A Voronoi-Based Mixed-Integer Gauss-Newton Algorithm for MINLP Arising in Optimal Control

Andrea Ghezzi[1], Léo Simpson[2], Adrian Bürger[3], Clemens Zeile[4], Sebastian Sager[4], Moritz Diehl[1,5]

Abstract— **We present a new algorithm for addressing nonconvex Mixed-Integer Nonlinear Programs (MINLPs) where the cost function is of nonlinear least squares form. We exploit this structure by leveraging a Gauss-Newton quadratic approximation of the original MINLP, leading to the formulation of a Mixed-Integer Quadratic Program (MIQP), which can be solved efficiently. The integer solution of the MIQP is used to fix the integer variables of the original MINLP, resulting in a standard Nonlinear Program. We introduce an iterative procedure to repeat the optimization of the two programs in order to improve the solution. To guide the iterations towards unexplored regions, we devise a strategy to partition the integer solution space based on Voronoi diagrams. Finally, we first illustrate the algorithm on a simple example of MINLP and then test it on an example of real-world complexity concerning the optimal control of an energy system. Here, the new algorithm outperforms state-of-the-art methods, finding a solution with a lower objective value, at the cost of requiring an increased runtime compared to other approximate methods.**

I. Introduction

This paper introduces an algorithm to approximately solve nonconvex Mixed-Integer Nonlinear Programs (MINLP) of the form

$$\begin{aligned} \min_{y \in \mathbb{Z}^{n_y}, z \in \mathbb{R}^{n_z}} \quad & F(y,z) \\ \text{s.t.} \quad & G(y,z) = 0, \\ & H(y,z) \leq 0, \\ & y \in \mathbb{P}, \end{aligned} \tag{1}$$

where the cost function F is of the following nonlinear least squares form

$$F(y,z) := \frac{1}{2}\|F_1(y,z)\|_2^2 + F_2(y,z), \tag{2}$$

with $F_1 : \mathbb{R}^{n_y+n_z} \to \mathbb{R}^{n_1}$ and $F_2 : \mathbb{R}^{n_y+n_z} \to \mathbb{R}$ two differentiable functions and $\|\cdot\|_2$ the L_2 norm. The integer variables y are restricted to be in $\mathbb{P}$, a convex polyhedron. For fixed integer variables, we define the feasible set of continuous variables as follows

$$\mathcal{Z}(y) := \{z \mid G(y,z) = 0,\ H(y,z) \leq 0\} \tag{3}$$

Assumption 1. *For any $y \in \mathbb{P}$, the set $\mathcal{Z}(y)$ is nonempty.*

Note that Assumption 1 can be satisfied by slacking the constraints and adding the slacks to the objective function via exact penalty functions.

We propose an algorithm to approximately solve (nonconvex) MINLPs of the form (1) by sequentially solving a MIQP and a Nonlinear Program (NLP). The MIQP serves as a local approximation of the MINLP (1), providing an integer feasible solution which can be computed efficiently with existing solvers. By fixing the integer part of the original MINLP, we obtain an NLP which is solved in order to generate a feasible iterate that is optimal with respect to the continuous variables. Our algorithm iterates over these two steps. More precisely, to improve the solution, we solve again these two problems by using the current optimal solution as a linearization point for the MIQP. Between one iteration and the next, we create a convex subset of the integer space based on Voronoi diagrams which excludes previously visited solutions from the search space. We present the computation of the Voronoi diagram as a good heuristic to enforce a disciplined search in the integer solution space, but we cannot guarantee to find global minimizers. Secondly, we apply the algorithm to a Mixed-Integer Optimal Control Problem (MIOCP) for an energy system with real-world complexity where we obtain a solution in a reasonable time. This solution is better, in the sense of lower objective value, than the one computed by state-of-the-art methods for nonlinear MIOCP with this complexity.

Related work: Generic MINLPs are considered as difficult optimization problems as they combine the difficulties of Mixed-Integer Linear Programs (MILPs) and NLPs [1]. To this end, a large part of this research field deals with convex MINLPs, for which solvers such as Bonmin [2] and SHOT [3] have been developed. Our approach is based on approximating the (nonconvex) MINLP with MIQP, which is computationally advantageous. Similar approaches use Sequential Quadratic Programming (SQP)-type algorithms [4] or quadratic outer approximation [5] as part of a sequence of MIQP to be solved. We refer to [6] for a recent review on MINLP methods.

Nonconvex MINLPs arise in particular from the discretization of Mixed-Integer Optimal Control Problems (MIOCPs). For MIOCPs, the combinatorial integral approximation de-

[1] Department of Microsystems Engineering (IMTEK), University of Freiburg, 79110 Freiburg, Germany {andrea.ghezzi, moritz.diehl}@imtek.uni-freiburg.de

[2] Research and Development, Tool-Temp AG, Switzerland leo.simpson@tool-temp.ch

[3] Karlsruhe University of Applied Sciences, 76133 Karlsruhe, Germany adrian.buerger@h-ka.de

[4] Institute for Mathematical Optimization, Faculty of Mathematics, Otto von Guericke University Magdeburg, 39106 Magdeburg, Germany, {clemens.zeile, sager}@ovgu.de

[5] Department of Mathematics, University of Freiburg, 79104 Freiburg, Germany

This research was supported by DFG via Research Unit FOR 2401, project 424107692, SPPs 1962 and 2331 and GRK 2297 MathCoRe (314838170) and by the EU via ELO-X 953348. We thank Armin Nurkanović and Rudolf Reiter for helpful discussions.

composition [7]–[9] is a well-established approach that solves the discretized MIOCP with relaxed integer variables before approximating the relaxed variables with integer ones obtained from a simple MILP. Recently, this decomposition approach was extended by using an MIQP for the second stage of the algorithm based on the idea to linearize the MINLP around the solution of the relaxed problem and using a quadratic Gauss-Newton approximation of the objective [10]. We propose a generalization of this method by solving a sequence of such Gauss-Newton based MIQPs guided by Voronoi-region constraints.

In the context of model-predictive control of MIOCPs, a mixed-integer sequential quadratic programming algorithm has been proposed in [11], but in contrast to our work no NLPs are solved as intermediate steps.

II. Problem formulation

Before presenting the algorithm, we introduce the two optimization problems we have to solve during the algorithm iterations.

A. Gauss-Newton approximation

We show how the generic MINLP (1) can be locally approximated by a convex MIQP, obtained from a Gauss-Newton approximation [10]. First, we define the linearization of the function F_1 at $(\bar{y}, \bar{z})$ as a Taylor series of first order

$$F_{1,\mathrm{L}}(y, z; \bar{y}, \bar{z}) := F_1(\bar{y}, \bar{z}) + \frac{\partial F_1}{\partial(y,z)}(\bar{y}, \bar{z}) \begin{bmatrix} y - \bar{y} \\ z - \bar{z} \end{bmatrix}. \quad (4)$$

In the same way, we define the linearizations $F_{2,\mathrm{L}}(y, z; \bar{y}, \bar{z})$, $G_{\mathrm{L}}(y, z; \bar{y}, \bar{z})$ and $H_{\mathrm{L}}(y, z; \bar{y}, \bar{z})$. Hence, we can define the Gauss-Newton approximation of F as

$$F_{\mathrm{GN}}(y, z; \bar{y}, \bar{z}) := \frac{1}{2}\|F_{1,\mathrm{L}}(y, z; \bar{y}, \bar{z})\|_2^2 + F_{2,\mathrm{L}}(y, z; \bar{y}, \bar{z}) \quad (5)$$

Eventually, we denote by $\mathbb{V}$ a convex polyhedron that contains $\bar{y}$. $\mathbb{V}$ will be computed during the algorithm's runtime and might restrict the feasible set of y. Thus, given the linearization point $(\bar{y}, \bar{z})$, the local convex quadratic approximation of the original MINLP (1) is given by the following MIQP

$$\mathcal{P}_{\mathrm{GN}}(\bar{y}, \bar{z}, \mathbb{V}): \quad \begin{aligned} \min_{y \in \mathbb{Z}^{n_y}} \quad & J_{\mathrm{GN}}(y; \bar{y}, \bar{z}) && (6a) \\ \text{s.t.} \quad & y \in \mathbb{P} \cap \mathbb{V}, && (6b) \end{aligned}$$

where

$$\begin{aligned} J_{\mathrm{GN}}(y; \bar{y}, \bar{z}) := \min_{z \in \mathbb{R}^{n_z}} \quad & F_{\mathrm{GN}}(y, z; \bar{y}, \bar{z}) && (7a) \\ \text{s.t.} \quad & G_{\mathrm{L}}(y, z; \bar{y}, \bar{z}) = 0, && (7b) \\ & H_{\mathrm{L}}(y, z; \bar{y}, \bar{z}) \leq 0. && (7c) \end{aligned}$$

The solution of the optimization problem $\mathcal{P}_{\mathrm{GN}}(\bar{y}, \bar{z}, \mathbb{V})$ is denoted by $y^\star$.

Note that problems (6) and (7) can be stated as one MIQP in the variables $y \in \mathbb{Z}^{n_y}, z \in \mathbb{R}^{n_z}$. We adopt the notation above because we are only interested to obtain the integer variables y from this problem.

B. Continuous optimization with fixed integers

In the second optimization step required by our algorithm, we optimize over the continuous variables z of the MINLP (1) while the integer variables are kept fixed to $y^\star$. Concretely, we solve the following NLP

$$\mathcal{P}(y^\star): \quad \begin{aligned} J(y^\star) := \min_{z \in \mathbb{R}^{n_z}} \quad & F(y^\star, z) && (8a) \\ \text{s.t.} \quad & G(y^\star, z) = 0, && (8b) \\ & H(y^\star, z) \leq 0. && (8c) \end{aligned}$$

The solution of the optimization problem $\mathcal{P}(y^\star)$ is denoted by $z^\star$. The evaluation of the cost function at the optimal solution, $J(y^\star) = F(y^\star, z^\star)$, is the guiding value function of our algorithm.

In the next section we devise an algorithm that alternates between the optimization problems (6) and (8) to approximate the solution of the MINLP (1).

Remark 1. *The general class of MINLP formulated as* (1) *includes discretized MIOCPs, which can be cast into the form* (1) *by considering* $z := (\mathbf{x}, \mathbf{u})$*, where* $\mathbf{x} := (x_0, \ldots, x_N)$ *and* $\mathbf{u} := (u_0, \ldots, u_{N-1})$ *denote the sequence of the system states and continuous controls over the prediction horizon* N*. The integer variables* y *can also be stacked and rewritten in the stage-wise fashion as* $\mathbf{y} = (y_0, \ldots, y_{N-1})$*. The cost function* (2) *is typical in Optimal Control Problems (OCPs), for instance in tracking costs.*

III. Proposed Algorithm

The proposed procedure is described in Algorithm 1. In order to run Algorithm 1, we need an initial guess $(\bar{y}_0, \bar{z}_0)$ for the linearization that leads to the first MIQP. We outline a way to compute an initial guess in Subsection III-C. The first two steps of the algorithm consist of the solution of problems (6) and (8). In detail, we firstly solve $\mathcal{P}_{\mathrm{GN}}(\bar{y}_k, \bar{z}_k, \mathbb{V}_k)$ from which we get the integer solution $y_k^\star$. Secondly, we use this solution in order to solve $\mathcal{P}(y_k^\star)$, from which we get the optimal continuous solution $z_k^\star$ and the objective value denoted by $J_k := F(y_k^\star, z_k^\star)$. The if-condition (Line 5 of Algorithm 1) is true if the new solution point yields a lower objective value than the previous best solution. This condition is always true at the first iteration, because we initialize $\bar{J}_0$ to be $+\infty$. The instructions executed inside the if-condition keep track of the best solution so far, i.e., the tuple $(\bar{y}_k, \bar{z}_k)$, which yields the lowest objective value so far.

Eventually, we compute the Voronoi diagram $\mathbb{V}_{k+1}$ for the current best solution $\bar{y}_{k+1}$ given the sequence of integer solutions computed so far $(y_0^\star, \ldots, y_k^\star)$. In the first iteration, we have $(\bar{y}_1; y_0^\star)$ where $\bar{y}_1 \equiv y_0^\star$, therefore the Voronoi diagram is trivial and corresponds to $\mathbb{R}^{n_y}$.

A. Voronoi diagram computation

Let $\|\cdot\|$ be a (possibly weighted) Euclidean norm: for $x \in \mathbb{R}^{n_y}$, $\|x\| := \sqrt{x^\top W x}$ with $W \in \mathbb{R}^{n_y \times n_y}$ a symmetric positive-definite matrix. Given the collection of integer solutions computed $(y_0^\star, \ldots, y_k^\star)$ and the current best solution

Algorithm 1 Voronoi-Based Mixed-Integer Gauss-Newton

```
1:  Init.: $(\bar{y}_0, \bar{z}_0)$, $\bar{J}_0 \leftarrow +\infty$, $\mathbb{V}_0 \leftarrow \mathbb{R}^{n_y}$, $k \leftarrow 0, n_{\mathrm{NI}} \leftarrow 0$
2:  while not $(y_k^\star \equiv \bar{y}_k$ or $n_{\mathrm{NI}} > n_{\mathrm{NI,max}})$ do:
3:      Solve $\mathcal{P}_{\mathrm{GN}}(\bar{y}_k, \bar{z}_k, \mathbb{V}_k)$, Store solution in $y_k^\star$
4:      Solve $\mathcal{P}(y_k^\star)$, Store solution in $z_k^\star$ and value in $J_k$
5:      if $J_k < \bar{J}_k$ then:          ▷ Update the best solution
6:          $\bar{y}_{k+1} \leftarrow y_k^\star, \bar{z}_{k+1} \leftarrow z_k^\star, \bar{J}_{k+1} \leftarrow J_k$
7:          $n_{\mathrm{NI}} \leftarrow 0$
8:      else:
9:          $\bar{y}_{k+1} \leftarrow \bar{y}_k, \bar{z}_{k+1} \leftarrow \bar{z}_k, \bar{J}_{k+1} \leftarrow \bar{J}_k$
10:         $n_{\mathrm{NI}} \leftarrow n_{\mathrm{NI}} + 1$
11:     end if
12:     Compute Voronoi set $\mathbb{V}_{k+1}$ given $(\bar{y}_{k+1}; y_0^\star, \dots, y_k^\star)$
13:     $k \leftarrow k + 1$
14: end while
15: return $\bar{y}_k$, $\bar{z}_k$, $\bar{J}_k$
```

$\bar{y}_{k+1}$, the Voronoi set $\mathbb{V}_{k+1}$ is defined as follows

$$\mathbb{V}_{k+1} := \tag{9}$$
$$\left\{y \in \mathbb{R}^{n_y} \;\middle|\; \|y - \bar{y}_{k+1}\| \leq \|y - y_i^\star\|, i \in \mathbb{Z}_{[0,k]}, \bar{y}_{k+1} \neq y_i^\star\right\}$$

where $\mathbb{Z}_{[0,k]} := \{0, \dots, k\}$. It is easy to verify that $\bar{y}_{k+1} \in \mathbb{V}_{k+1}$, i.e., the linearization point of the MINLP is always feasible with respect to the new constraints.

Lemma 1. *The set $\mathbb{V}_{k+1}$ is a polyhedron for $k \in \mathbb{N}$, i.e., it can be defined with only affine constraints. More precisely, we have*

$$\mathbb{V}_{k+1} := \left\{y \in \mathbb{R}^{n_y} \;\middle|\; (a_{i,k+1})^\top y \leq b_{i,k+1},\, i \in \mathbb{Z}_{[0,k]}\right\}, \tag{10}$$

with $a_{i,k+1} := 2W(y_i^\star - \bar{y}_{k+1})$ and $b_{i,k+1} := \|y_i^\star\|^2 - \|\bar{y}_{k+1}\|^2$.

Proof: Based on our norm definition, we remark the following equivalences for the inequalities for $i \in \mathbb{Z}_{[0,k]}$:

$$\begin{aligned}
& \|y - \bar{y}_{k+1}\| \leq \|y - y_i^\star\| \\
\Leftrightarrow\;& \|y - \bar{y}_{k+1}\|^2 \leq \|y - y_i^\star\|^2 \\
\Leftrightarrow\;& \|y\|^2 - 2\bar{y}_{k+1}^\top W y + \|\bar{y}_{k+1}\|^2 \\
& \qquad\qquad \leq \|y\|^2 - 2y_i^{\star\top} W y + \|y_i^\star\|^2 \\
\Leftrightarrow\;& 2\left(y_i^\star - \bar{y}_{k+1}\right)^\top W y \leq \|y_i^\star\|^2 - \|\bar{y}_{k+1}\|^2,
\end{aligned}$$

which directly shows the desired property. □

Remark 2 (Voronoi diagram in case of binary variables)**.** *For the important special case of binary optimization variables $y \in \{0,1\}^{n_y}$ and weighting matrix $W = I$, the inequality $\|y - \bar{y}_{k+1}\| \leq \|y - y_i^\star\|$ used in* (9) *to define the Voronoi sets can be formulated as*

$$\sum_{j \in N(\bar{y})} y_j + \sum_{j \in B(\bar{y})} (1 - y_j) \leq \sum_{j \in N(y_i^\star)} y_j + \sum_{j \in B(y_i^\star)} (1 - y_j), \tag{11}$$

with $\mathbb{Z}_{[1,n_y]} = \{1, \dots, n_y\}$ that defines the index sets

$$B(\bar{y}) := \{j \in [n_y] \mid \bar{y}_j = 1\}, \quad N(\bar{y}) := [n_y] \setminus B(\bar{y}). \tag{12}$$

In (11)*, the left-hand side counts the number of entries in which y differs from $\bar{y}$, and the right-hand side for y and $y_i^\star$. Using this inequality instead of* (10) *has the advantages that no matrix-vector products need to be calculated for new iterates $\bar{y}_k$ and that the integer coefficients might avoid numerical issues of the MIQP solver.*

B. Termination condition

In the following, we explain the termination condition of Algorithm 1. Given the complexity of the problem class, we can only derive a heuristic-based stopping criterion. We stop the iterations when $y_k^\star \equiv \bar{y}_k$, which means that the current iteration has yielded the same integer solution as the so far best one, i.e., $\mathcal{P}_{\mathrm{GN}}(\bar{y}_k, \bar{z}_k, \mathbb{V}_k)$ outputs $\bar{y}_k$ itself. When this situation occurs, the algorithm cannot continue since the solution of the NLP $\mathcal{P}(y_k^\star)$ provides no new solution so that the Voronoi set remains unchanged. In fact, we have partitioned the integer space into k-regions and in the center of the k-th region there is the best integer feasible solution so far which cannot be easily improved in its own region. As will be shown in Section V, this condition might require numerous iterations and computation time before it is fulfilled, especially for complex problems.

As a result, we relax the termination criterion including the condition $n_{\mathrm{NI}} > n_{\mathrm{NI,max}}$, where n_{NI} is a counter for the consecutive non-improving iterations and $n_{\mathrm{NI,max}} \in \mathbb{N}^+$ is a user-defined parameter. Every time a new solution is computed and the if-condition at line 5 of Algorithm 1 is not satisfied, we have a non-improving iteration and the counter is incremented, i.e., $n_{\mathrm{NI}} \leftarrow n_{\mathrm{NI}} + 1$. On the other hand, if the if-condition is satisfied, the counter is reset, meaning $n_{\mathrm{NI}} \leftarrow 0$.

C. Obtaining an initial guess

The MIQP $\mathcal{P}_{\mathrm{GN}}(\bar{y}, \bar{z}, \mathbb{V})$ needs a linearization point $(\bar{y}, \bar{z})$ for the first iteration of the algorithm. A possible way to compute an initial guess is by continuous relaxation of the integer constraints in the MINLP (1) and solving the corresponding relaxed NLP as follows

$$\begin{aligned}
\min_{y \in \mathbb{R}^{n_y}, z \in \mathbb{R}^{n_z}} \quad & F(y, z) && \text{(13a)} \\
\text{s.t.} \quad & G(y, z) = 0, && \text{(13b)} \\
& H(y, z) \leq 0, && \text{(13c)} \\
& y \in \mathbb{P}. && \text{(13d)}
\end{aligned}$$

The solution of this problem, denoted by $y^\circ \in \mathbb{R}^{n_y}$ and $z^\circ \in \mathbb{R}^{n_z}$, might be used as first linearization point $(\bar{y}_0, \bar{z}_0)$. Note that, with this initialization, the first iteration of Algorithm 1 corresponds to the three-step algorithm presented in [10].

IV. Tutorial mixed-integer optimization example

We illustrate the behavior of the proposed algorithm for a MINLP where we can have an effective graphical

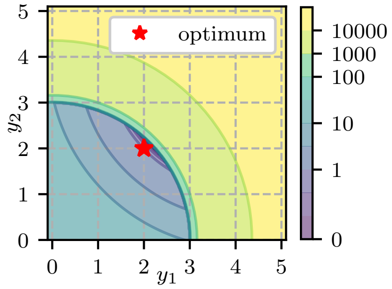

Fig. 1. Level lines of the cost function of problem (14). The red asterisk denotes the global minimizer.

TABLE I

ITERATIONS OF ALGORITHM 1 FOR THE EXAMPLE IV

k	$\bar{y}_k$	$\bar{J}_k$	$\mathbb{V}_k$	$y_k^\star$	$J_k^\star$
0	$(0,4)$	7016.81	$\mathbb{R}^2$	$(4,3)$	16001.01
1	$(0,4)$	7016.81	$\begin{bmatrix}8 & -2\end{bmatrix} y \leq \begin{bmatrix}9\end{bmatrix}$	$(1,3)$	1010.61
2	$(1,3)$	1010.61	$\begin{bmatrix}-2 & 2\\ 6 & 0\end{bmatrix} y \leq \begin{bmatrix}6\\ 15\end{bmatrix}$	$(2,2)$	8.41
3	$(2,2)$	8.41	$\begin{bmatrix}-4 & 4\\ 4 & 2\\ -2 & 2\end{bmatrix} y \leq \begin{bmatrix}8\\ 17\\ 2\end{bmatrix}$	$(2,2)$	8.41

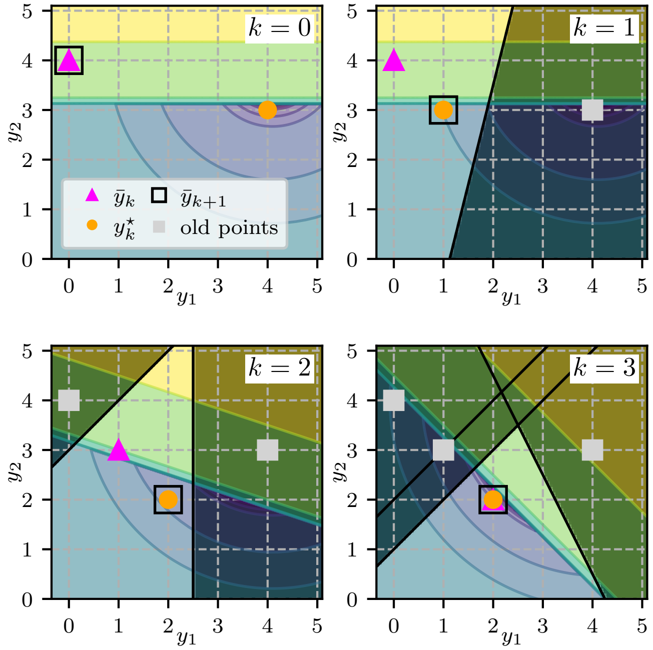

Fig. 2. Representation of algorithm iterations. The level lines correspond to the MIQP $\mathcal{P}_{\mathrm{GN}}(\bar{y}_k, \bar{z}_k, \mathbb{V}_k)$, the darkened areas describe the regions excluded by the Voronoi set $\mathbb{V}_k$ and the black lines correspond to the linear functions used to create the Voronoi set.

representation. Let us first consider the following MINLP

$$\min_{\substack{(y_1,y_2)\in\mathbb{Z}^2,\\ z\in\mathbb{R}}} \quad (y_1-4.1)^2+(y_2-4.0)^2+\lambda z \tag{14a}$$

$$\text{s.t.} \quad y_1^2+y_2^2-r^2-z \leq 0, \tag{14b}$$

$$-z \leq 0, \tag{14c}$$

where $r=3$ and $\lambda=1000$. The term λz in (14a) can be seen as a penalization of the violation of some quadratic constraint $y_1^2+y_2^2-r^2 \leq 0$. The shape of the cost function and the global minimizer of (14) are represented in Figure 1. The global minimizer can be found graphically and corresponds to $(y_1,y_2,z)=(2,2,0)$. Note that z is determined by $z=\max(0, y_1^2+y_2^2-r^2)$. Given a linearization point $(\bar{y}_1,\bar{y}_2,\bar{z})$, the Gauss-Newton problem $\mathcal{P}_{\mathrm{GN}}$ is equivalent to Problem (14) except for constraint (14b) which is substituted by its first order Taylor series as follows

$$-\bar{y}_1^2-\bar{y}_2^2+2y_1\bar{y}_1+2y_2\bar{y}_2-z \leq r^2. \tag{15}$$

Also, for the Gauss-Newton problem z is implicitly defined as $z=\max(0,-\bar{y}_1^2-\bar{y}_2^2+2y_1\bar{y}_1+2y_2\bar{y}_2-r^2)$. For this reason, in the following, we focus only on the values of y_1, y_2. By a slight abuse of notation, we stack y_1, y_2 in the vector y_k as $y_k := (y_1, y_2)$ where the subscript k denotes the algorithm's iteration.

We apply Algorithm 1 to solve (14). In order to better illustrate the algorithm we start from the user defined guess $\bar{y}_0=(0,4), \bar{z}_0=7$, which is an integer-feasible solution. Therefore, we can initialize $\bar{J}_0$ with its actual value and not with $+\infty$. The results of each iteration are collected in Table I, while Figure 2 gives a graphical interpretation of the iterations.

In the first iteration, the new point computed $(y_0^\star, z_0^\star)$ has a higher objective value than the initial guess $\bar{y}_0$, therefore the latter is kept as linearization point for the next iteration. Now, from Figure 2, one can notice the importance of the Voronoi diagram. In fact, in the iteration $k=1$, the Voronoi set $\mathbb{V}_1$ is preventing the solution of $\mathcal{P}_{\mathrm{GN}}(\bar{y}_1, \bar{z}_1, \mathbb{V}_1)$ to be again the point $y_0^\star=(4,3)$, which is the minimizer for $\mathcal{P}_{\mathrm{GN}}(\bar{y}_1, \bar{z}_1, \mathbb{R}^{n_y})$, as the MIQP level lines in Figure 2 suggest. The two plots on the top share the same level lines because they have the same linearization point $\bar{y}_k=(0,4)$. Therefore, without the tightening of the search space enforced by $\mathbb{V}_1$, the algorithm would be stuck in a cycle, in which it solves $\mathcal{P}_{\mathrm{GN}}(\bar{y}_0, \bar{z}_0, \mathbb{R}^{n_y})$ and finds $y_0^\star$, which is discarded for its higher objective compared to $\bar{y}_0$. Then, we would iterate endlessly by computing again $\mathcal{P}_{\mathrm{GN}}(\bar{y}_0, \bar{z}_0, \mathbb{R}^{n_y})$.

At the end of the second iteration, we find a new best solution $y_1^\star=(1,3)$ that serves for the linearization as part of the next iteration $\bar{y}_2=y_1^\star$. Once again, the Voronoi set $\mathbb{V}_2$ prevents the MIQP to reach the minimizer in $y=(3,2)$, which has a high objective value for the original MINLP (14) since z is necessarily >0. Hence, the solution of $\mathcal{P}_{\mathrm{GN}}(\bar{y}_2, \bar{z}_2, \mathbb{V}_2)$ gives $y_2^\star=(2,2)$ and the solution of $\mathcal{P}(y_2^\star)$ gives $z_2^\star=0$. The corresponding objective value is $J_2^\star=8.41$. This constructed solution is the new best point, so, in the next iteration, we linearize around it.

In the last iteration, the solution of $\mathcal{P}_{\mathrm{GN}}(\bar{y}_3, \bar{z}_3, \mathbb{V}_3)$ returns

$y_3^\star$, which is equal to $\bar{y}_3 = (2, 2)$. The termination condition of the algorithm is now satisfied. Thus, we stop the iterations and return the best solution determined $\bar{y}_3 = (2, 2), \bar{z}_3 = 0$ which is the global optimum of (14).

V. MIOCP for a renewable energy system

To test the proposed algorithm on a more complex example, we conduct a numerical case study for an extended version of the solar thermal climate system described in [12]. The system is installed at the Faculty of Management Science and Engineering at Karlsruhe University of Applied Sciences and is used to provide cooling for the building's main hall. Cooling energy can be generated by switching on the solar-thermally driven adsorption cooling machine (ACM) of the system or a free cooling (FC) mode that can be used when the ambient temperature is sufficiently low. In addition to these components described in [12], an additional switched cooling device in the form of a heat pump (HP) has recently been added to the system to increase its flexibility, i.e., to facilitate operation in case of low solar thermal energy or provision of additional cooling power. Hence, the system considered in this section is an example of real world complexity and a test for the behavior of the algorithm. A schematic of the system depicting the hydraulic connections considered in this study is shown in Figure 3. For simplicity, we assume that ACM and HP are directly connected to the recooling tower (in the real setup, these are connected via a buffer storage and a heat exchanger). The electric energy required to drive the system is considered to be either bought from the grid or provided by PV panels.

The system dynamics are modeled using a set of Ordinary Differential Equations (ODE) with differential states, continuous controls and binary controls. The three binary controls b_{ac}, b_{hp}, and b_{fc} corresponding to the switching of the ACM, the HP, and the FC mode are indicated in Figure 3. The system state includes the temperature of the flat plate solar collectors T_{fpsc} and the vacuum tube solar collectors T_{vtsc}, four temperature levels for the stratified hot temperature storage denoted by $T_{\mathrm{ht},1}, T_{\mathrm{ht},2}, T_{\mathrm{ht},3}, T_{\mathrm{ht},4}$, and the temperature of the low temperature storage T_{lt}. The conditions that affect the system operation are the ambient temperature, the solar irradiance on the solar collectors, the power generation of the local PV panels, the price of electric energy, and the defined cooling load profile.

Now, a MIOCP can be formulated with the aim to provide the cooling load required by the building while keeping the system state inside prescribed bounds and reducing the energy cost for the system operation. Then, the MIOCP is solved via a *direct approach*, i.e., we first discretize the system, then we optimize. Specifically, the system is discretized over a horizon of 24 hours with a non-equidistant grid with time steps of 15 minutes from 4 a.m. to 10 p.m. and of 30 minutes for the rest of the day. In total, we have a time horizon of $N = 84$ steps. For the NLP $\mathcal{P}$ we use a direct collocation scheme to obtain an accurate integration. The MINLP formulation (1) is obtained as discussed in Remark 1. In detail, the vector y contains the three binary controls stacked for each time step, z contains the state trajectory, the continuous controls and possibly the slack variables used to make the MIOCP feasible. The function $G(y, z)$ expresses the system dynamics, while $H(y, z)$ are the constraints acting on the system. The function $\|F_1(y, z)\|_2^2$ contains the tracking cost for achieving the requested cooling load while $F_2(y, z)$ expresses the operating cost. For the MIQP $\mathcal{P}_{\mathrm{GN}}$ we derive the G_{L} matrix from the discretization of the MIOCP via multiple-shooting in order to obtain a smaller matrix compared to the one derived by direct collocation and to speed up the solution of the MIQP. To implement the discretization schemes we use CasADi [13] via its Python interface. To solve the NLPs we use Ipopt [14] with HSL MA57 [15] as internal linear solver, while we use Gurobi [16] with integer gap set to 5% to solve the MIQPs. The simulations are conducted on a computer with Intel Xeon E5-2687W @3.1 GHz, 8 cores and 32 GB RAM.

We compute the initial guess for linearization according to Subsection III-C by solving the corresponding NLP with continuous relaxation of the binary variables (13) and run Algorithm 1 with $n_{\mathrm{NI,max}} = 15$. Figure 4 shows the behavior of the algorithm with increasing number of iterations. J_{23} yields the lowest objective value and since the solution is not improving during the upcoming $n_{\mathrm{NI,max}}$ iterations, the algorithm stops at iteration 38 with $n_{\mathrm{NI}} = n_{\mathrm{NI,max}}$. As shown in the figure, the ratio between J_k and J°, i.e., the integer gap, drops by 30% between the first iteration and the best iteration of the algorithm, which is a remarkable result. This improvement is achieved mainly by a lower operating cost and smaller constraint relaxation for T_{lt}. Note that the computation of the first iteration corresponds to the algorithm proposed in [10].

The comparison of the energy cost for operating the system with the two solutions is depicted in Figure 5. The orange line c_{best}, corresponding to the cumulative cost for operating the system with the best found solution, is always under the blue line c_{first}, which is obtained by applying the solution of the first iteration. We assumed that buying and selling energy from and to the grid has the same price, which is given by the green dash-dotted line.

In Figure 6, we compare the state trajectories and the controls given by the solutions at the first iteration and at the best iteration respectively. One can notice that the two solutions involve different binary controls resulting in different state trajectories. As stated above, the best solution yields slightly lower temperature constraint violations for T_{lt}.

The main drawback of the proposed algorithm is the high computation time. We illustrate the cumulative runtime for the two solvers in the bottom plot of Figure 4. It shows that the algorithm takes more than 150 hours and its main portion is due to the solution of the MIQPs. To mitigate this effect, one could set a lower value for $n_{\mathrm{NI,max}}$ or a slightly higher integer gap for Gurobi. In addition, the solution time of the MIQP could be further reduced by increasing the number of available CPU cores or by the development of a tailored Branch-and-Bound algorithm designed for this specific MIQP type.

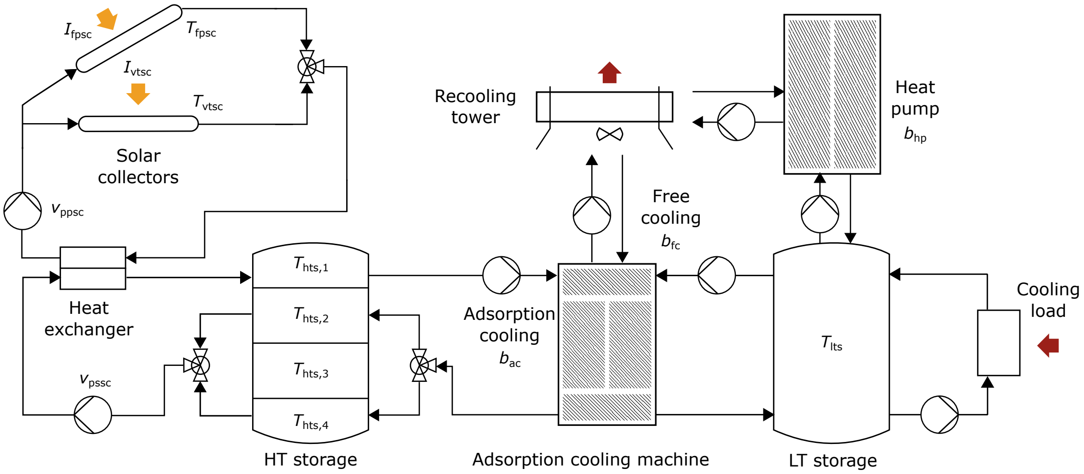

Fig. 3. Schematic of the energy system considered. A detailed explanation of the system can be found in [12].

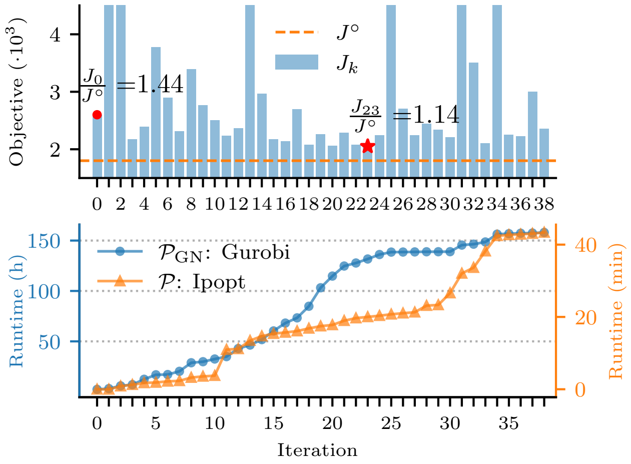

Fig. 4. Algorithm performance as a function of the iterations. Top: Objective values for the constructed solutions represented by the blue bars. The red dot and red asterisk indicate the first and best constructed solution, respectively. The objective value of the initial guess, denoted as J° and depicted by the orange dashed line, is used as a lower bound for benchmarking the mixed-integer solutions. Bottom: Cumulative runtime for solving the MIQPs on the left y-axis and the NLPs on the right y-axis. Gurobi is used for MIQPs, Ipopt for the NLPs.

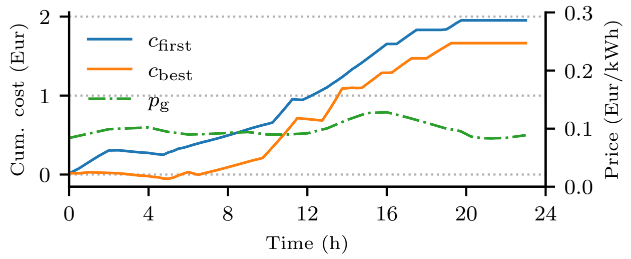

Fig. 5. Comparison of the cumulative energy cost for the first and best determined solution of Algorithm 1, c_{first} and c_{best}, respectively. The dashed-dotted green line p_{g} indicates the price of energy over the test day.

In the literature, an established method to solve MIOCPs is the Combinatorial Integral Approximation (CIA) approach [7]. This approach has a shorter runtime compared to the algorithm proposed in this work since the integer variables are computed in a MILP with a special structure. An interesting property of the CIA is that by refining the control discretization grid one can asymptotically obtain the optimal solution. However, when the grid cannot be further refined and one looks for the optimal solution for a given grid, the proposed algorithm is beneficial and one can outperform the solution constructed by CIA already in the first iteration (see [10]). Finally, the runtime of generic (nonconvex) MINLP solvers such as Bonmin or SCIP is generally beyond an acceptable time range for complex and real-world MIOCPs such as the energy problem under consideration. Thus, the heuristically determined solution from our algorithm is good and comprised of a high solution quality with still acceptable runtime.

VI. Conclusions

In this work, we have presented an algorithm that approximately solves nonconvex MINLPs with a cost function of a nonlinear least squares form. The novel algorithm combines an iterative procedure inspired by Mixed-Integer SQP with a Voronoi-based heuristic that promotes a disciplined search within the integer solution space. In the tutorial example, we have shown that the local quadratic approximation provides valid approximations of the original problem, but only with the tightening provided by the Voronoi diagram the algorithm can carry out meaningful iterations and avoids deadlock situations. The algorithm can be adopted for solving discretized MIOCP with a predominant complexity in the continuous dynamics, in contexts where a small runtime is not a strict requirement and where it is fundamental to improve available solutions. Future work could try to speed up the MIQP solution process with dedicated algorithms that exploit its OCP structure. Eventually, we might investigate variations of the algorithm that satisfy convergence guarantees.

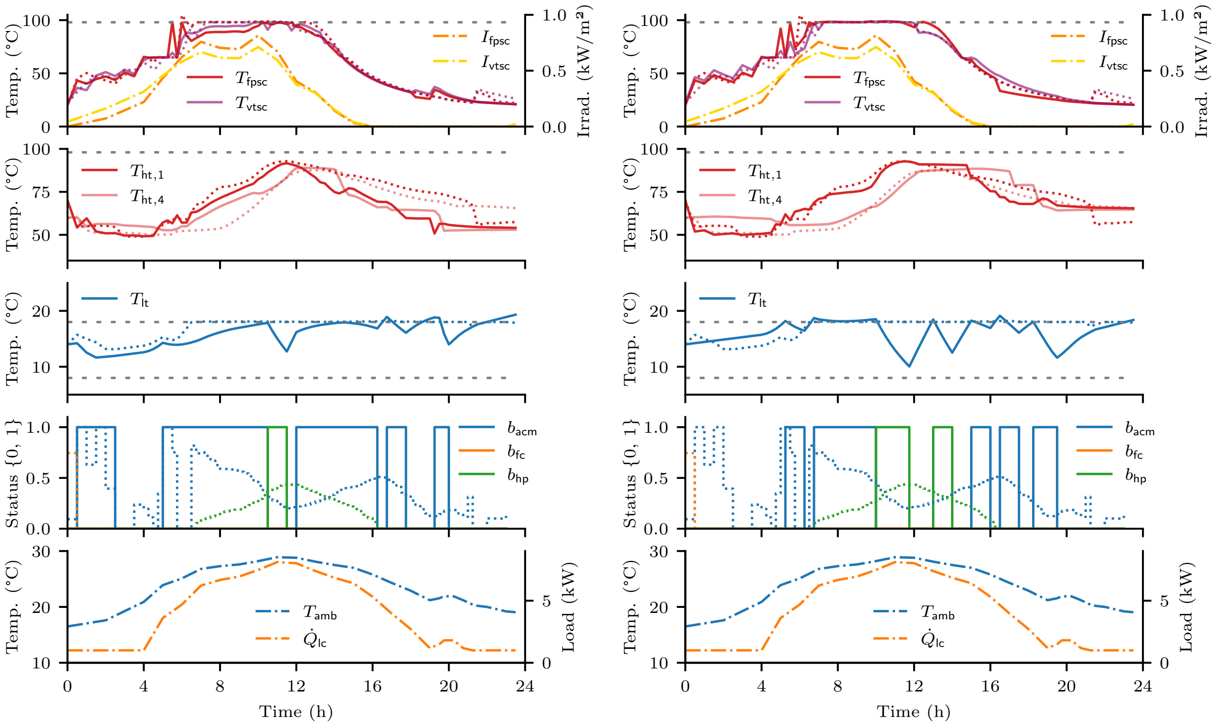

Fig. 6. Comparison of the system trajectories computed at the first iteration of the algorithm (left) and at iteration 23 (right), which is the best determined solution in terms of the objective. The state bounds are expressed in the plots by horizontal dashed gray lines. The dotted lines are the trajectories of the relaxed solutions (the initial guess) and the dash-dotted lines correspond to the ambient conditions. Specifically, the solar irradiation on the vacuum tube solar collectors (VTSC) and the flat plate solar collectors (FPSC) is denoted by I_{vtsc} and I_{fpsc}, respectively, the ambient temperature by T_{amb}, and the cooling load profile by $\dot{Q}_{\mathrm{lc}}$.

References

[1] J. Kronqvist, D. E. Bernal, A. Lundell, and I. E. Grossmann, “A review and comparison of solvers for convex MINLP,” *Optimization and Engineering*, vol. 20, no. 2, pp. 397–455, 2019.

[2] P. Bonami, L. Biegler, A. Conn, G. Cornuéjols, I. Grossmann, C. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, “An Algorithmic Framework For Convex Mixed Integer Nonlinear Programs,” IBM T. J. Watson Research Center, Tech. Rep., 2005.

[3] J. Kronqvist, A. Lundell, and T. Westerlund, “The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming,” *Journal of Global Optimization*, vol. 64, no. 2, pp. 249–272, 2016.

[4] O. Exler, T. Lehmann, and K. Schittkowski, “A comparative study of SQP-type algorithms for nonlinear and nonconvex mixed-integer optimization,” *Mathematical Programming Computation*, vol. 4, no. 4, pp. 383–412, 2012.

[5] R. Fletcher and S. Leyffer, “Solving Mixed Integer Nonlinear Programs by Outer Approximation,” *Mathematical Programming*, vol. 66, pp. 327–349, 1994.

[6] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, “Mixed-integer nonlinear optimization,” *Acta Numerica*, vol. 22, pp. 1–131, 2013.

[7] S. Sager, M. Jung, and C. Kirches, “Combinatorial integral approximation,” *Mathematical Methods of Operations Research*, vol. 73, no. 3, pp. 363–380, 2011.

[8] S. Sager, H. G. Bock, and M. Diehl, “The integer approximation error in mixed-integer optimal control,” *Mathematical Programming (Series A)*, vol. 133, pp. 1—23, 2012.

[9] C. Zeile, T. Weber, and S. Sager, “Combinatorial integral approximation decompositions for mixed-integer optimal control,” *Algorithms*, vol. 15, no. 4, p. 121, 2022.

[10] A. Bürger, C. Zeile, A. Altmann-Dieses, S. Sager, and M. Diehl, “A Gauss-Newton-based decomposition algorithm for nonlinear mixed-integer optimal control problems,” *Optimization-Online preprint*, 2022, accessed: 2022–10–20. [Online]. Available: https://optimization-online.org/2022/04/8890/

[11] R. Quirynen and S. Di Cairano, “Sequential quadratic programming algorithm for real-time mixed-integer nonlinear mpc,” in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 993–999.

[12] A. Bürger, D. Bull, P. Sawant, M. Bohlayer, A. Klotz, D. Beschütz, A. Altmann-Dieses, M. Braun, and M. Diehl, “Experimental operation of a solar-driven climate system with thermal energy storages using mixed-integer nonlinear model predictive control,” *Optimal Control Applications and Methods*, vol. 42, pp. 1293–1319, 2021.

[13] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – a software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[14] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[15] HSL, “A collection of Fortran codes for large scale scientific computation.” 2011, accessed: 2022–10–20. [Online]. Available: http://www.hsl.rl.ac.uk

[16] Gurobi Optimization, LLC, “Gurobi Optimizer Reference Manual,” 2022, accessed: 2022–10–20. [Online]. Available: https://www.gurobi.com