Optimization Letters (2025) 19:1883–1904
https://doi.org/10.1007/s11590-025-02190-9

ORIGINAL PAPER

Mixed-integer linearity in nonlinear optimization: a trust region approach

Alberto De Marchi[1]

Received: 11 March 2024 / Accepted: 5 February 2025 / Published online: 26 February 2025

Abstract
Bringing together nonlinear optimization with polyhedral and integrality constraints enables versatile modeling, but poses significant computational challenges. We investigate a method to address these problems based on sequential mixed-integer linearization with trust region safeguard, computing feasible iterates via calls to a generic mixed-integer linear solver. Convergence to critical, possibly suboptimal, feasible points is established for arbitrary starting points. Finally, we present numerical applications in nonsmooth optimal control and optimal network design and operation.

Keywords Mixed-integer programming · Nonlinear programming · Successive linearization schemes · Trust region methods

1 Introduction

The operation of complex, integrated process systems demands efficient use of resources whilst imposing tight safety constraints. Mixed-integer optimization provides a powerful and flexible mathematical template for modeling many tasks that involve discrete and continuous variables [1, 2]. This work lies at the intersection of nonlinear and integer programming, leaning toward a continuous optimization perspective, and explores how nonlinear programming (NLP) techniques can help as heuristics. We focus here on the minimization of a smooth function subject to polyhedral constraints, an important problem class in NLP, complemented by integrality specifications. The denomination "mixed-integer linearity" in the title refers to the mixed nature of real- and integer-valued decision variables and to the linear

✉ Alberto De Marchi
alberto.demarchi@unibw.de

[1] Institute of Applied Mathematics and Scientific Computing, Department of Aerospace Engineering, University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany

relationships restricting their admissible values. In particular, we consider optimization problems with a nonlinear objective and mixed-integer linear constraints in the form

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\quad \text{over} \quad x \in \mathcal{X} := \bar{\mathcal{X}} \cap \left\{ x \in \mathbb{R}^n \,|\, x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I} \right\} \end{aligned} \tag{P}$$

with objective function $f : \mathcal{X} \to \mathbb{R}$ and nonempty feasible set $\mathcal{X} \subseteq \mathbb{R}^n$. The latter is described by a closed convex polyhedral set $\bar{\mathcal{X}} \subseteq \mathbb{R}^n$ and integrality constraints defined by some index set $\mathcal{I} \subseteq \{1, \dots, n\}$. We intend to exploit the tractable, although difficult, structure of $\mathcal{X}$ in our numerical method, in particular avoiding quadratic terms that may arise from local models or regularizations and instead relying on mixed-integer linear programs (MILP) as subproblems. To do so, we follow [3, Sect. 3] and make use of polyhedral norms, e.g., ℓ_p with $p \in \{1, \infty\}$, instead of the classical Euclidean ℓ_2. Then, a practical assumption is that MILPs involving $\mathcal{X}$ can be efficiently solved, leveraging mature LP and MILP technology, despite their NP-completeness.

We investigate (P) with the implicit assumption that there are both real- and integer-valued variables as well as some couplings thereof in $\mathcal{X}$ and nonlinearities in f. Should this not be the case, off-the-shelf methods for MILP and NLP could be readily adopted for tackling (P). Then, partitioning the decision variables x into real-valued u and integer-valued z, we consider hereafter some blanket assumptions on (P). First of all, we suppose that

(A1) function f is continuously differentiable and ∇f is locally Lipschitz continuous;
(A2) the objective function f is of the form $(u, z) \mapsto f_1(u) + \langle f_2, z \rangle$, for some smooth function f_1 and vector f_2 of suitable size;
(A3) all feasible values for the integer-valued variables z lie in a bounded set.

Assumption (A1) concerns the objective function f and its smoothness, and requires only local Lipschitz gradient continuity [4]. Assumption (A2) of integer-linearity for f is introduced, in line with [5], to rule out the need for derivative approximations or exotic notions; e.g., the difference formula in [6]. Notice, however, that Assumption (A2) is considered for clarity of presentation and it is not restrictive. A generic nonlinear, albeit smooth, dependence of f on the integer-valued variables can always be incorporated with a suitable reformulation that satisfies Assumption (A2), possibly adding auxiliary real-valued variables and complementing $\mathcal{X}$ with linear equality constraints. Specification (A3) is not only reasonable and often valid in practice [1], the prominent modeling tool being binary variables, but it also greatly simplifies the presentation, with minor algorithmic consequences.

Focusing on (P) without any convexity assumptions, we will not demand global optimality but seek instead an affordable local "solution". The underlying optimality notion is defined by localization in a trust region. We analyze the similarities of projected gradient-type methods with trust region linearization schemes, highlighting how polyhedral norms may lead to undesirable properties for the former, making the latter an interesting alternative. After detailing an iterative numerical scheme, we

investigate its global convergence properties, in the sense of arbitrary initial points. The algorithm generates feasible iterates with (nonmonotonically) decreasing objective, and relies on averaged merit values to monitor and enforce convergence.

Related work

The ideas behind sequential linearization and trust region methods are not new [7, 8], but there has been little progress toward relaxing some fundamental assumptions. Sequential linearization methods for problems with polyhedral constraints heavily exploit convexity of the feasible set [8], notwithstanding that the approach has been applied to particular nonconvex problems as well, prominently by [9] for complementarity constraints. In contrast, proximal algorithms have been developed for the fully nonconvex structured setting, with globalization based on line search [10] and trust region [11]. However, since they rely on local quadratic models to achieve sufficient decrease, their direct application to (P) is hindered in practice.

Contributions and outline

We trace back the sequential linearization approach and overcome the issues arising from the feasible set being nonconvex. This is made possible by treating the integralities through a black-box, both theoretically and numerically. Section 2 is dedicated to optimality concepts for (P) and focuses on a notion of criticality that requires access only to a linear minimization oracle. We propose in Sect. 3 a numerical method based on sequential mixed-integer linearization, with a globalization mechanism that brings together trust region and line search. The algorithm is characterized and global convergence results are established under mild assumptions. Finally, in Sect. 4 we report numerical results on two nontrivial example problems, illustrating the algorithmic behavior and performance.

2 Approach and optimality concepts

This section is dedicated to juxtaposing projection and trust region approaches to tackle (P). Optimality notions and characterizations are then discussed in view of the numerical algorithm developed later on.

2.1 From projections to trust regions

Attracted by the success of proximal-gradient methods to numerically solve nonconvex nonsmooth problems, one may be tempted to address (P) with such methods recast with polyhedral norms. The classical *round* setting involves an ℓ_2-norm minimization to compute a projection onto the feasible set $\mathcal{X}$, whereas polyhedral projections based on ℓ_1- or ℓ_∞-norm can be reformulated in terms of MILP [8, 9, 11]. A round projected gradient update at $\bar{x} \in \mathcal{X}$ is given by

$$x^+ \in \operatorname{proj}_{\mathcal{X}}(\bar{x} - \gamma \nabla f(\bar{x})) = \operatorname*{arg\,min}_{x \in \mathcal{X}} \frac{1}{2\gamma} \|\bar{x} - \gamma \nabla f(\bar{x}) - x\|_2^2 \tag{1}$$

for some stepsize $\gamma > 0$. Owing to the structure of $\mathcal{X}$, the round projection requires solving a mixed-integer *quadratic* program (MIQP), because of the quadratic term. Seeking a MILP subproblem, one could replace the Euclidean norm with a polyhedral one as distance metric and update according to

$$x^+ \in \arg\min_{x \in \mathcal{X}} \|\bar{x} - \gamma \nabla f(\bar{x}) - x\|_p. \tag{2}$$

Unfortunately, taking such projections with $p \in \{1, \infty\}$ can lead to odd situations. In fact, not only polyhedral projections can be set-valued even for convex sets, but a (badly selected) projected gradient may yield an increase of the objective, regardless of the stepsize.

Moving away from the projection point-of-view, we can consider a linear model of f around $\bar{x}$ and, to avoid artificial unboundedness, replace the quadratic regularization term inherent in (1) with a trust region constraint. Such trust region subproblem at $\bar{x} \in \mathcal{X}$ with radius $\Delta > 0$ reads

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle \qquad \text{subject to} \quad \|x - \bar{x}\|_{PL} \leq \Delta, \tag{3}$$

where $\|\cdot\|_{PL} : \mathbb{R}^n \to \mathbb{R}_+$ denotes the application of a norm on the real-valued entries only, according to the index set $\mathcal{I}$.[1] Despite the trust region stipulation being relative only to the real-valued variables, hence the notation "PL" for "partial localization", (3) is well posed owing to Assumption (A3), which makes the intersection of $\mathcal{X}$ with (closed) *PL-balls*

$$\mathbb{B}_{PL}(x, \Delta) := \left\{ w \in \mathbb{R}^n \mid \|w - x\|_{PL} \leq \Delta \right\}$$

always a compact set. Instead, the partial localization with $\|\cdot\|_{PL}$ gives rise to a more inclusive definition of neighborhoods, and hence a comparatively strong concept of "local optimality", see Sect. 2.2 below.

Since the construction of (3) is similar in spirit to (1), one can expect a connection between the two approaches. For an ℓ_p-projected gradient scheme, $1 \leq p \leq \infty$, the next iterate from a feasible point $\bar{x} \in \mathcal{X}$ is obtained from (2), whose solutions $x^+ \in \mathcal{X}$ must satisfy $\|x^+ - \bar{x} + \gamma \nabla f(\bar{x})\|_p \leq \gamma \|\nabla f(\bar{x})\|_p$ and, using the triangle inequality,

$$\|x^+ - \bar{x}\|_p \leq \|x^+ - \bar{x} + \gamma \nabla f(\bar{x})\|_p + \| - \gamma \nabla f(\bar{x})\|_p \leq 2\gamma \|\nabla f(\bar{x})\|_p.$$

This means that a projected gradient subproblem imposes a trust region-like condition, whereas a trust region subproblem does not seek a projection but instead to minimize (a local model of) the objective function. Thus, as on a quest to avoid round terms, we explore the more natural trust region perspective with polyhedral

[1] If the index set $\mathcal{I}$ is nonempty, $\|\cdot\|_{PL}$ lacks positive definiteness, since $\|x\|_{PL} = 0 \not\Rightarrow x = 0$, and therefore it is not a norm. However, $\|\cdot\|_{PL}$ is subadditive and absolutely homogeneous, namely it satisfies

$$\|x + y\|_{PL} \leq \|x\|_{PL} + \|y\|_{PL} \qquad \text{and} \qquad \|\alpha x\|_{PL} = |\alpha| \|x\|_{PL}$$

for all $x, y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$.

norms. More precisely, aiming at MILP subproblems, we adopt a polyhedral norm for constructing $\|\cdot\|_{PL}$, deliberately excluding round norms such as the classical ℓ_2 -norm and the weighted norm $\|\cdot\|_M$ for some $M \succ 0$. Therefore, possible definitions for $\|\cdot\|_{PL}$ are $\|x\|_{PL} := \sum_{i\notin\mathcal{I}} |x_i|$ and $\|x\|_{PL} := \max_{i\notin\mathcal{I}} |x_i|$, respectively, for the ℓ_1- and ℓ_∞-norm.

2.2 Optimality concepts

Considering the problem of minimizing a continuous real function f over a closed set $\mathcal{X}$, it is clear what a global solution is: we call $\bar{x} \in \mathcal{X}$ a *global minimizer* for (P) if $f(\bar{x}) \leq f(x)$ for all $x \in \mathcal{X}$. In this context, however, the notion of *local* minimizer remains elusive, because of the delicate role played by neighborhoods, which can easily make *all* isolated points (of $\mathcal{X}$) local minimizers. For constructing a useful localized optimality concept, we consider the notion of neighborhoods with partial localization induced by the $\|\cdot\|_{PL}$, as adopted in (3).

Definition 1 A *PL-neighborhood* of a point $\bar{x} \in \mathcal{X}$ is a subset of $\mathbb{R}^n$ that contains an open PL-ball $\left\{w \in \mathbb{R}^n \mid \|w - \bar{x}\|_{PL} < \Delta\right\}$ for some $\Delta > 0$. A point $\bar{x} \in \mathcal{X}$ is called a *PL-isolated* point of $\mathcal{X}$ if there exists a PL-neighborhood of $\bar{x}$ that does not contain any other point of $\mathcal{X}$.

Notice that for PL-isolated points of $\mathbb{R}^n$, Definition 1 reduces to the classical notion of isolatedness. This gives rise to the following definition, consistent with the global counterpart.

Definition 2 (*minimizers*) A point $\bar{x} \in \mathcal{X}$ is called a *local minimizer* for (P) if there exists $\Delta > 0$ such that $f(\bar{x}) \leq f(x)$ for all $x \in \mathcal{X} \cap \mathbb{B}_{PL}(\bar{x}, \Delta)$. If this property holds for all $\Delta > 0$, then $\bar{x}$ is called a *global minimizer* for (P).

Seeking affordable algorithms to find candidate solutions for (P), we focus on strong (and yet practical) necessary conditions for such (local) optimality notion [12, Chapter 3]. According to [10, Sect. 3], $\bar{x}$ is *critical* if it is γ-critical for some $\gamma > 0$, i.e., $\bar{x} \in \operatorname{proj}_{\mathcal{X}}(\bar{x} - \gamma\nabla f(\bar{x}))$, and such notion is stronger than stationarity for nonconvex problems. Due to the difficulties discussed above in relation to round projections, we shall consider a notion based on (3) instead. Thus, we define the criticality measure Ψ associated to (P) as follows:

$$\forall x \in \mathcal{X},\ \forall \Delta \geq 0: \quad \Psi(x;\Delta) := \max\left\{\langle \nabla f(x), x - w\rangle \mid w \in \mathcal{X} \cap \mathbb{B}_{PL}(x, \Delta)\right\}. \tag{4}$$

Since $x \in \mathcal{X}$, the lower bound $\Psi(\cdot;\Delta) \geq 0$ holds for all $\Delta \geq 0$. Although analogous to the criticality measure of [7, 8], which focused on (P) without integrality constraints, in (4) Ψ depends additionally on the trust region radius Δ, which provides a localization—dispensable in the convex setting. Considering (global) minimizers

of the program obtained by linearization of f about $\bar{x}$ and localization with a trust region, (4) leads to the following definition.

Definition 3 (*criticality*) Given some $\Delta > 0$, a point $\bar{x} \in \mathcal{X}$ is called Δ-*critical* for (P) if $\Psi(\bar{x};\Delta) = 0$; it is called *critical* for (P) if it is Δ-critical for some $\Delta > 0$.

Remark that Definition 3 is well-posed: if there exists some $\bar{\Delta} > 0$ such that $\Psi(\bar{x};\bar{\Delta}) = 0$, namely $x = \bar{x}$ is optimal with $\bar{\Delta}$, then $\bar{x}$ is necessarily optimal for all $\Delta \in [0, \bar{\Delta}]$, for otherwise it could not be optimal for the relaxed constraint with $\bar{\Delta}$.

Definition 3 is closely related to stationarity in nonlinear optimization, interpreted as the absence of feasible first-order descent directions, but localized as criticality in (nonconvex) structured optimization [10]. The following foundational yet simple result states that criticality constitutes indeed a necessary first-order optimality condition for (P), in the sense of Definitions 2 and 3, respectively.

Proposition 1 *Let $\bar{x} \in \mathbb{R}^n$ be a local minimizer for (P). Then, $\bar{x} \in \mathcal{X}$ is critical.*

Proof Feasibility of $\bar{x}$ is readily inherited. Owing to Assumptions (A2)–(A3), optimality and criticality coincide with regard to the integer-valued variables. Then, the assertion is elementary if $\nabla f(\bar{x}) = 0$ or $\bar{x} \in \mathcal{X}$ is PL-isolated. Otherwise, continuous differentiability of $f : \mathcal{X} \to \mathbb{R}$ guarantees that no local improvements are possible (see, e.g., [12, Lemma 3.2]), concluding the proof. □

Proposition 1 portrays criticality as a necessary condition for (local and global) optimality, regardless of constraint qualifications. An analogous concept is that of B-stationarity: a point $\bar{x} \in \mathcal{X}$ is called *B-stationary* for (P) if $-\nabla f(\bar{x}) \in \mathcal{N}_{\mathcal{X}}^{F}(\bar{x})$, where $\mathcal{N}_{\mathcal{X}}^{F}(\bar{x})$ denotes the regular (or Fréchet) normal cone of $\mathcal{X}$ at $\bar{x}$ [13, Def. 6.38]. A relevant stationarity notion in nonlinear optimization, B-stationarity can be interpreted as the absence of feasible first-order descent directions, see [9]. Although B-stationarity and criticality coincide for (P) in the special case $\mathcal{I} = \emptyset$, the following illustrative example with complementarity constraints shows that, in general, B-stationarity $\not\Longrightarrow$ criticality. A more comprehensive characterization of criticality is left for future elaborations, possibly tailored to prominent problem classes, such as NLP with combinatorial structures.

Example 1 Consider (P) with $\mathcal{X} \subset \mathbb{R}^2 \times \mathbb{Z}$ described by

$$0 \le u_1 \le Uz, \quad 0 \le u_2 \le U(1-z), \quad z \in \{0,1\}$$

for some fixed $U > 0$. Notice that the feasible set $\mathcal{X}$ provides a lifted mixed-integer linear description of the nonconvex set $\mathcal{C} \subset \mathbb{R}^2$ obtained by intersecting the complementarity constraint $0 \le u_1 \perp u_2 \ge 0$ with the hyperbox $[0, U]^2$. The notation $u_1 \perp u_2$ here stands for the requirement $u_1 u_2 = 0$. The problematic point $\bar{u} = (0,0)$, in its ambient space $\mathbb{R}^2$, is B-stationary if and only if $-\nabla f_1(\bar{u}) \in \mathcal{N}_{\mathcal{C}}^{F}(\bar{u}) = (-\infty, 0]^2$.

Now we focus on $\bar{x}_0 = (\bar{u}, 0)$ and $\bar{x}_1 = (\bar{u}, 1)$, the only two points of $\mathcal{X}$ recovering $\bar{u}$ when projected onto the plane (u_1, u_2). Elaborating on Definition 3 we find that

the problematic points $\bar{x}_0$ and $\bar{x}_1$ are critical if and only if $-\nabla f(\bar{x}_0) \in (-\infty, 0]^3$ and $-\nabla f(\bar{x}_1) \in (-\infty, 0]^2 \times [0, \infty)$, respectively. If the objective f does not depend on the binary variable z (that is, if $f_2 = 0$), then the conditions for B-stationarity of $\bar{u}$ in $\mathcal{C}$ and those for criticality of $\bar{x}_0$ and $\bar{x}_1$ in $\mathcal{X}$ coincide. In contrast, if $f_2 \neq 0$, so that f explicitly depends on z, then criticality is strictly more restrictive than B-stationarity, which corresponds to $-\nabla f(\bar{x}_0) \in \mathbb{R} \times (-\infty, 0] \times \mathbb{R}$ and $-\nabla f(\bar{x}_1) \in (-\infty, 0] \times \mathbb{R}^2$.

Before developing and analyzing a numerical scheme for (P), we shall introduce an approximate counterpart of Definition 3. The following definition requires that a point $\bar{x} \in \mathcal{X}$ is not too far from being a minimizer for (3), as monitored using Ψ from (4).

Definition 4 (*ε-criticality*) Given some $\varepsilon > 0$ and $\Delta > 0$, a point $\bar{x} \in \mathcal{X}$ is called *ε-Δ-critical* for (P) if $\Psi(\bar{x};\Delta) \leq \varepsilon$; it is called *$\varepsilon$-critical* for (P) if it is ε-Δ-critical for some $\Delta > 0$.

For later reference we give the following characterization of noncritical points.

Lemma 2 *Let $\bar{x} \in \mathcal{X}$ be not critical for (P). Then, $\nabla f(\bar{x}) \neq 0$, $\bar{x}$ is not a PL-isolated point of $\mathcal{X}$ and, for all $\Delta > 0$, there exist $x \in \mathcal{X} \cap \mathbb{B}_{PL}(\bar{x}, \Delta)$ and $\varepsilon > 0$ such that $\langle \nabla f(\bar{x}), x - \bar{x} \rangle \leq -\varepsilon$. Moreover, for any given $\delta > 0$ the inequalities*

$$\Psi(\bar{x};\delta) \geq \Psi(\bar{x};\Delta) \geq \frac{\Delta}{\delta} \Psi(\bar{x};\delta) \tag{5}$$

hold for all $\Delta \in [0, \delta]$.

Proof If $\nabla f(\bar{x}) = 0$ or $\bar{x}$ is PL-isolated, then it follows from (4) that $\bar{x} \in \mathcal{X}$ must be critical according to Definition 3. This proves the first two assertions. The third is obtained similarly, by negating the criticality property for $\bar{x}$.

Let us consider the sandwich inequalities in (5). Pick an arbitrary but fixed $\delta > 0$ and let $\Delta \in (0, \delta]$. The remaining case $\Delta = 0$ is immediate since $\Psi(\bar{x};0) = 0$ and $\Psi(\cdot;\cdot) \geq 0$. Since $\Delta \leq \delta$, the left inequality directly follows from the definition in (4). Before addressing the inequality on the right, we rewrite (4) in the convenient form

$$\forall x \in \mathcal{X},\ \forall \Delta \geq 0 : \quad \Psi(x;\Delta) := \max \left\{ -\langle \nabla f(x), d \rangle \mid x + d \in \mathcal{X},\ \|d\|_{PL} \leq \Delta \right\}$$

and we denote by d_Δ a (global) maximizer in this definition of $\Psi(x;\Delta)$, for any given $\Delta \geq 0$. Then, by construction of d_δ, it must be $\|d_\delta\|_{PL} \leq \delta$ and, owing to homogeneity of $\|\cdot\|_{PL}$, we obtain the bound $\|d_\delta \Delta/\delta\|_{PL} \leq \Delta$. Finally, together with (4), this bound implies the following inequality

$$\Psi(\bar{x};\Delta) = -\langle \nabla f(\bar{x}), d_\Delta \rangle \geq -\langle \nabla f(\bar{x}), d_\delta \Delta/\delta \rangle = -\frac{\Delta}{\delta} \langle \nabla f(\bar{x}), d_\delta \rangle = \frac{\Delta}{\delta} \Psi(\bar{x};\delta),$$

concluding the proof. □

In contrast to [8, Lemma 3.1], which is valid for problems with convex constraints, the noncriticality assumption in Lemma 2 is essential for the sandwich inequalities (5) in our context. Counterexamples are readily obtained by taking a point $\bar{x}$ that is Δ-critical but not δ-critical for some $\delta > \Delta > 0$.

3 Sequential mixed-integer linearization method

We now introduce our iterative numerical scheme based on local mixed-integer linear approximations. Then, we investigate its behavior and show the convergence to critical points under mild assumptions.

3.1 Algorithm

The proposed method, detailed in Algorithm 1, builds upon a sequence of MILP subproblems of the form (3). After solving the trust region linearized subproblem at (3), which guarantees that $x^{k+1} \in \mathcal{X} \cap \mathbb{B}_{PL}(x^k, \Delta_k)$ and

$$\Psi(x^k;\Delta_k) = \Psi_k := \langle \nabla f(x^k), x^k - x^{k+1} \rangle \geq 0, \tag{6}$$

the quality of the tentative update x^{k+1} from x^k is tested. In particular, we monitor criticality and terminate at Step 5 if (approximately) satisfied. Then, a sufficient decrease condition requires at Step 6 that $f(x^{k+1}) \leq m_k - \varrho\Psi(x^k;\Delta_k)$ holds for the update x^{k+1} to be accepted; otherwise, similarly to a backtracking line search procedure, the trust region radius Δ_k is reduced and the attempt discarded. With the lens of trust region methods, Step 6 compares the predicted linearized improvement against the actual change, with the ratio parameter $\varrho > 0$ acting as threshold for acceptance. In fact, the actual change a_k at Step 4 refers to a merit value m_k which coincides with the cost only if $\kappa_m = 1$, owing to Step 8. For $\kappa_m \in (0, 1)$ the acceptance criterion is less conservative and potentially allows larger radii and faster convergence in practice, with the objective $f(x^k)$ decreasing nonmonotonically; see [4]. Finally, the radius is updated at Step 9, where one can apply classical trust region rules or a line search-like reset, based on $\rho_k := a_k/\Psi_k$, such as

$$\Delta_{k+1} := \begin{cases} \kappa\Delta_k & \text{if } \rho_k < \varrho_1, \\ \Delta_k & \text{if } \varrho_1 \leq \rho_k < \varrho_2, \\ \Delta_k/\kappa & \text{if } \varrho_2 \leq \rho_k \end{cases} \qquad \text{or} \qquad \Delta_{k+1} \in [\Delta_{\min}, \Delta_{\max}], \tag{7}$$

respectively, given parameters such that $\varrho \leq \varrho_1 < \varrho_2 < 1$ and $0 < \Delta_{\min} \leq \Delta_{\max}$.

Algorithm 1 Successive mixed-integer linearization algorithm for (P)

Input: $x^0 \in \mathcal{X}$, $\varepsilon > 0$
Data: $\Delta_0 > 0$, $\varrho, \kappa \in (0,1)$, $\kappa_m \in (0,1]$
1 set $m_0 \leftarrow f(x^0)$
2 **for** $k = 0, 1, 2 \ldots$ **do**
3 compute $x^{k+1} \in \arg\min \left\{ \langle \nabla f(x^k), x \rangle \,|\, x \in \mathcal{X} \cap \mathbb{B}_{\mathrm{PL}}(x^k, \Delta_k) \right\}$
4 set $a_k \leftarrow m_k - f(x^{k+1})$ and $\Psi_k \leftarrow \langle \nabla f(x^k), x^k - x^{k+1} \rangle$
5 **if** $\Psi_k \leq \varepsilon$ **then return** x^k // ε-Δ_k-critical
6 **if** $a_k < \varrho \Psi_k$ **then**
7 set $\Delta_k \leftarrow \kappa \Delta_k$ and **go to** Step 3
8 set $m_{k+1} \leftarrow (1 - \kappa_m) m_k + \kappa_m f(x^{k+1})$
9 select Δ_{k+1} by trust region update (7)

We note that the input requirement in Algorithm 1 of a feasible initial point is not a restriction: given some $x^0 \notin \mathcal{X}$, it is enough to project it onto the feasible set by solving a MILP, for instance $\min_{x \in \mathcal{X}} \|x - x^0\|_1$, and run the algorithm from a solution thereof.

Remark 1 An optional refinement step can also be integrated in Algorithm 1, similarly to magic steps [12, Sect. 8.2], equality-constrained quadratic programming (EQP) steps of [8], and bound-constrained quadratic programming (BQP) steps of [9]. One can also consider NLP steps, by fixing the integer-valued variables and solving an NLP with polyhedral constraints [1, 3]. Without impairing the convergence properties, this optional step can improve the objective value while maintaining feasibility, possibly exploiting additional smoothness of f, as discussed in Sect. 4 below.

Finally, the termination criterion at Step 5 is based on the approximate criticality notion of Definition 4, and supported by the following convergence analysis, which guarantees finite termination for any $\varepsilon > 0$ when f is bounded from below over $\mathcal{X}$.

3.2 Convergence analysis

Assumption (A1) on the smoothness of f is sufficient to prove that the backtracking due to Step 7 eventually stops. In particular, the condition at Step 6 is violated for a sufficiently small Δ_k, after finitely many attempts. Consequently, Algorithm 1 is well-defined and produces a sequence of feasible iterates $\{x^k\}$ along with decreasing merit values $\{m_k\}$; cf. Lemma 5. Then, it remains to prove that the accumulation points are also critical. After the asymptotic characterization in Lemma 6, we show that, in a neighborhood of a feasible but noncritical point, the subproblem at Step 3 eventually generates an acceptable step while the trust region radius remains

bounded away from zero. Finally, Theorem 1 synthesizes the main convergence results.

We start with a preliminary result about the reduction implied by Step 3, which follows from Taylor's theorem. Notice how Assumption (A2) allows to drop the integer-valued variables from the quadratic term of the expansion.

Lemma 3 *Consider a function $f : \mathcal{X} \to \mathbb{R}$, a nonempty compact set $\Omega \subseteq \mathrm{dom} f$, and suppose Assumptions (A1)–(A3) hold. Partitioning $x = (u, z)$ with u and z the real- and integer-valued variables, respectively, define*

$$L := \sup \left\{ \frac{\|\nabla f(x_1) - \nabla f(x_2)\|_1}{\|u_1 - u_2\|_\infty} \mid \begin{array}{l} x_1, x_2 \in \Omega, \\ u_1 \neq u_2 \end{array} \right\}.$$

Then $L < \infty$, and for all $x_1, x_2 \in \Omega$ the following upper bound is valid:

$$f(x_2) \leq f(x_1) + \langle \nabla f(x_1), x_2 - x_1 \rangle + \frac{L}{2} \|u_2 - u_1\|_\infty^2.$$

Proof By Assumption (A2) it is $\nabla f(x) = (\nabla f_1(u), f_2)$, hence $L < \infty$ coincides with the Lipschitz constant of ∇f over Ω, finite by Assumption (A1). Then, the asserted inequality follows from Taylor-type arguments, such as in the descent lemma [13, Lemma 2.64], by considering the variation from x_1 to x_2 along real- and integer-valued components separately. We sketch the second part of the proof as follows. Define $\widehat{x} := (u_2, z_1)$ and, for simplicity, assume $\widehat{x} \in \Omega$. Then, Assumption (A2) and Taylor's theorem, e.g. [9, Lemma 4.3], yield the inequality

$$\begin{aligned} f(x_2) &= f(\widehat{x}) + \langle \nabla f(\widehat{x}), x_2 - \widehat{x} \rangle \\ &\leq f(x_1) + \langle \nabla f(x_1), \widehat{x} - x_1 \rangle + \frac{L}{2} \|\widehat{x} - x_1\|_\infty^2 + \langle \nabla f(\widehat{x}), x_2 - \widehat{x} \rangle. \end{aligned}$$

Finally, exploiting the structure of ∇f and the definition of $\widehat{x}$, one recognizes the asserted upper bound. □

Next, we employ Lemma 3 to prove that the iterations of Algorithm 1 are well-defined.

Lemma 4 (well-definedness) *Consider (P) under Assumptions (A1)–(A3). Then the condition at Step 6 of Algorithm 1 is violated after a finite number of steps.*

Proof If x^k is critical, then it is $\bar{\Delta}$-critical for some $\bar{\Delta} > 0$, and $x = x^k$ solves the subproblem at Step 3 as soon as $\Delta_k \leq \bar{\Delta}$. Regardless of the choice $x^{k+1} \in \mathcal{X}$, it must yield $\Psi_k = 0$ and immediate termination of the algorithm, for any $\varepsilon > 0$.

If x^k is not critical, then we rely on the sandwich inequalities (5) given by Lemma 2. The claim will follow from the fact that $m_0 = f(x^0)$ and, for each $k \in \mathbb{N}$, the acceptance according to Step 6 and the update rule at Step 8 imply that

$$\begin{aligned} m_{k+1} &:= (1-\kappa_m)m_k + \kappa_m f(x^{k+1}) \\ &\geq (1-\kappa_m)\left[f(x^{k+1}) + \varrho\Psi_k\right] + \kappa_m f(x^{k+1}) \geq f(x^{k+1}), \end{aligned} \tag{8}$$

proving that the nonmonotone acceptance criterion is less conservative than the monotone one (which has $\kappa_m = 1$ and $m_k = f(x^k)$ for all $k \in \mathbb{N}$). Moreover, all attempts satisfy $x^{k+1} \in \mathcal{X}$ and $\|x^{k+1} - x^k\|_{PL} \leq \Delta_k$. In particular, all attempts x^{k+1} remain in a bounded set, owing to Assumption (A3) and to the trust region stipulation, so there exists a constant $L_k > 0$ related to ∇f over that set, in the sense of Lemma 3. Then, all attempts x^{k+1} satisfy the upper bound

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L_k}{2}\|u^{k+1} - u^k\|_\infty^2. \tag{9}$$

Combined with (8) and then (6), (9) yields

$$\begin{aligned} \rho_k &:= \frac{a_k}{\Psi_k} = \frac{m_k - f(x^{k+1})}{\Psi_k} \geq \frac{f(x^k) - f(x^{k+1})}{\Psi_k} \geq \frac{\Psi_k - \frac{L_k}{2}\|u^{k+1}-u^k\|_\infty^2}{\Psi_k} \\ &= 1 - \frac{L_k}{2}\frac{\|u^{k+1}-u^k\|_\infty^2}{\Psi_k} \geq 1 - \frac{L_k\delta_k}{2}\frac{\|u^{k+1}-u^k\|_\infty^2}{\Delta_k\Psi(x^k;\delta_k)}, \end{aligned} \tag{10}$$

where the last inequality follows from the lower bound in (5), for any fixed, sufficiently large $\delta_k > 0$ and the associated $\Psi(x^k;\delta_k) > 0$. Due to the equivalence of norms in $\mathbb{R}^n$, there exists a constant $c_{PL} > 0$ relating $\|x\|_{PL}$ and $\|u\|_\infty$, providing a bound in the form $\|u^{k+1} - u^k\|_\infty \leq c_{PL}\|x^{k+1} - x^k\|_{PL}$. Therefore, from (10) we have that

$$2(1-\rho_k)\frac{\Psi(x^k;\delta_k)}{L_k\delta_k} \leq \frac{\|u^{k+1} - u^k\|_\infty^2}{\Delta_k} \leq \frac{c_{PL}^2\|x^{k+1} - x^k\|_{PL}^2}{\Delta_k} \leq c_{PL}^2\Delta_k,$$

where the last step uses the trust region stipulation with radius Δ_k due to Step 3 of Algorithm 1. Since x^k, L_k and δ_k are fixed while Δ_k is decreased, the condition $\rho_k \geq \varrho$ becomes satisfied as soon as

$$\Delta_k \leq 2\frac{1-\varrho}{c_{PL}^2}\frac{\Psi(x^k;\delta_k)}{L_k\delta_k}, \tag{11}$$

violating the condition at Step 6, after finitely many backtracks since $\kappa \in (0,1)$. □

Lemma 5 (*descent behavior*) *Consider* (*P*) *under Assumptions* (*A*1)–(*A*3) *and the iterates generated by Algorithm* 1. *Then*,

(i) *the sequence* $\{m_k\}$ *is monotonically decreasing and, for every* $k \in \mathbb{N}$, *one has* $\Psi_k \geq 0$ *and*

$$f(x^{k+1}) + (1-\kappa_m)\varrho\Psi_k \leq m_{k+1} \leq m_k - \kappa_m\varrho\Psi_k. \tag{12}$$

(ii) *Every iterate* x^k *is feasible, namely* $x^k \in \mathcal{X}$, *and satisfies* $f(x^k) \leq f(x^0)$.

Proof Based on Lemma 4, the iterates of Algorithm 1 are well-defined.

For (i), by the update rule at Step 8 and the condition at Step 6, we have that

$$m_{k+1} := (1-\kappa_m)m_k + \kappa_m f(x^{k+1}) \leq (1-\kappa_m)m_k + \kappa_m\left[m_k - \varrho\Psi_k\right] = m_k - \kappa_m\varrho\Psi_k$$

holds upon acceptance. The lower bound on m_{k+1} is readily obtained from (8). Then, it follows from (6) that $\Psi_k \geq 0$ for each $k \in \mathbb{N}$.

Let us consider (ii) now. Owing to $x^0 \in \mathcal{X}$ and Step 3, every iterate x^k is feasible by construction. Then, since $\kappa_m \in (0,1]$ and $\varrho\Psi_k \geq 0$, it follows from $m_0 := f(x^0)$ and (i) that $f(x^k) \leq m_k \leq \cdots \leq m_0 := f(x^0) < \infty$. □

Lemma 6 *Consider* (*P*) *under Assumptions* (*A*1)–(*A*3) *and the iterates generated by Algorithm* 1. *Suppose* $\inf_{\mathcal{X}} f > -\infty$. *Then*,

(i) $\{f(x^k)\}$ *and* $\{m_k\}$ *converge*, *the latter from above*, *to a finite value* $f_\star \geq \inf_{\mathcal{X}} f$;
(ii) $\sum_{k\in\mathbb{N}} \Psi_k < \infty$;
(iii) $\lim_{k\to\infty} \Psi_k = 0$.

Proof Based on Lemma 4, the iterates of Algorithm 1 are well-defined and we may assume the sequence $\{x^k\}$ is infinite (it suffices to skip Step 5).

The assertion in (i) regarding $\{m_k\}$ follows from (12) with the premises. Then, considering the update rule at Step 8,

$$\kappa_m f(x^{k+1}) = m_{k+1} - (1-\kappa_m)m_k = \kappa_m m_k + (m_{k+1} - m_k),$$

the convergence of $\{m_k\}$ implies that of $\{f(x^k)\}$ to the same value $f_\star$, since $\kappa_m \in (0,1]$.

Regarding (ii), a telescoping argument on (12), together with (i), yields

$$m_0 - f_\star \geq \sum_{k=0}^{j} m_k - m_{k+1} \geq \kappa_m\varrho \sum_{k=0}^{j} \Psi_k \geq 0.$$

The finite sum follows from the independence of the (finite) upper bound $m_0 - f_\star$ on $j \in \mathbb{N}$. Finally, assertion (iii) follows from (ii), since $\Psi_k \geq 0$ by Lemma 5. □

Owing to (6), Lemma 6(iii) gives that $\{\Psi(x^k;\Delta_k)\} \to 0$ if $\inf_{\mathcal{X}} f > -\infty$. Therefore, if $\{\Delta_k\}$ remains bounded away from zero, we expect accumulation points of $\{x^k\}$ to be critical. Indeed, for any noncritical point $x^\infty \in \mathcal{X}$ and sequence $\{x^k\}$ such that $x^k \to x^\infty$, the trust region radius $\{\Delta_k\}$ can be bounded from below by some positive constant. In the following result, inspired by [9, Lemma 4.5], by *relative PL-neighborhood* we mean the intersection of a PL-neighborhood of x^∞ in $\mathbb{R}^n$ with the feasible set $\mathcal{X}$.

Lemma 7 *Consider* (*P*) *under Assumptions* (*A*1)–(*A*3). *Let a point* $x^\infty \in \mathcal{X}$ *be not critical for* (*P*). *Then*, *there exists a nonsingleton relative PL-neighborhood* N^∞ *of*

x^∞ such that for any sequence $\{x^k\} \subset N^\infty$ with $x^k \to x^\infty$ the steps of Algorithm 1 generate a sequence $\{\Delta_k\}$ bounded away from zero.

Proof Owing to noncriticality and Lemma 2, x^∞ is not a PL-isolated point of $\mathcal{X}$, hence there exists a nonempty nonsingleton relative PL-neighborhood N^∞ of x^∞. Moreover, the sandwich inequalities (5) hold for any fixed $\delta_\infty > 0$ and the associated $\Psi(x^\infty;\delta_\infty) > 0$. Denote L_∞ the Lipschitz constant of ∇f over N^∞, finite by Assumption (A1). Since the trust region radius is reduced when needed for sufficient decrease, and not otherwise, Δ_k is not reduced below the uniform bound given by $\kappa\Delta_\infty$, where

$$\Delta_\infty := 2\frac{1-\varrho}{c_{PL}^2}\frac{\Psi(x^\infty;\delta_\infty)}{L_\infty\delta_\infty} > 0$$

is the worst-case threshold for acceptance at Step 6; see (11). Analogous arguments apply with the reset rule for Δ_{k+1} in (7), concluding the proof. □

By virtue of Lemma 7, the trust region radius does not vanish in the vicinity of noncritical points, hence sufficiently small steps can always be accepted there, which deliver sufficient reduction and improvement. We are now in a position to prove our main convergence result, connecting accumulation points and criticality. The following Theorem 1 is akin to [8, Thm 3.8] and [9, Thm 4.2].

Theorem 1 *Consider (P) under Assumptions (A1)–(A3) and let $x^0 \in \mathcal{X}$ be an arbitrary but fixed point. Then, relative to the iterates generated by Algorithm 1 with $\varepsilon = 0$, one of the following mutually exclusive outcomes must occur:*

(i) $\lim_{k\to\infty} f(x^k) = -\infty$;
(ii) *Algorithm 1 terminates at a critical point, that is, x^k solves the subproblem at Step 3 and $\Psi_k = 0$ for some $k \in \mathbb{N}$;*
(iii) *Algorithm 1 generates an infinite sequence of iterates $\{x^k\}$ with decreasing merit values $\{m_k\}$. If $\{x^k\}$ has an accumulation point x^∞, then x^∞ is feasible and critical for (P).*

Proof We need to consider only outcome (iii) because in the other cases we either observe (i) lower unboundedness or (ii) obtain a critical point by virtue of the termination condition.

Since the backtracking terminates finitely by Lemma 4, and each attempt is feasible for the subproblem at Step 3, we obtain by induction over the iterations that $x^k \in \mathcal{X}$ for all $k \in \mathbb{N}$. Furthermore, by Lemma 5 all iterations yield an improvement in terms of merit—and the objective value eventually decreases as well, owing to Lemma 6. It remains to show that every accumulation point of $\{x^k\}$ is critical.

Seeking a contradiction, we assume that an accumulation point x^∞ of the sequence $\{x^k\}$ is not critical. Let us denote $\{x^k\}_{k\in K}$ a subsequence such that $x^k \to_K x^\infty$ and,

possibly relabeling, restrict $\{x^k\}_{k\in K}$ to the PL-neighborhood N^∞ defined in Lemma 7, so that $x^k \in N^\infty$ for all $k \in K$.

Now we show that $\{\Psi_k\}_{k\in K}$ remains bounded away from zero. First of all, by noncriticality of x^∞, Lemma 2 gives that $\langle \nabla f(x^\infty), x^\infty - x\rangle \geq \varepsilon_0$ for some $x \in \mathcal{X}$ and $\varepsilon_0 > 0$. Then, recall the continuity of ∇f, the minimality property of x^{k+1} for $\langle \nabla f(x^k), \cdot - x^k\rangle$ within a radius Δ_k (bounded away from zero by Lemma 7), and the convergence $x^k \to x^\infty$. These facts imply the existence of some $\varepsilon_\infty > 0$ such that $\Psi_k \geq \varepsilon_\infty$ holds for all $k \in K$. Combined with (12), and for any $k_0 \in K$, the boundedness of $\{\Psi_k\}_{k\in K}$ away from zero results in

$$f(x^k) \leq m_k = m_{k_0} + \sum_{j=k_0}^{k-1}(m_{j+1} - m_j) \leq m_{k_0} - \sum_{j=k_0}^{k-1} \kappa_m \varrho \Psi_j$$
$$\leq m_{k_0} - \sum_{j=k_0}^{k-1} \kappa_m \varrho \varepsilon_\infty = m_{k_0} - (k - k_0)\kappa_m \varrho \varepsilon_\infty \to -\infty$$

as $k \to_K \infty$. This fact contradicts the (subsequential) convergence $f(x^k) \to_K f(x^\infty)$ by $x^k \to_K x^\infty$, thus showing that every accumulation point x^∞ of $\{x^k\}$ is critical. □

Finally, as a direct corollary to Theorem 1, finite termination with an ε-critical point can be established for any $\varepsilon > 0$, if $\inf_{\mathcal{X}} f > -\infty$, via Lemma 6.

4 Numerical examples

We report on numerical results for two example problems: the discrete-time optimal control of a hybrid system with hysteresis and the concurrent design and operation planning of a processing network.

To exploit the problem structure and the available technology, we consider also a variant of Algorithm 1 with the NLP refinement step suggested in Remark 1. In practice, the real-valued variables are perfected by solving the NLP obtained from (P) by fixing the integer-valued variables at their current value. To avoid solving many NLPs, this procedure is invoked only when the integer-valued variables remain untouched by a successful update. More precisely, the NLP refinement may take place before executing Step 8, only if $z^{k+1} = z^k$. An improved u^{k+1} is obtained by solving (up to ε-stationarity) the problem arising from (P) with the additional constraint $z = z^k$.

The numerical results are generated with a prototype MATLAB implementation of Algorithm 1, adopting `intlinprog` for the MILP subproblems arising at Step 3 and `fmincon` for the optional NLP subproblems.[2] The default algorithmic parameters are set as follows: tolerance $\varepsilon = 10^{-8}$, initial radius $\Delta_0 = 1$, sufficient

[2] The prototype implementation uses default options except the following: integer tolerance 10^{-6}, absolute gap, relative gap and constraint tolerances 10^{-8} for `intlinprog`; algorithm `sqp`, objective gradient provided, max iterations 1000, optimality and step tolerance ε, and constraint tolerance 10^{-8} for `fmincon`. Code run in MATLAB R2022b, update 10.

decrease parameter $\varrho = 1/10$, reduction factor $\kappa = 1/2$, and monotonicity parameter $\kappa_m = 1/2$. Step 9 uses the trust region update rule (7) with $\varrho_1 = \varrho$ and $\varrho_2 = 2\varrho$. The partial localization $\|\cdot\|_{PL}$ uses the ℓ_∞-norm, namely $\|x\|_{PL} := \max_{i \notin \mathcal{I}} |x_i|$. If an infeasible starting point x^0 is provided, a feasible one is computed by solving minimize $_{x \in \mathcal{X}} \|x - x^0\|_1$, tackled as

$$\underset{x \in \mathcal{X}, d_+, d_- \in \mathbb{R}^n}{\text{minimize}} \quad \langle 1, d_+ + d_- \rangle \qquad \text{subject to} \quad d_+, d_- \geq 0, \quad x - x_0 = d_+ - d_-.$$

Moreover, the execution of Algorithm 1 is stopped if the MILP solver fails, either by declaring the subproblem at Step 3 infeasible or resulting in a negative criticality measure Ψ_k at Step 4. These occurrences are both due to numerical issues within the MILP solver; in fact, all subproblems are guaranteed to be feasible and the criticality measure to be nonnegative, by construction, with *exact* MILP solves.

All source codes used to generate and analyze the results in Sect. 4 have been archived on Zenodo at DOI: 10.5281/zenodo.10007957.

4.1 Car with hysteretic turbo charger

In this section we apply Algorithm 1 to a numerical example for the point-to-point optimal control problem of a car with turbo charger. Inspired by [14, Sect. V], we consider a double-integrator point mass model equipped with a turbo accelerator subject to hysteresis effects.

The car is described by its position $q(t)$, velocity $v(t)$ and turbo state $w(t) \in \{0, 1\}$ for each time $t \in [0, T]$, $T > 0$. The control variables are the acceleration $a(t)$ and brake $b(t)$ pedals. The turbo is activated when the velocity exceeds $v \geq v_+ := 10$ and is deactivated when it falls below $v \leq v_- := 5$; when it is on, it makes the nominal thrust three times more effective. In contrast, the braking force remains unaffected by the turbo state. Summarizing, the state vector reads as $(q, v, w) \in \mathbb{R}^3$ with dynamics $\dot{q} = v$, $\dot{v} = f - b$, where the thrust f has two modes of operation, depending on the turbo state, described by $f(a) = a$ if $w = 0$ and $f(a) = 3a$ if $w = 1$. The turbo state w behaves according to the hysteresis characteristic described above. The acceleration control is bounded by $0 \leq a \leq a_{\max}$, brake control by $0 \leq b \leq b_{\max}$, and the velocity by $|v| \leq v_{\max}$. The initial state is $q(0) = v(0) = w(0) = 0$. Given a final time $T > 0$ and position $q_{\text{end}} > 0$ and parameters $\alpha_a, \alpha_b \geq 0$, we are interested in minimizing the control effort $\alpha_a \int_0^T a^2(t)\mathrm{d}t + \alpha_b \int_0^T b^3(t)\mathrm{d}t$ to achieve $q(T) = q_{\text{end}}$ and $v(T) = 0$.

By direct discretization one can formulate this hybrid optimal control problem in the form (P), by rewriting logical propositions as big-M constraints. Let us consider a time grid $0 = t_0 < \ldots < t_N = T$, with N intervals of size $h = T/N$, and state and control approximations, denoted $q_k, v_k, \ldots, b_k$, such that $q_k \approx q(t_k)$ for each $k = 0, \ldots, N$. Auxiliary variables f_k for the thrust f are included for clarity. Then, approximated via the trapezoidal rule, the discretized integral objective is

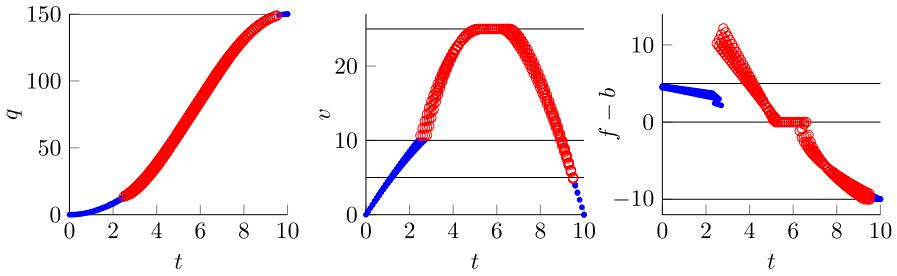

Fig. 1 Solutions for the problem of a turbo car with $N = 100$ discretization intervals: superimposed states and control trajectories of all test runs, with thresholds (black solid lines) and different markers when the turbo is active (red circle) or inactive (blue dot) (color figure online)

$$\alpha_a h \sum_{k=0}^{N-1} \frac{a_k^2 + a_{k+1}^2}{2} + \alpha_b h \sum_{k=0}^{N-1} \frac{b_k^3 + b_{k+1}^3}{2}. \tag{13}$$

With finite differences, the discrete-time dynamics read, for $k = 0, \dots, N-1$,

$$\frac{q_{k+1} - q_k}{h} = \frac{v_{k+1} + v_k}{2}, \quad \frac{v_{k+1} - v_k}{h} = \frac{f_{k+1} + f_k}{2} - \frac{b_{k+1} + b_k}{2}. \tag{14}$$

Here, the thrust f_k can be encoded by a linear big-M model, instead of *if-else* cases, as

$$\begin{aligned} f_k - a_k &\le M w_k, & f_k - a_k &\ge -M w_k, \\ f_k - 3a_k &\le M(1 - w_k), & f_k - 3a_k &\ge -M(1 - w_k) \end{aligned} \tag{15}$$

where $M > 0$ is fixed and large enough. The hysteresis characteristic corresponds to logical propositions describing activation and deactivation conditions, in the form

$$\begin{aligned} w_k = 0 \ \wedge \ v_k > v_+ \ &\Rightarrow \ w_{k+1} = 1, \quad w_k = 1 \ \wedge \ v_k < v_- \ \Rightarrow \ w_{k+1} = 0, \\ w_k = 0 \ \wedge \ v_k < v_+ \ &\Rightarrow \ w_{k+1} = 0, \quad w_k = 1 \ \wedge \ v_k > v_- \ \Rightarrow \ w_{k+1} = 1 \end{aligned}$$

for $k = 0, \dots, N-1$, with $\vee$ denoting the logical "and". Rewriting as clauses and using De Morgan's laws, these can be cast as algebraic big-M constraints, reading

$$\begin{aligned} v_k &\le v_+ + M(w_k + w_{k+1}), & v_k &\ge v_- - M(2 - w_k - w_{k+1}), \\ v_k &\ge v_+ - M(w_k + 1 - w_{k+1}), & v_k &\le v_- + M(1 - w_k + w_{k+1}). \end{aligned} \tag{16}$$

Finally, bounds are imposed only on the time grid, analogously to boundary conditions, as

$$0 \le a_k \le a_{\max}, \quad 0 \le b_k \le b_{\max}, -v_{\max} \le v_k \le v_{\max} \quad \text{for } k = 0, \dots, N, \tag{17}$$

$$q_0 = v_0 = w_0 = 0, \quad q_N = q_{\text{end}}, v_N = 0. \tag{18}$$

Table 1 Summary of results for the problem of a turbo car with $N \in \{25, 50, 100\}$ discretization intervals

		Min	q(25%)	Median	q(75%)	Max
$N = 25$	Objective value	71.20	71.20	71.20	71.20	74.35
	# Iterations	59	61	63	64	69
	# MILPs	125	130	131	133	141
	Runtime [s]	0.44	0.58	0.62	0.67	0.91
Initial projection	Runtime [s]	0.06	0.13	0.15	0.18	0.29
$N = 50$	Objective value	69.46	69.80	69.80	69.80	71.74
	# Iterations	60	65	66	68	84
	# MILPs	128	134	136	137	163
	Runtime [s]	1.01	1.30	1.37	1.44	1.93
Initial projection	Runtime [s]	0.26	0.56	0.66	0.76	1.71
$N = 100$	Objective value	68.66	68.95	69.6	69.6	69.6
	# Iterations	66	68	70	71	73
	# MILPs	134	136	137	139	141
	Runtime [s]	2.84	3.26	3.43	3.53	4.34
Initial projection	Runtime [s]	1.40	3.03	3.82	7.49	124.08

Statistics (minimum, maximum, and quartiles over different initial guesses) for the final objective value, runtime, and number of iterations and MILP subproblems. Algorithm 1's runtime does not include the polyhedral projection of infeasible initial guesses, provided for comparison

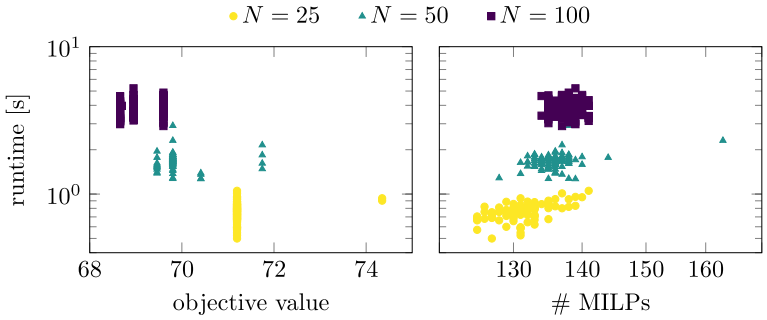

Fig. 2 Comparison for the problem of a turbo car with $N \in \{25, 50, 100\}$ discretization intervals. Objective value, runtime and number of MILP subproblems for all test runs

Overall, a discretization with N intervals results in a problem (P) according to (13)–(16) with $5(N + 1)$ real-valued and $(N + 1)$ binary-valued decision variables, $2N$ linear equality constraints, $4(N + 1) + 4N$ linear inequality constraints, and simple bounds.

Simulations We consider instances with $N \in \{25, 50, 100\}$ discretization intervals and parameters $\alpha_a = 1$, $\alpha_b = 10^{-2}$, $q_{\text{end}} = 150$, $T = 10$, $a_{\max} = 5$, $b_{\max} = 10$, $v_{\max} = 25$, and $M = 20$. For each instance we run Algorithm 1 starting from 100

initial guesses obtained by sampling a normal distribution with zero mean and standard deviation 10.

Algorithm 1 successfully returned in all cases with a critical point (within the specified tolerance). Solutions are depicted as trajectories in Fig. 1 and computational results are summarized in Table 1. Out of 100 runs, at most 4 different solutions are returned by Algorithm 1 for each instance. These have essentially the same pattern but slightly perturbed turbo activation point, as shown by Fig. 1 for $N = 100$. Moreover, Table 1 highlights that not only all solutions have very similar objective value, but also that the solution process is relatively consistent. In particular, the number of iterations and MILP subproblems are in narrow ranges, independent on the discretization level, whereas the runtime increases with finer discretizations, mostly because MILP subproblems become larger in size. These observations are illustrated in Fig. 2, which also shows that lower objective values are attained with finer discretizations. Overall, despite the problem size (up to $n = 606$ when $N = 100$) and our straightforward implementation without safeguards for numerical errors, Algorithm 1 was able to achieve accurate results in a few seconds.

4.2 Processing network design and operation

In this section we consider the synthesis of a processing network that was originally formulated in [15, Example 2], see also [2]. The problem entails the concurrent design and operation planning of a networked system, that is, nodes and flows along the edges are jointly optimized. Figure 3 shows the superstructure which involves the possible selection of 8 interconnected processing units. With this illustrative example we demonstrate the potential benefits of including NLP refinement steps in Algorithm 1 as well as its performance compared to a brute-force combinatorial enumeration coupled with NLP solves.

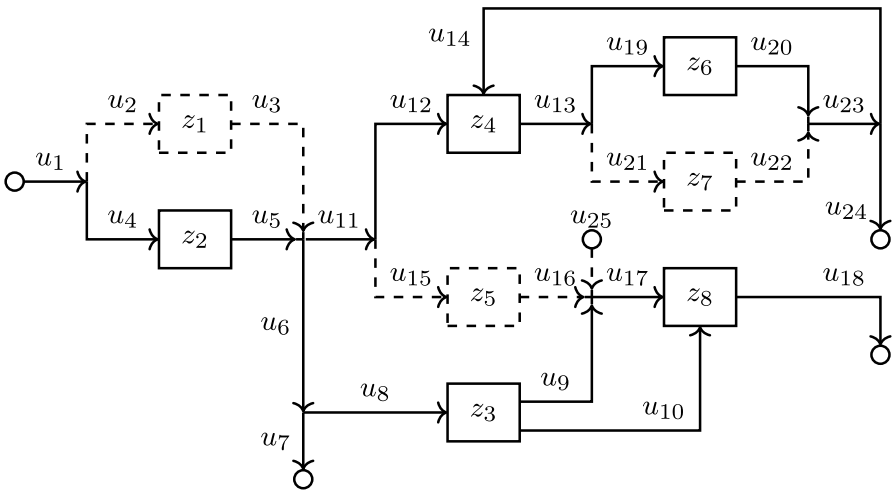

Fig. 3 Superstructure for the processing network example: units (boxes) and edges (lines) selected (solid) and discarded (dashed) by the optimal design

Table 2 Summary of computational results for the problem of a processing network.

		Min	q(25%)	Median	q(75%)	Max
Without NLP	Objective value	76.92	80.31	90.62	103.60	116.39
Refinement	# Iterations	186	$1000^{\bowtie}$	$1000^{\bowtie}$	$1000^{\bowtie}$	$1000^{\bowtie}$
	# MILPs	353	1882	1892.5	1907.5	1933
	Runtime [s]	1.27	6.15	6.19	6.24	6.47
With NLP	Objective value	59.85	72.09	83.75	95.65	107.56
Refinement	# Iterations	2	12.5	18	22	29
	# MILPs	6	32.5	45.5	54	73
	# NLPs	1	12	16	20.5	28
	Runtime [s]	0.05	0.23	0.31	0.39	0.76
Fixed-integer NLP	Objective value	59.85	72.09	83.70	95.65	107.56
	Runtime [s]	0.02	0.02	0.03	0.03	0.04
Initial projection	Runtime [s]	0.01	0.01	0.01	0.01	0.09

Statistics (minimum, maximum, and quartiles over different initial guesses) for the final objective value, runtime, and number of iterations, MILP and NLP subproblems. Algorithm 1's runtime includes NLP solves but not the polyhedral projection of infeasible initial guesses, provided for comparison

The symbol $^{\bowtie}$ stands for a limit reached

The original model consists of 25 real-valued u and 8 Boolean variables Z to model flows on the lines and the existence of processing units. The objective function includes costs for the selected processing units, operating costs and revenues from sales of products. The formulation involves material balance and flow constraints, logical propositions, and disjunctions for consistent physical modeling. Boolean specifications are transformed into linear algebraic constraints on binary variables z. For instance, the proposition $Z_3 \Rightarrow Z_1 \vee Z_2$ is rewritten as the clause $Z_1 \vee Z_2 \vee \neg Z_3$ and then transformed into $z_1 + z_2 \geq z_3$. Specifications such as $Z_1 \veebar Z_2$, with $\veebar$ the logical "xor", are encoded as algebraic linear constraints on the binary-valued variables: $z_1 + z_2 = 1$; see [15, Table 4]. Disjunctions involving linear constraints, e.g., $Z_5 \Rightarrow u_{15} = 2u_{16}$ and $\neg Z_5 \Rightarrow u_{15} = u_{16} = 0$, are converted into linear big-M constraints. For disjunctions with nonlinear expressions we introduce auxiliary real-valued variables w and augment the objective with a quadratic penalty term. For instance, $Z_1 \Rightarrow \varphi_1(u) \leq 0$ becomes $w_1 \leq M(1 - z_1)$, for some large enough $M > 0$, with the nonlinear term $\lambda[\varphi_1(u) - w_1]^2$ added to the objective function, for some large penalty parameter $\lambda > 0$.

Simulations

Consider the problem of [15, Example 2], reformulated as discussed above with parameters $\lambda = M = 10^3$; in the form (P) there are 30 real-valued and 8 binary-valued decision variables, 11 linear equality constraints, 37 linear inequality constraints, and simple bounds. We run Algorithm 1 on this instance starting from 100 initial guesses obtained by sampling a standard normal distribution; we then execute the variant of Algorithm 1 with NLP refinement from the same initial guesses. To stop the execution, we limit the number of successful iterations (that is, actual

updates) to 1000. Finally, for comparing with a tree search approach, we enumerate all $2^8 = 256$ integer combinations and test their compatibility with $\mathcal{X}$ (by checking feasibility of the associated fixed-integer LP). For each of the only 8 feasible combinations, we solve up to ε-stationarity the corresponding fixed-integer NLP, starting from 10 random initial guesses and picking the best solution to obtain a baseline objective value.

Algorithm 1 does not solve any of the 100 runs to the specified tolerance, hitting the iteration limit in 96 cases and otherwise generating a negative criticality measure. In contrast, the variant with NLP refinement returns successfully in 69 cases; the MILP solver declares the subproblem infeasible in 8 cases and generates a negative criticality measure in the remaining 23 cases. The best solution (in terms of objective value) is visualized in Fig. 3, while computational results are summarized in Table 2. Furthermore, Algorithm 1 performs similarly with the looser tolerance $\varepsilon = 10^{-6}$, whereas the variant with NLP refinement returns successfully in 98 cases, with the MILP solver generating a negative criticality measure in the two remaining.

Without NLP refinement, the algorithm was not able to reach an approximate critical point within the tolerance and the iteration limit. Behaving like a steepest descent method, the plain Algorithm 1 suffers from the slow tail convergence that is typical of first-order methods. Our numerical observations suggest that the integer components often settle on their final value after a few iterations, then leaving the real-valued components to progress steadily, albeit slowly, toward criticality. In contrast, despite the computational cost of additional NLP solves, the algorithm can

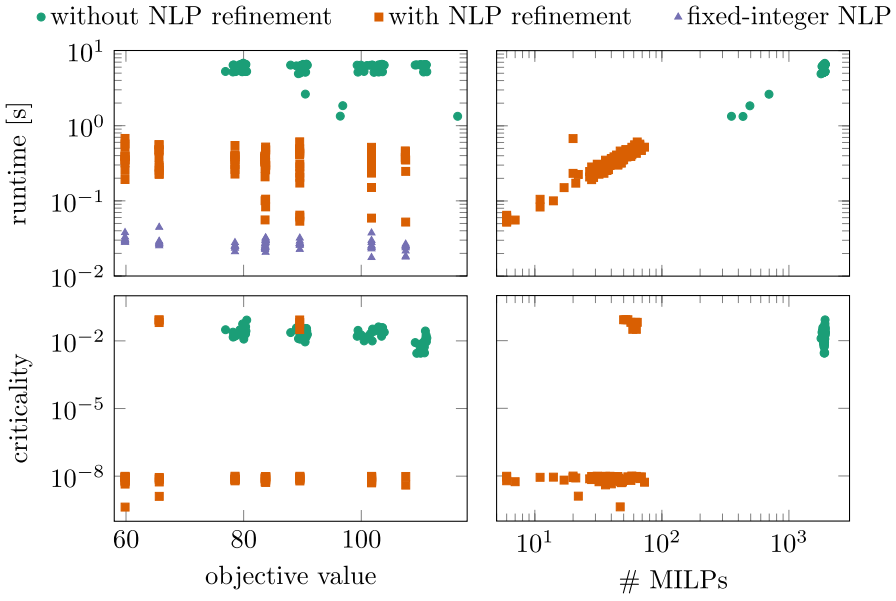

Fig. 4 Comparison of results for the problem of a processing network. Objective value, criticality measure, runtime and number of MILP subproblems for all test runs. Runtimes of fixed-integer NLP solves refer only to feasible integer combinations and does not include the preliminary feasibility check. Test runs returning with negative criticality are omitted in the lower plots

converge quickly when using second-order information. It is particularly important, however, that the integer-valued variables identify soon and correctly the active set. The relatively low number of iterations reported in Table 2 indicate the potential benefits of NLP refinement steps to speed up convergence, as illustrated in Fig. 4.

Finally, we point out that, in the variant with NLP refinement, Algorithm 1 always recovered one of the solutions that are obtained by enumerating the binary combinations and solving the associated fixed-integer NLP, thus indicating a good performance in terms of objective value too. Not including the preliminary step needed to ascertain feasibility, the runtime of these fixed-integer NLP solves is reported in Table 2 and Fig. 4 to contrast with that of Algorithm 1. With NLP refinement, Algorithm 1's computation time amounts to that of NLP and MILP solves, hence it hinges on how many integer combinations are explored. Analogously, typical MINLP techniques (such as nonlinear branch and bound, outer approximation and Benders decomposition) involve searching on a tree whose nodes correspond to NLPs; as a result, their performance depends on the branching strategy and the relaxation tightness [1]. How these methodologies can be integrated with Algorithm 1 and investigations on their relative efficiency are left for future assessments.

5 Final remarks

The results in this paper could be extended by integrating acceleration schemes and exploring the effects of inexact subproblem solves on the convergence. A connection with other stationarity notions should be established, at least for some prominent settings, such as problems with disjunctive constraints. Weaker assumptions on the problem structure and stronger optimality concepts could also be examined, for instance replacing $\Psi(x;\Delta)$ with $\Psi(x;\Delta)/\Delta$ as criticality measure, see Lemma 2. Future research may focus on active-set warm-starting and lazy constraints for computational efficiency, taking advantage of the similarity of successive inner MILPs. The proposed approach could also be adopted to tackle problems involving additional nonlinear constraints, by integrating it within sequential partially-constrained optimization schemes, such as penalty and barrier methods.

Acknowledgements The author thanks the anonymous reviewers for their stimulating comments and acute observations, which led eventually to stronger arguments and improved clarity.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability All source codes used to generate and analyze the results in Sect. 4 have been archived on Zenodo at DOI: 10.5281/zenodo.10007957.

Declarations

Conflict of interest The author declares that they have no conflict of interest.

References

1. 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
2. Grossmann, I.E., Trespalacios, F.: Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming. AIChE J. **59**(9), 3276–3295 (2013). https://doi.org/10.1002/aic.14088
3. D'Ambrosio, C., Frangioni, A., Liberti, L., Lodi, A.: A storm of feasibility pumps for nonconvex MINLP. Math. Program. **136**(2), 375–402 (2012). https://doi.org/10.1007/s10107-012-0608-x
4. De Marchi, A.: Proximal gradient methods beyond monotony. J. Nonsmooth Anal. Optim. (2023). https://doi.org/10.46298/jnsao-2023-10290
5. Quirynen, R., Di Cairano, S.: Sequential quadratic programming algorithm for real-time mixed-integer nonlinear MPC. In: 2021 60th IEEE Conference on Decision and Control (CDC), pp. 993–999 (2021). https://doi.org/10.1109/CDC45484.2021.9683714
6. Exler, O., Schittkowski, K.: A trust region SQP algorithm for mixed-integer nonlinear programming. Optim. Lett. **1**(3), 269–280 (2007). https://doi.org/10.1007/s11590-006-0026-1
7. Yuan, Y.-X.: Conditions for convergence of trust region algorithms for nonsmooth optimization. Math. Program. **31**(2), 220–228 (1985). https://doi.org/10.1007/BF02591750
8. Byrd, R.H., Gould, N.I.M., Nocedal, J., Waltz, R.A.: On the convergence of successive linear-quadratic programming algorithms. SIAM J. Optim. **16**(2), 471–489 (2005). https://doi.org/10.1137/S1052623403426532
9. Kirches, C., Larson, J., Leyffer, S., Manns, P.: Sequential linearization method for bound-constrained mathematical programs with complementarity constraints. SIAM J. Optim. **32**(1), 75–99 (2022). https://doi.org/10.1137/20M1370501
10. Themelis, A., Stella, L., Patrinos, P.: Forward-backward envelope for the sum of two nonconvex functions: further properties and nonmonotone linesearch algorithms. SIAM J. Optim. **28**(3), 2274–2303 (2018). https://doi.org/10.1137/16M1080240
11. Aravkin, A.Y., Baraldi, R., Orban, D.: A proximal quasi-Newton trust-region method for nonsmooth regularized optimization. SIAM J. Optim. **32**(2), 900–929 (2022). https://doi.org/10.1137/21M1409536
12. Birgin, E.G., Martínez, J.M.: Practical Augmented Lagrangian Methods for Constrained Optimization. Society for Industrial and Applied Mathematics, Philadelphia (2014). https://doi.org/10.1137/1.9781611973365
13. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. CMS Books in Mathematics, Springer, Cham (2017). https://doi.org/10.1007/978-3-319-48311-5
14. Nurkanović, A., Diehl, M.: Continuous optimization for control of hybrid systems with hysteresis via time-freezing. IEEE Control Syst. Lett. **6**, 3182–3187 (2022). https://doi.org/10.1109/LCSYS.2022.3181888
15. Türkay, M., Grossmann, I.E.: Logic-based MINLP algorithms for the optimal synthesis of process networks. Comput. Chem. Eng. **20**(8), 959–978 (1996). https://doi.org/10.1016/0098-1354(95)00219-7