reformulation has been used in the numerical experiments. Data sets, codes and results generated during the current study are available from the corresponding author on reasonable request.

Declarations

Conflicts of Interest No funding was received for conducting this study. The authors have no relevant financial or non-financial interests to disclose.

References

1. Abramson, M.A., Audet, C., Dennis, J.E., Jr., Digabel, S.L.: Orthomads: a deterministic mads instance with orthogonal directions. SIAM J. Optim. **20**(2), 948–966 (2009). https://doi.org/10.1137/080716980
2. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: On augmented Lagrangian methods with general lower-level constraints. SIAM J. Optim. **18**(4), 1286–1309 (2007). https://doi.org/10.1137/0606547
3. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: Augmented Lagrangian methods under the constant positive linear dependence constraint qualification. Math. Program. **111**, 5–32 (2008). https://doi.org/10.1007/s10107-006-0077-1
4. Andreani, R., Martínez, J.M., Schuverdt, M.L.: On the relation between constant positive linear dependence condition and quasinormality constraint qualification. J. Optim. Theory Appl. **125**(2), 473–483 (2005). https://doi.org/10.1007/s10957-004-1861-9
5. Antonov, I.A., Saleev, V.: An economic method of computing lp_τ-sequences. USSR Comput. Math. Math. Phys. **19**(1), 252–256 (1979). https://doi.org/10.1016/0041-5553(79)90085-5
6. Belotti, P., Bonami, P., Forrest, J.J., Ladanyi, L., Laird, C., Lee, J., Margot, F., Waechter, A.: Basic Open-source Nonlinear Mixed Integer Programming (2010). https://www.coin-or.org/Bonmin/
7. Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., Mahajan, A.: Mixed-integer nonlinear optimization. Acta Numer **22**, 1–131 (2013). https://doi.org/10.1017/S0962492913000032
8. Bertsekas, D.P.: Constrained Optimization and Lagrange Multiplier Methods. Academic Press, New York (1982). https://doi.org/10.1016/C2013-0-10366-2
9. Birgin, E.G., Martínez, J.M.: Practical Augmented Lagrangian Methods for Constrained Optimization. SIAM, Philadelphia (2014). https://doi.org/10.1137/1.9781611973365
10. Birgin, E.G., Martínez, J.M.: Block coordinate descent for smooth nonconvex constrained minimization. Comput. Optim. Appl. **83**(1), 1–27 (2022). https://doi.org/10.1007/s10589-022-00389-5
11. Garnett, Roman: Bayesian Optimization. Cambridge University Press (2023)

12. Bolusani, S., Besançon, M., Bestuzheva, K., Chmiela, A., Dionísio, J., Donkiewicz, T., van Doornmalen, J., Eifler, L., Ghannam, M., Gleixner, A., Graczyk, C., Halbig, K., Hedtke, I., Hoen, A., Hojny, C., van der Hulst, R., Kamp, D., Koch, T., Kofler, K., Lentz, J., Manns, J., Mexi, G., Mühmer, E., Pfetsch, M.E., Schlösser, F., Serrano, F., Shinano, Y., Turner, M., Vigerske, S., Weninger, D., Xu, L.: The SCIP Optimization Suite 9.0. ZIB-Report 24-02-29, Zuse Institute Berlin (2024). https://nbn-resolving.org/urn:nbn:de:0297-zib-95528
13. Bratley, P., Fox, B.: Implementing Sobol's quasirandom sequence generator (algorithm 659). ACM Transactions on Mathematical Software **29**(1), 49–57 (2003). https://doi.org/10.1145/42288.214372
14. Bueno, L.F., Haeser, G., Kolossoski, O.: On the paper "augmented Lagrangian algorithms for solving the continuous nonlinear resource allocation problem". Eur. J. Oper. Res. **313**(3), 1217–1222 (2024). https://doi.org/10.1016/j.ejor.2023.11.001
15. Cristofari, A., De Santis, M., Lucidi, S., Rinaldi, F.: A two-stage active-set algorithm for bound-constrained optimization. J. Optim. Theory Appl. **172**(2), 369–401 (2017). https://doi.org/10.1007/s10957-016-1024-9
16. Cristofari, A., Di Pillo, G., Liuzzi, G., Lucidi, S.: An augmented Lagrangian method exploiting an active-set strategy and second-order information. J. Optim. Theory Appl. **193**(1–3), 300–323 (2022). https://doi.org/10.1007/s10957-022-02003-4
17. Dakin, R.J.: A tree-search algorithm for mixed integer programming problems. Comput. J. **8**(3), 250–255 (1965). https://doi.org/10.1093/comjnl/8.3.250
18. De Santis, M., Di Pillo, G., Lucidi, S.: An active set feasible method for large-scale minimization problems with bound constraints. Comput. Optim. Appl. **53**(2), 395–423 (2012). https://doi.org/10.1007/s10589-012-9506-7
19. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002). https://doi.org/10.1007/s101070100263
20. Facchinei, F., Lucidi, S.: Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems. J. Optim. Theory Appl. **85**(2), 265–289 (1995). https://doi.org/10.1007/BF02192227
21. Fasano, G., Liuzzi, G., Lucidi, S., Rinaldi, F.: A linesearch-based derivative-free approach for non-smooth constrained optimization. SIAM J. Optim. **24**(3), 959–992 (2014). https://doi.org/10.1137/130940037
22. Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation. Math. Program. **66**, 327–349 (1994). https://doi.org/10.1007/BF01581153
23. Garneau, M.L.: Modelling of a Solar Thermal Power Plant for Benchmarking Blackbox Optimization Solvers. Master's thesis, Polytechnique Montréal (2015). https://publications.polymtl.ca/1996/
24. Geoffrion, A.M.: Generalized denders decomposition. J. Optim. Theory Appl. **10**, 237–260 (1972). https://doi.org/10.1007/BF00934810
25. Gill, P.E., Zhang, M.: A projected-search interior-point method for nonlinearly constrained optimization. Comput. Optim. Appl. **88**(1), 37–70 (2024). https://doi.org/10.1007/s10589-023-00549-1
26. Gleixner, A., Hendel, G., Gamrath, G., Achterberg, T., Bastubbe, M., Berthold, T., Christophel, P.M., Jarck, K., Koch, T., Linderoth, J., Lübbecke, M., Mittelmann, H.D., Ozyurt, D., Ralphs, T.K., Salvagnin, D., Shinano, Y.: MIPLIB 2017: data-driven compilation of the 6th mixed-integer programming library. Math. Program. Comput. (2021). https://doi.org/10.1007/s12532-020-00194-3
27. Gupta, O.K., Ravindran, A.: Branch and bound experiments in convex nonlinear integer programming. Manage. Sci. **31**(12), 1533–1546 (1985). https://doi.org/10.1287/mnsc.31.12.1533
28. Kelley, C.T.: Implicit Filtering. SIAM (2011)
29. Liuzzi, G., Lucidi, S., Piccialli, V.: A partition-based global optimization algorithm. J. Global Optim. **48**(1), 113–128 (2010). https://doi.org/10.1007/s10898-009-9515-y
30. Liuzzi, G., Lucidi, S., Rinaldi, F.: Derivative-free methods for mixed-integer constrained optimization problems. J. Optim. Theory Appl. **164**(3), 933–965 (2015). https://doi.org/10.1007/s10957-014-0617-4
31. Liuzzi, G., Lucidi, S., Rinaldi, F.: An algorithmic framework based on primitive directions and non-monotone line searches for black-box optimization problems with integer variables. Math. Program. Comput. **12**(4), 673–702 (2020). https://doi.org/10.1007/s12532-020-00182-7
32. Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. SIAM J. Optim. **20**(1), 172–191 (2009). https://doi.org/10.1137/080724083
33. Qi, L., Wei, Z.: On the constant positive linear dependence condition and its application to SQP methods. SIAM J. Optim. **10**(4), 963–981 (2000). https://doi.org/10.1137/S105262349732662

34. Stubbs, R.A., Mehrotra, S.: A branch-and-cut method for 0–1 mixed convex programming. Math. Program. **86**, 515–532 (1999). https://doi.org/10.1007/s101070050103
35. Stubbs, R.A., Mehrotra, S.: Generating convex polynomial inequalities for mixed 0–1 programs. J. Global Optim. **24**, 311–332 (2002). https://doi.org/10.1023/A:1020351410169