Journal of Optimization Theory and Applications (2022) 195:1056–1086
https://doi.org/10.1007/s10957-022-02114-y

Outer Approximation for Mixed-Integer Nonlinear Robust Optimization

Martina Kuchlbauer[1] · Frauke Liers[1] · Michael Stingl[1]

Received: 21 August 2021 / Accepted: 7 September 2022 / Published online: 27 October 2022

Abstract

Currently, few approaches are available for mixed-integer nonlinear robust optimization. Those that do exist typically either require restrictive assumptions on the problem structure or do not guarantee robust protection. In this work, we develop an algorithm for convex mixed-integer nonlinear robust optimization problems where a key feature is that the method does not rely on a specific structure of the inner worst-case (adversarial) problem and allows the latter to be non-convex. A major challenge of such a general nonlinear setting is ensuring robust protection, as this calls for a global solution of the non-convex adversarial problem. Our method is able to achieve this up to a tolerance, by requiring worst-case evaluations only up to a certain precision. For example, the necessary assumptions can be met by approximating a non-convex adversarial via piecewise relaxations and solving the resulting problem up to any requested error as a mixed-integer linear problem.

In our approach, we model a robust optimization problem as a nonsmooth mixed-integer nonlinear problem and tackle it by an outer approximation method that requires only inexact function values and subgradients. To deal with the arising nonlinear subproblems, we render an adaptive bundle method applicable to this setting and extend it to generate cutting planes, which are valid up to a known precision. Relying on its convergence to approximate critical points, we prove, as a consequence, finite convergence of the outer approximation algorithm.

As an application, we study the gas transport problem under uncertainties in demand

Communicated by Miguel F. Anjos.

✉ Martina Kuchlbauer
martina.kuchlbauer@fau.de

Frauke Liers
frauke.liers@fau.de

Michael Stingl
michael.stingl@fau.de

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany, Cauerstr. 11, 91058 Erlangen, Germany

and physical parameters on realistic instances and provide computational results demonstrating the efficiency of our method.

Keywords Robust optimization · Mixed-integer nonlinear optimization · Outer approximation · Bundle method · Gas transport problem

Mathematics Subject Classification 90C17 · 90C30 · 90C11 · 90C47 · 90C35

1 Introduction

In recent years, tremendous progress has been made in developing algorithms for mixed-integer nonlinear optimization problems (MINLP). Nevertheless, they remain one of the most challenging optimization problems studied to date, and in particular, the global solution of even reasonably-sized instances can be out of reach. In addition, optimization problems are typically prone to uncertainties in the input data due to measurement errors, fluctuations or insufficient knowledge of the underlying applications' characteristics. Ignoring these uncertainties might lead to decisions that are not only suboptimal but even infeasible.

In robust optimization, we typically first define uncertainty sets containing the realizations we wish to protect against. Decisions that are feasible for all realizations within the uncertainty sets are termed robust feasible and from these, the ones with the best objective value are called robust optimal. This results in an optimization problem of the form

$$\begin{aligned} \min_{x,y} \quad & C(x,y) \\ \text{s.t.} \quad & V_i(x,y,u) \leq 0 \quad \forall u \in \mathcal{U}, i \in \{1,\dots,n\} \\ & x \in X, y \in Y \cap \mathbb{Z}^{n_y}, \end{aligned} \tag{1}$$

with decision variables x, y and an uncertain parameter u. The task to determine the worst-case realization of the uncertainty for a candidate solution is called adversarial problem. For (1), this reads $\max_{u\in\mathcal{U}, i\in[n]} V_i(x,y,u)$. Although robust optimization problems are not tractable in general, practically efficient solution approaches have been developed for broad classes of problems, for example for robust combinatorial and for mixed-integer linear optimization. However, robust mixed-integer nonlinear problems are still very challenging both in theory and in practice, where the development of general approaches is still in its infancy. For a recent review of the current state-of-the-art, we refer to [26].

Reformulations of the robust counterparts to an algorithmically tractable problem rely on strong assumptions on the problem structure. In particular, it is usually necessary for such exact reformulation approaches to assume that the problem is convex in the decisions (x, y in (1)) and fulfills properties such as (hidden) concavity in the uncertainty (u in (1)) [6]. In the non-convex case, one may use a reformulation of a robust MINLP as polynomial optimization problem, which works if the contributing functions are polynomials and the uncertain parameters are contained in a semialgebraic set (see, e.g., [24, 25]). In this paper, we pursue a different direction. Rather than an exact reformulation approach or constraining functions to be polynomials, we choose a direct outer approximation approach. For this, we consider problems that are of convex type with respect to the decision variables (see Assumption 3.1). On the other hand, we allow for nonsmoothness, a general non-concave dependence on uncertainties and inexact worst case evaluations. Moreover, our only assumption for the uncertainty set is compactness.

Our approach then yields solutions that are robust feasible up to a tolerance and we consider both discrete and continuous decisions to be taken in the robust problem. The considered class of problems for example occurs in robust gas transport problems with discrete-continuous control decisions, nonlinear physical constraints and uncertainties in physics and demand.

In order to develop the algorithm, robust MINLPs are rewritten as nonsmooth MINLPs using an optimal value function of the adversarial problem. For an overview of state-of-the-art methods for nonsmooth MINLPs we refer to [13], where, among others, outer approximation approaches, extended level bundle methods and extended cutting plane methods are discussed. Our approach relies on the outer approximation concept to treat the nonsmooth MINLP. Outer approximation (OA) is an algorithm that is used for solving MINLPs in wide contexts. For an introduction and references, we refer to [17]. In the algorithm, a mixed-integer and typically linear master problem is solved to global optimality, as originally proposed in [10] and [14]. Iteratively, for fixed integral decisions, continuous subproblems are solved. Outer approximation for nonsmooth MINLP was first discussed in [12, 34, 35]. For the practical application of such a method, a concept for the solution of the arising nonsmooth subproblems is required. [9] suggests to use a proximal bundle method for the latter and demonstrates how appropriate cutting planes can be extracted at a solution.

Our Contribution Our approach follows the same lines, but we face an additional challenge: for a general non-convex adversarial problem, the determination of the worst case, which is required for the evaluation of the optimal value function, is itself not tractable in general. Thus, to achieve algorithmic tractability, we allow for *inexact* worst-case evaluations. In order to cope with this inexactness on the level of the subproblems, we modify an adaptive bundle method from [23], which was recently developed for the solution of nonlinear robust optimization problems with continuous variables. Due to the inexactness, in contrast to [9], we only have access to an outer approximation of the exact subdifferential. Nevertheless, we are able to show that cutting planes for the outer approximation can be extracted, which are valid up to a quantifiable error. With this, we are able to prove correctness and finite convergence of the OA method in the presence of inexactness. In detail, we are able to guarantee that the approximate solution determined by our OA algorithm is optimal up to a given tolerance. Moreover, the robust constraints are satisfied up to a tolerance, which is determined by the inexactness in the worst-case evaluation. The OA algorithm with the adaptive bundle method is outlined as a general algorithm independent from algorithmic details on the approximate solution of the adversarial problem. Here, we use piecewise linear relaxations of non-convexities and solve them via a mixed-integer linear optimizer. However, we point out that our approach can also be used with alternative methods that find an approximate worst case. To evaluate the performance of the novel algorithm, we specify it for the robust gas transport problem with discrete-continuous decisions. We demonstrate its efficiency by showing that our approach efficiently solves large realistic instances that could not be solved before.

We note that another avenue to treat inexactness in MINLP problems is described in [1], where an inexact version of a Benders' decomposition is used. The combinatorial setting considered there allows for binary decisions and continuous subproblems are allowed to be solved inexactly by an oracle. In contrast to our method, finite convergence is ensured via no-good cuts. The oracle's response then only has to result in valid inequalities that do not necessarily cut off the current iterate. Also for smooth MINLPs, alternative concepts exist, which can handle inexactness. Among them is the one in [27], where approximately fulfilled optimality conditions for the subproblems are required.

Structure This work is structured as follows. Although the presented algorithm is fully general, we prefer to start with an example application that falls into the considered class of problems in order to ease understanding of the subsequently introduced technical considerations. Thus, in Sect. 2, we briefly introduce the robust gas transport problem. In Sect. 3, we then derive the general setting of a nonsmooth MINLP that models a robust MINLP and present the framework of an OA method for this. The adaptive bundle method for continuous subproblems and resulting optimality conditions are presented in Sect. 4. In Sect. 5, we derive an OA algorithm that can deal with inexactness in function values, subgradients, and hence cutting planes obtained from subproblem solutions. The type of inexactness thereby matches our results for the bundle method's output. We also prove convergence of the OA algorithm. Finally, we present and discuss computational results for the gas transport problem in Sect. 6.

2 An Example Application for the Class of Problems Studied Here

We consider the stationary discrete-continuous gas transport problem, see [21], under uncertainties. A decomposition approach for the continuous robust two-stage gas transport problem is presented in [3] and a set containment approach for deciding robust feasibility of this problem is proposed in [4].

In this problem, we aim to find a control of active elements, such as compressors, valves or control valves, that minimizes the costs while ensuring that all demands are satisfied and that no technical or physical constraints are violated. Feasibility needs to be maintained even under uncertainties in demand and pressure loss coefficients.

A gas network is modeled by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with $|\mathcal{V}| = n$, $|\mathcal{A}| = m$ and an incidence matrix $A \in \{-1, 0, 1\}^{n \times m}$. The arcs model pipes, compressors and valves. A state variable $q \in \mathbb{R}^m$ denotes the gas flow, $d \in \mathbb{R}^n$ denotes the given balanced demand and flow conservation must hold: $Aq = d$. Squared pressure values at the nodes are denoted by $\pi \in \mathbb{R}^n$ and must fulfill bounds. For one root node $r \in \mathcal{V}$, the pressure value is assumed to be fixed. The pressure change at compressors is associated with a convex and differentiable cost function $w(\cdot)$, which is to be minimized.

The pressure loss on an arc $a \in \mathcal{A}$, i.e., the difference between squared pressures at connected nodes depends on the type of arc and we distinguish between pipes and compressors. The pressure losses on pipes depend on the flow values and directions as well as on pressure loss coefficients $\lambda_a > 0$. In detail, we have for every pipe $a = (u, v)$ the non-convex behavior [21]

$$\pi_v - \pi_u = -\lambda_a q_a |q_a|. \tag{2}$$

For compressors, we use a linear compressor model where a pressure loss is assigned to every compressor $a \in \mathcal{A}$ and depends on continuous and binary decision variables, x and y, respectively. The binary variables y determine if a compressor is active and the continuous variables x determine the pressure increase at active compressors. The pressure loss at every active compressor $a = (u, v)$ is then evaluated as

$$\pi_v - \pi_u = x \cdot y, \tag{3}$$

which leads to a non-convex cost function $w(x \cdot y)$. Compressors in bypass mode and open valves both behave like pipes with no pressure loss. We have further binary decisions y on the opening of valves.

We robustly protect against uncertainties in demand and pressure loss coefficients that are contained in a compact uncertainty set, i.e., $(d, \lambda) \in \mathcal{U}$. After uncertain parameters d and λ are realized, the second-stage state variables q and π uniquely adjust themselves by fulfilling flow conservation and the pressure loss constraints (2)–(3). We require that the pressure values are bounded both from above and from below by $\pi \in [\underline{\pi}, \overline{\pi}]$. Further, we can write the pressure values, due to their uniqueness, as a function of the decision variables (x, y) and the uncertain parameters (d, λ). This results in the following discrete-continuous robust gas transport problem.

$$\begin{aligned} \min_{x,y} \quad & w(x \cdot y) & & (RO_{gas}) \\ \text{s. t.} \quad & \underline{\pi}_v - \pi_v(x, y; d, \lambda) \le 0 & \forall (d, \lambda) \in \mathcal{U}, v \in \mathcal{V} & \\ & \pi_v(x, y; d, \lambda) - \overline{\pi}_v \le 0 & \forall (d, \lambda) \in \mathcal{U}, v \in \mathcal{V} & \\ & x \in [\underline{x}, \overline{x}],\ y \in \{0, 1\}^{n_y}. & & \end{aligned}$$

The function $\pi_v(\cdot)$ can be evaluated by solving a system of nonlinear and non-convex equations that involve, e.g., (2). This formulation relies on reformulation results in [3, 16]. Now, with

$$V_v\big(x, y, (d, \lambda)\big) := \max \Big\{\underline{\pi}_v - \pi_v(x, y; d, \lambda), \pi_v(x, y; d, \lambda) - \overline{\pi}_v\Big\}, \tag{4}$$

we can rewrite the robust constraints via one constraint by

$$G(x, y) = \max_{(d,\lambda)\in\mathcal{U}} \sum_{v\in\mathcal{V}} V_v^+\big(x, y, (d, \lambda)\big) \le 0, \tag{5}$$

where the superscript '+' denotes the positive part. With this, we write the discrete-continuous robust gas transport problem in (RO_{gas}) as

$$\begin{aligned} \min_{x,y} \quad & w(x \cdot y) \\ \text{s. t.} \quad & G(x, y) \le 0 \\ & x \in [\underline{x}, \overline{x}],\ y \in \{0, 1\}^{n_y}. \end{aligned} \tag{P_{gas}}$$

For the case that no compressor is part of a cycle, it turns out that the constraint function G is convex with respect to the continuous variable x. We refer to [2] for a discussion of the appropriateness of this assumption.

Lemma 2.1 *Under the assumption that no compressor is part of a cycle, the function $G(x, y)$ is convex in x.*

This lemma follows from the analysis in [3] and we omit the proof here.

We have outlined this example application already here in order to ease understanding of the subsequent sections where the general class of discrete-continuous robust nonlinear problems is defined and where we present the novel OA algorithm that is able to solve them.

3 Outer Approximation for Mixed-Integer Nonlinear Robust Optimization

We write a robust optimization problem with a compact uncertainty set $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ as

$$\begin{aligned} \min_{x,y} \quad & C(x, y) \\ \text{s. t.} \quad & V_i(x, y, u) \le 0 \quad \forall u \in \mathcal{U}, i \in \{1, \dots, n\} \\ & x \in X,\ y \in Y \cap \mathbb{Z}^{n_y}. \end{aligned} \tag{6}$$

The variables have dimensions n_x and n_y, respectively. We have that X is a full-dimensional box of form $X = [\underline{x}, \overline{x}] \subseteq \mathbb{R}^{n_x}$ and that Y is compact. Moreover, the objective function $C : \mathbb{R}^{n_x+n_y} \to \mathbb{R}$ and the constraint functions $V_i : \mathbb{R}^{n_x+n_y+n_u} \to \mathbb{R}$ are locally Lipschitz continuous and satisfy the following convexity-type assumption.

Assumption 3.1 The functions $C(\cdot, \cdot)$ and $V_i(\cdot, \cdot, u)$, for every $u \in \mathcal{U}$, $i \in [n]$, fulfill the following generalized convexity assumption when denoted by f. The function $f : X \times Y \cap \mathbb{Z}^{n_y} \to \mathbb{R}$ is convex with respect to x and it is true that for any pair $(x, y) \in X \times Y \cap \mathbb{Z}^{n_y}$, there exists a joint subgradient (s^x, s^y) such that the following subgradient inequality is satisfied:

$$f(x,y) + \begin{pmatrix} s^x \\ s^y \end{pmatrix}^T \begin{pmatrix} \bar{x} - x \\ \bar{y} - y \end{pmatrix} \leq f(\bar{x}, \bar{y}) \quad \forall (\bar{x}, \bar{y}) \in X \times Y \cap \mathbb{Z}^{n_y}. \tag{7}$$

A sufficient condition for this assumption is joint convexity of the function f on $X \times Y$. The other way round, our assumption only implies convexity in the continuous variable x. It is worth mentioning that, when we rely on Assumption 3.1, we have to make sure that all subgradients we use indeed satisfy inequality (7), while this is automatically true if convexity is assumed. More generally, it also suffices to specify how to derive subgradients that fulfill (7). This covers the setting of the gas transport, in which the functions are convex in the continuous decisions (see Lemma 2.1) and, despite a lack of convexity (see (3)), one can derive subgradients with respect to the binary decisions that fulfill (7). Further, here, we do not require convexity or concavity in the uncertain parameter u. This hence covers the gas transport setting with the non-convex dependence of pressure values on the uncertain parameters. Now, we reformulate the robust optimization problem (6) as a nonsmooth MINLP with finitely many constraints using the nonsmooth function

$$G(x,y) := \max_{u \in \mathcal{U}} V(x,y,u) := \max_{u \in \mathcal{U}} \sum_{i=1}^{n} V_i^+(x,y,u)$$

as a constraint function. We obtain

$$\begin{aligned} \min_{x,y} \quad & C(x,y) \\ \text{s.t.} \quad & G(x,y) \leq 0 \\ & x \in X,\, y \in Y \cap \mathbb{Z}^{n_y}. \end{aligned} \tag{P}$$

We note that the assumed generalized convexity of the functions $V_i(x, y, u)$ directly carries over to G. To evaluate G, it is necessary to solve an adversarial problem that determines a worst-case parameter maximizing the constraint violation. To make this concept clear, we mention that in the robust gas transport problem, the adversarial problem, i.e., to evaluate the function G in (4)–(5), is to find for fixed control decisions a realization of demand and physical parameters that maximizes the violation of pressure bounds.

The goal is to solve the MINLP (P) via an outer approximation approach. We sketch the general framework of an OA method for (P) here by closely following [14, 35]. In an OA method, a master problem and a subproblem are solved in every iteration. The master problem is a mixed-integer linear problem (MIP) that is a relaxation of the original problem (P). Solving an MIP is in general NP-hard. However, many algorithmic enhancements were developed so that MIPs can typically be solved to global optimality by modern available solvers, even for large instances (see, e.g., [7]). The linear relaxation of the original problem (P) in iteration K is the master problem:

$$\begin{aligned} \min_{x,y,\theta} \quad & \theta && (MP^K) \\ \text{s.t.} \quad & \theta \leq \Theta^K - \epsilon_{oa} \\ & \textit{[linearized objective function]} \leq \theta \\ & \textit{[linearized constraint function]} \leq 0 \\ & x \in X,\ y \in Y \cap \mathbb{Z}^{n_y}, \end{aligned}$$

where $\Theta^K \in \mathbb{R}$ denotes the objective value of the current best known solution and $\epsilon_{oa} > 0$ is a previously fixed optimality tolerance as typically used in an OA method (e.g., by [14]). We detail below the linearized constraints, which are generated via function values and subgradients. After termination, one has detected infeasibility or has found a feasible ϵ_{oa}-optimal solution. By ϵ_{oa}-optimality, we mean that the objective value deviates from the optimal objective value by at most ϵ_{oa}.

Every OA iteration involves the solution of a subproblem where all integer variables are fixed and one determines best values only for the continuous variables. However, the subproblems are nonlinear. The solution of a subproblem, i.e., the resulting mixed-integer candidate solution, is then used to generate linearized constraints that are valid for all feasible solutions of the original problem (P) that still need to be considered. These constraints act as cutting planes that are added to the master problem (MP^K) and strengthen the relaxation of (P). Further, they are chosen such that every feasible integer assignment is visited only once, so that the OA method converges finitely with a global ϵ_{oa}-optimal solution to the original MINLP.

In each iteration, one candidate integer solution from the feasible set Y is fixed. For this fixed integer assignment, we solve a subproblem. This is either a continuous subproblem of (P) or, in the case of its infeasibility, a so-called feasibility problem that minimizes the violation of constraints. For a fixed integer assignment y_K, the continuous subproblem is

$$\begin{aligned} \min_{x \in X} \quad & C(x, y_K) && (NLP(y_K)) \\ \text{s.t.} \quad & G(x, y_K) = \max_{u \in \mathcal{U}} V(x, y_K, u) \leq 0. \end{aligned}$$

If the continuous subproblem is infeasible, we solve the feasibility problem, which minimizes the violation of the constraint $G(x, y) \leq 0$ and is written as

$$\min_{x \in X} G(x, y_K) = \min_{x \in X} \max_{u \in \mathcal{U}} V(x, y_K, u). \qquad (F(y_K))$$

Next, we detail the linearized constraints, i.e., the cutting planes in the master problem (MP^K). We first split the set of integer points in Y into two sets, depending on whether the corresponding continuous subproblem is feasible or not:

$$\begin{aligned} T &= \{y \in Y \cap \mathbb{Z}^{n_y} \mid (NLP(y_K)) \text{ is feasible}\}, \\ S &= \{y \in Y \cap \mathbb{Z}^{n_y} \mid (NLP(y_K)) \text{ is infeasible}\}. \end{aligned}$$

In the course of an OA algorithm, we collect the investigated integer points in subsets $T^K \subseteq T, S^K \subseteq S$. For fixed y_K, we denote by x_K a continuous solution to ($NLP(y_K)$) or ($F(y_K)$). To strengthen the relaxation of the master problem, we collect linearizations at the mixed-integer candidate solutions (x_K, y_K). In detail, we approximate the functions C and G (or only G in the case of infeasibility) by linearizations generated by function values and subgradients, (α_K, β_K), (ξ_K, η_K), evaluated at (x_K, y_K). In an iteration K, the linearized constraints (i.e., the constraints in (MP^K)) are of the form

$$\begin{aligned} C(x_J, y_J) + (\alpha_J^T, \beta_J^T) \begin{pmatrix} x - x_J \\ y - y_J \end{pmatrix} &\leq \theta && \forall y_J \in T^K \\ G(x_J, y_J) + (\xi_J^T, \eta_J^T) \begin{pmatrix} x - x_J \\ y - y_J \end{pmatrix} &\leq 0 && \forall y_J \in T^K \\ G(x_L, y_L) + (\xi_L^T, \eta_L^T) \begin{pmatrix} x - x_L \\ y - y_L \end{pmatrix} &\leq 0 && \forall y_L \in S^K. \end{aligned} \tag{8}$$

These cutting planes are then added to the master problem. To avoid cutting off an optimal solution to the original problem (P), the cutting planes must be valid in the following sense: they cut off a point only if it is infeasible or does not improve the current best objective value by more than ϵ_{oa}. Further, to ensure finite convergence of the algorithm, the cutting planes must cut off the current assignment of integer variables. To ensure this and hence correctness of an OA method, one usually requires in a nonsmooth setting that the function values and subgradients fulfill KKT conditions of the subproblem and therefore assumes that Slater's condition holds (see, e.g., [9, 35]). We proceed similarly and also assume that Slater's condition holds in the following form.

Assumption 3.2 If ($NLP(y_K)$) is feasible, then there is an $x \in int(X)$ with $V_i(x, y_K, u) < 0 \; \forall u \in \mathcal{U}, i \in [n]$.

To illustrate this assumption, we briefly concretize it for the gas transport problem from the preceding section: there, we require that for every possible realization of the uncertain parameters, i.e., demand and pressure loss coefficients, there exists a control of the active elements such that all pressure bounds are strictly fulfilled, i.e., $\pi \in (\underline{\pi}, \overline{\pi})$.

In the presented setting, the solution of the subproblems ($NLP(y_K)$) and ($F(y_K)$) is a challenging task that is not accessible by standard methods. In particular, we face non-concavities in the uncertain parameters so that the constraint function G, i.e., the adversarial problem $G(x, y) = \max_{u \in \mathcal{U}} V(x, y, u)$ for given x, y, may be only approximately evaluable. In the next section, we investigate which properties can be ensured for solutions to the subproblems in the presence of such inexact worst-case evaluations.

4 An Adaptive Bundle Method for the Continuous Subproblems

To solve continuous nonsmooth optimization problems, bundle methods are a practically efficient approach. As the latter are usually applied in an unconstrained setting, we write the subproblems as unconstrained problems and penalize constraint violation in the objective. Instead of ($NLP(y_K)$), we aim to solve the following unconstrained problem with an l_1-penalty term $P_X(x) = \sum_{i=1}^{n_x} \max\{0, x_i - \overline{x}, \underline{x} - x_i\}$ and sufficiently large penalty parameters $\psi, \psi_X > 0$:

$$\min_{x \in \mathbb{R}^{n_x}} \left(C(x, y_K) + \psi \max_{u \in \mathcal{U}} V(x, y_K, u) + \psi_X P_X(x) \right). \qquad (NLP_\psi(y_K))$$

The existence of finite penalty parameters is ensured by Assumption 3.2. In particular, it ensures the existence of finite penalty parameters such that a point x_K solves ($NLP_\psi(y_K)$) if and only if it solves ($NLP(y_K)$). This can be seen by using, e.g., [33, Theorem 7.21] and we omit the details here. To treat ($F(y_K)$), we also use a penalty formulation with a sufficiently large penalty parameter ψ_X:

$$\min_{x \in \mathbb{R}^{n_x}} \max_{u \in \mathcal{U}} V(x, y_K, u) + \psi_X P_X(x). \qquad (F_\psi(y_K))$$

In practice, we simply choose certain penalty parameters ψ, ψ_X at the beginning, which are then increased if required (see end of this section). We present in this section an algorithm for the approximate solution of the penalized subproblems. We first write the objective functions in the abstract form

$$\min_{x \in \mathbb{R}^{n_x}} f(x) = \min_{x \in \mathbb{R}^{n_x}} \max_{u \in \mathcal{U}} v(x, u), \qquad (9)$$

where

$$\begin{aligned} f(x) &= C(x, y_K) + \psi G(x, y_K) + \psi_X P_X(x), \\ v(x, u) &= C(x, y_K) + \psi V(x, y_K, u) + \psi_X P_X(x). \end{aligned}$$

For the feasibility problem ($F_\psi(y_K)$), we set $C \equiv 0$ and $\psi = 1$, so that no separate discussion is required. We further note that the integer variable y_K is fixed during the solution of a subproblem, so that we largely omit it in the remainder of this section.

Due to the unconstrained and nonsmooth character of (9), in principal a proximal bundle type method can be applied for its solution. However, as (9) is solved in the context of an outer approximation scheme, not only an (approximate) solution x_K of (9) is required, but also cutting planes in the sense of (8) have to be extracted. While in a continuously differentiable setting, appropriate cutting planes can be determined a posteriori by computing the gradients of the objective and constraints at the solution x_K, in a nonsmooth setting, the situation is more involved. Roughly speaking, the reason is that the subgradients at the solution x_K are not unique and one has to choose them such that they fulfill first-order optimality conditions. To overcome this, suitable sequences of subgradients have to be constructed while the bundle algorithm is carried out. This is demonstrated in [9] using an exact penalization proximal bundle algorithm. While in principle the same idea can be applied to our setting, we are facing the additional difficulty that every evaluation of the function f requires the solution of the adversarial problem

$$\max_{u \in \mathcal{U}} V(x, y, u). \tag{10}$$

This is in general a hard problem if the constraint functions V_i are non-concave in the uncertain parameter u as in the robust gas transport problem from Sect. 2. Thus, we follow [23] and use in the following the relaxed assumption that problem (10) can only be solved with a prescribed finite precision $\epsilon_f > 0$. This means that there is an oracle, which provides for every given x and ϵ_f an approximate worst case

$$u_x \in \{u_x \in \mathcal{U} \mid v(x, u_x) \geq \max_{u \in \mathcal{U}} v(x, u) - \epsilon_f\}, \tag{11}$$

so that $v(x, u_x) \geq f(x) - \epsilon_f$. Using this, we can define an overestimator for f as

$$f_a(x) := v(x, u_x) + \epsilon_f. \tag{12}$$

Furthermore, as a consequence of the inexactness, instead of an element from the exact Clarke subdifferential of f at x, the best we can hope for is the following approximate subgradient of f (see [23]).

$$g_x \in \partial_x v(x, u_x).$$

This is an element of the Clarke subdifferential of $v(\cdot, u_x)$ with u_x defined by the choice in (11). This approximate subgradient lies in the following set, which is the convex hull that contains $\partial_x v(x, u_x)$ and the subdifferentials for all uncertain parameters that better approximate the worst case.

$$\tilde{\partial}_a f(x) := \text{conv}\{g \mid g \in \partial_x v(x, u), u \in \mathcal{U},\ v(x, u) \geq v(x, u_x)\}. \tag{13}$$

This set can be interpreted as an outer approximation of the exact Clarke subdifferential at x. In [23], an adaptive proximal bundle method is suggested for this setting. This algorithm—as most bundle methods—generates a sequence of serious and trial iterates. The serious iterates form a sequence, which approach an approximate solution of (9). For each serious iterate, one generates a sequence of trial iterates that improve the local approximation of f around the serious iterate by information about function values and subgradients. The algorithm is able to work with elements from the approximate subdifferential (13) if the error in (12) is chosen as follows: given a current trial iterate x^k (with inner loop counter k) and current serious iterate x, the error bound ϵ_f^k for $f_a(x^k)$ is set to

$$\epsilon_f^k = \epsilon'' \|x - x^k\|^2. \tag{14}$$

Here, ϵ'' is a previously chosen algorithmic parameter. For this, the following convergence result is derived in [23].

Corollary 4.1 *[23, Corollaries 2, 3, 4] Let x_1 be such that $\Omega := \{x \in \mathbb{R}^{n_x} : f(x) \leq f_a(x_1)\}$ is bounded and let f be lower C^1. Let us use [23, Algorithm 1] with a stopping tolerance $\tilde{\epsilon} = 0$ in line 3 there. Then two different situations can occur:*

(1) The algorithm stops with a serious iterate x_j after finitely many iterations. Then we set $\bar{x} := x_j$.

(2) All inner loops terminate finitely, but the outer loop does not terminate finitely. In this case, the sequence of serious iterates x_j has at least one accumulation point; we denote this by $\bar{x}$.

In either of the cases, it holds that $0 \in \tilde{\partial}_a f(\bar{x})$.

Now, applying the adaptive bundle method to problem (9) for an approximate solution x_K, we obtain $0 \in \tilde{\partial}_a f(x_K)$ or—exploiting the structure of f—$0 \in \tilde{\partial}_a(C + \psi G + \psi_X P_X)(x_K)$. Given that C and P_X do not depend on u and by exploiting the convexity of C, G and P_X with respect to x, this can be rewritten as

$$0 \in \partial C(x_K) + \psi \tilde{\partial}_a G(x_K) + \psi_X \partial P_X(x_K). \tag{15}$$

The challenge for the remainder of this section is now to construct elements $\alpha \in \partial C(x_K)$, $\tilde{\xi} \in \tilde{\partial}_a G(x_K)$ and $\zeta \in \partial P_X(x_K)$, such that

$$\alpha + \psi \tilde{\xi} + \psi_X \zeta = 0, \tag{16}$$

i.e., to construct suitable (approximate) subgradients of C, G and P_X that realize the optimality condition (15). It will turn out later that α and $\tilde{\xi}$ can then be used to construct cutting planes in the sense of (8). As already outlined above, such realizations cannot be computed solely on the basis of the knowledge of x_K. Rather than this, a deeper insight into the algorithm from [23] and a couple of modifications, which we outline in the sequel, are required. The full modified algorithm is detailed in Algorithm 2 in Appendix A. We first require that whenever in Algorithm 2 an approximate subgradient of f at a point x is evaluated, it is computed as

$$g_x = s_C^x + \psi s_G^x + \psi_X s_X^x, \tag{17}$$

where $s_C^x \in \partial C(x)$, $s_G^x \in \partial_x V(x, u_x)$, $s_X^x \in \partial P_X(x)$, and u_x is an approximate worst case in the sense of (11). This is realized in lines 20-21 of Algorithm 2.

Next, we make use of the so-called *aggregate subgradient*, which—together with (17)—will be the key for deriving α and $\tilde{\xi}$ in (16) and played already a crucial role in the convergence proof in [23]. To introduce this, we repeat the definition of the convex working model used in [23]. This is, at a serious iterate x, the piecewise linear function

$$\phi_k(\cdot, x) := \max\{m_l(\cdot, x) \mid 0 \leq l \leq k-1\}.$$

Here, $m_l(\cdot, x)$ are cutting planes that are generated using approximate function values and subgradients. More precisely,

$$m_0(\cdot, x) := f_a(x) + {g_x}^T(\cdot - x),$$

with $g_x \in \partial_x v(x, u_x)$, is a so-called exactness plane at the serious iterate x, and

$$m_l(\cdot, x) := t_l(\cdot) - r_l,\ l = 1, \dots, k-1$$

are cutting planes at trial iterates x^l, where $t_l(\cdot)$ is the tangent plane at x^l and r_l a downshift with respect to the serious iterate x. Then, in every iteration the following convex problem is solved in order to generate a new trial iterate x^k.

$$\min_{z \in \mathbb{R}^{n_x}} \phi_k(z, x) + (z-x)^\top Q_x (z-x) + \frac{\tau_k}{2}\|z-x\|^2. \tag{18}$$

Here, $\tau_k > 0$ is a proximity control parameter and Q_x a symmetric matrix, which can be used to model second-order information. Now the aggregate subgradient g_k^* is defined as

$$g_k^* = (Q_x + \tau_k \mathcal{I})(x - x^k).$$

As x^k is the unique minimizer of (18), g_k^* is a subgradient of the current working model, i.e., $g_k^* \in \partial \phi_k(x^k, x)$.

Now, we carry out the following three steps: First, we see that g_k^* tends to 0 along a suitable subsequence. The claim is detailed in Lemma 4.1. We omit its proof as the latter is a straight-forward combination of arguments from the convergence analysis in [23]. Based on this, we introduce a stopping criterion for the bundle method. Second, we show that every g_k^* can be split according to the partition in (17) and construct the output of the bundle method. Third, we prove in Theorem 4.1 that the limits of the individual parts of g_k^* satisfy the optimality condition (16). This implies that the modified bundle method, with the proposed stopping criterion and output, generates subgradients that fulfill the optimality condition (16).

Lemma 4.1 *[23, proofs of Lemma 6, Theorem 1, Corollaries 3, 4]; [28, proofs of Lemma 4, 7] We use the same assumptions as in Corollary 4.1 and use [23, Algorithm 1] without any stopping criterion. Then one of the following situations occur:*

(1) The inner loop at a serious iterate x does not terminate finitely. Then $x^k \to x$, and there is a subsequence of the inner loop indices k such that $\phi_k(x^k, x) \to f_a(x)$, and $g_k^* \to 0 \in \tilde{\partial}_a f(x)$.

(2) All inner loops terminate finitely, but the outer loop does not terminate finitely. The sequence of all trial and serious iterates is bounded and there is a choice of pairs of outer and inner loop indices $(j, k(j))$ and a subsequence of the outer loop indices j such that $x_j \to \bar{x}$, $x^{k(j)} \to \bar{x}$, $\phi_{k(j)}(x^{k(j)}, x_j) \to f_a(\bar{x})$ and $g_{k(j)}^* \to 0 \in \tilde{\partial}_a f(\bar{x})$.

Now, in order to make sure that we stop the proximal bundle algorithm at a point, where the aggregate subgradient is small, we apply the following modification: rather than using the abstract convergence criterion $0 \in \tilde{\partial}_a f(x)$, we apply the criterion $\|g_k^*\| \leq \epsilon^*$ with a tolerance $\epsilon^* = 0$ (see Algorithm 2, line 8). Hence, in the cases in which the bundle algorithm does not terminate finitely, we have a sequence of aggregate subgradients that converges to $g^* = 0 \in \tilde{\partial}_a f(\bar{x})$. In the case of finite termination, we find $g_k^* = 0$ for a finite k.

Next we derive a partition of an aggregate subgradient g_k^*, which is analogous to the partition by which subgradients were derived in (17). In order to do so, we first denote the subgradients at former trial iterates x^l by g_l, so that $g_l \in \tilde{\partial}_a f(x^l)$, and elements of the exactness plane via the index $l = 0$. As $\phi_k(\cdot, x)$ is convex and piecewise affine linear with slopes g_l, the aggregate subgradient $g_k^* \in \partial \phi_k(x^k, x)$ is a convex combination of subgradients g_l with $l < k$:

$$g_k^* = \sum_{l=0}^{k-1} \lambda_l^k g_l, \quad \sum_{l=0}^{k-1} \lambda_l^k = 1. \tag{19}$$

Further, $\lambda_l^k > 0$ for an $l \in \{0, \ldots, k-1\}$ only if $m_l(x^k, x) = \phi_k(x^k, x)$. As in the course of the former bundle iterations, $g_l \in \tilde{\partial}_a f(x_l)$ with $l < k$ has been computed via (17), we have the following partitions.

$$g_l = s_C^l + \psi s_G^l + \psi_X s_X^l, \quad s_C^l \in \partial_x C(x^l), s_G^l \in \tilde{\partial}_a G(x^l), s_X^l \in \partial P_X(x^l). \tag{20}$$

Accordingly, we define, denoting by j the outer loop counter of the current serious iterate x,

$$\alpha_j^k = \sum_{l=0}^{k-1} \lambda_l^k s_C^l, \qquad \tilde{\xi}_j^k = \sum_{l=0}^{k-1} \lambda_l^k s_G^l, \qquad \zeta_j^k = \sum_{l=0}^{k-1} \lambda_l^k s_G^l. \tag{21}$$

We note that the weights $\lambda_l^k, l \in \{0, \ldots, k-1\}$ can be computed by solving a linear system of equations in line 11 of Algorithm 2.

In practice, we choose a $\epsilon^* > 0$ and stop the bundle method as soon as $\|g_k^*\|$ is sufficiently small at an iteration k with a solution $x_K = x^k$. As subgradients for the current iteration K of the outer approximation algorithm, we then choose

$$\alpha_K = \alpha_j^k, \quad \tilde{\xi}_K = \tilde{\xi}_j^k. \tag{22}$$

In Theorem 4.1, we finally formalize and justify this choice from a theoretical perspective, i.e., for $\epsilon^* = 0$. In detail, we show that the choice (22) is correct in the sense of the optimality condition (16) if we have $\|g_k^*\| = 0$ for a finite k. Moreover, we prove that in all cases where the algorithm does not finitely converge, we can define a suitable limit of (22) instead. We distinguish the following three cases of output of the bundle method.

(1) In the case that the algorithm converges finitely within an inner loop at a serious iterate x_j with $\|g_k^*\| = 0$, we choose

$$\begin{aligned} x_K &= x_j, \ \epsilon_G^K = \epsilon_f^{x_j}, \ \tilde{G}(x_K, y_K) = V(x_j, y_K, u_{x_j}), \\ \alpha_K &= \alpha_j^k, \ \tilde{\xi}_K = \tilde{\xi}_j^k, \ \zeta_K = \zeta_j^k. \end{aligned} \tag{23}$$

(2) In the case that the algorithm converges with infinitely many iterations in an inner loop at a serious iterate x_j, we choose a subsequence of iterates k as in Lemma 4.1, (1), and set

$$\begin{aligned} &x_K = x_j, \quad \epsilon_G^K = \epsilon_f^{x_j}, \quad \tilde{G}(x_K, y_K) = V(x_j, y_K, u_{x_j}), \\ &\alpha_K = \lim_{k\to\infty} \alpha_j^k, \quad \tilde{\xi}_K = \lim_{k\to\infty} \tilde{\xi}_j^k, \quad \zeta_K = \lim_{k\to\infty} \zeta_j^k. \end{aligned} \tag{24}$$

(3) In the case that the algorithm does not converge in an inner loop, we choose a cluster point of the serious iterates as a solution. We choose a cluster point $\bar{x}$ and a sequence of indices $k(j)$ as in Lemma 4.1, (2):

$$\begin{aligned} &x_K = \bar{x}, \quad \epsilon_G^K = 0, \quad \tilde{G}(x_K, y_K) = \lim_{j\to\infty} V(x_j, y_K, u_{x_j}), \\ &\alpha_K = \lim_{j\to\infty} \alpha_j^{k(j)}, \quad \tilde{\xi}_K = \lim_{j\to\infty} \tilde{\xi}_j^{k(j)}, \quad \zeta_K = \lim_{j\to\infty} \zeta_j^{k(j)}. \end{aligned} \tag{25}$$

It is noted that in (1) and (2) above, $\epsilon_f^{x_j}$ denotes the error, which was required to compute the *serious* iterate x_j (see (14)). The choices in (23), (24) and (25) are justified by the following result.

Theorem 4.1 *The adaptive bundle method with subgradient generation in Algorithm 2 with* $\epsilon^* = 0$*, with (23), (24) and (25) as output, and under the assumptions as in Corollary 4.1, generates subgradients* $\alpha_K \in \partial C(x_K)$*,* $\tilde{\xi}_K \in \tilde{\partial}_a G(x_K)$ *and* $\zeta_K \in \partial P_X(x_K)$ *with* $\alpha_K + \psi\tilde{\xi}_K + \psi_X \zeta_K = 0$*.*

One of the key tools for the proof of this theorem is a suitable choice of a convex overestimator of the working model $\phi_k(\cdot, x)$ at a serious iterate x. In detail, we use as an overestimator the point-wise supremum of all cutting planes that could be generated at potential trial iterates z. These take the form $m_z(\cdot, x) = f_a(z) + {g_z}^T(\cdot - z) - r_z$, where $f_a(z)$ and $g_z \in \partial_x v(z, u_z)$ are approximate function values and approximate subgradients at z, and r_z a downshift with respect to x. With these, the overestimator is defined as

$$\begin{aligned} \phi(\cdot, x) := \sup\Big\{ &\big\{m_z(\cdot, x) | u_z \in \mathcal{U}, v(z, u_z) \geq f(z) - \epsilon''\|x - z\|^2, z \in B(0, M) \setminus \{x\}\big\} \\ &\cup \big\{m_0(\cdot, x) | {u_x}' \in \mathcal{U}, v(x, {u_x}') = v(x, u_x)\big\}\Big\}, \end{aligned} \tag{26}$$

where M is chosen such that x and all possible trial iterates lie in $B(0, M)$ (see [23]). In an analogous way, we define an overestimator for G only:

$$\begin{aligned} \phi^G(\cdot, x) := \sup\Big\{ &\big\{m_z^G(\cdot, x) | u_z \in \mathcal{U}, v(z, u_z) \geq f(z) - \epsilon''\|x - z\|^2, z \in B(0, M) \setminus \{x\}\big\} \\ &\cup \big\{m_0^G(\cdot, x) | {u_x}' \in \mathcal{U}, v(x, {u_x}') = v(x, u_x)\big\}\Big\}, \end{aligned} \tag{27}$$

with cutting planes $m_z^G(\cdot, x) = G_a(z) + {s_G^z}^T(\cdot - z) - r_z$ at z with approximate function value $G_a(z)$ and approximate subgradient $s_G^z \in \partial_x\big(\sum_{i=1}^n V(z, u_z)\big)$. Before we move on to the proof of Theorem 4.1, we import an auxiliary result from [23].

Lemma 4.2 *[23, Lemma 3] For the first-order model $\phi(\cdot, x)$, as defined in (26), the following properties hold:*

(1) $\phi(x, x) = f_a(x)$,
(2) $\partial\phi(x, x) \subseteq \tilde{\partial}_a f(x)$,
(3) $\phi_k(\cdot, x) \leq \phi(\cdot, x)$ for the working model $\phi_k(\cdot, x)$.

The same holds if we replace ϕ by the first-order model ϕ^G defined in (27) and f by G.

Proof of Theorem 4.1 We divide the proof into two claims concerning convergence in the inner or the outer loop, respectively.

Claim 4.1 *If the inner loop at a serious iterate $x = x_j$ does not terminate finitely, then there is a subsequence of the indices k such that the limits of subsequences $\alpha := \lim_{k\to\infty} \alpha_j^k$, $\tilde{\xi} := \lim_{k\to\infty} \tilde{\xi}_j^k$ and $\zeta := \lim_{k\to\infty} \zeta_j^k$ exist and fulfill*

$$\alpha \in \partial C(x),\ \tilde{\xi} \in \tilde{\partial}_a G(x),\ \zeta \in \partial P_X(x),\ 0 = \alpha + \psi\tilde{\xi} + \psi_X\zeta. \tag{28}$$

Proof We recall that the aggregate subgradients at the trial iterates x^k are given by (19)-(20). From Lemma 4.1, (1), we have $x^k \to x$, so that, with local boundedness of the Clarke subdifferential [8] and compactness of $\mathcal{U}$, we obtain boundedness of the sequences $(s_C^k)_k$, $(s_G^k)_k$, $(s_X^k)_k$. The sequences $(\alpha_j^k)_k$, $(\tilde{\xi}_j^k)_k$, $(\zeta_j^k)_k$ as their respective convex combinations thus are bounded as well. For the aggregate subgradient g_k^*, it holds that

$$g_k^* = \alpha_j^k + \psi\tilde{\xi}_j^k + \psi_X\zeta_j^k. \tag{29}$$

By Lemma 4.1, (1), g_k^* converges, passing to a subsequence, to 0. We pass to a suitable subsequence of this such that the limits α, $\tilde{\xi}$ and ζ are well-defined. Further, it follows from (29) that $0 = \alpha + \psi\tilde{\xi} + \psi_X\zeta$.

It remains to prove the first part of (28). We use the definition of the first-order models in (26) and (27). As the functions C and the l_1-penalty function P_X are convex and not affected by inexactness, we have

$$\phi(\cdot, x) = C(\cdot) + \psi\phi^G(\cdot, x) + \psi_X P_X(\cdot). \tag{30}$$

This first-order model ϕ is approximated by the working model $\phi_k(\cdot, x)$, which is refined in every iteration. At the iterate x^k, we have $\phi_k(x^k, x) = m_l(x^k, x)$ for l with $\lambda_k^l > 0$, so that the working model's value can be written as $\phi_k(x^k, x) = \sum_{l=0}^{k-1} \lambda_l^k m_l(x^k, x)$. We now partition the cutting planes $m_l(\cdot, x)$ analogously to (17). First, we recall that every cutting plane is generated by a tangent plane and a downshift term: $m_l(\cdot, x) = t_l(\cdot) - r_l$. The slope of the tangent plane $t_l(\cdot)$ is g_l, derived via $g_l = s_C^l + \psi s_G^l + \psi_X s_X^l$. As we have this partition at hand, we can partition the cutting plane $m_l(\cdot, x)$ into

$$m_l(\cdot,x) = m_l^C(\cdot,x) + \psi m_l^G(\cdot,x) + \psi_X m_l^X(\cdot,x),$$

with $m_l^C(\cdot,x) = t_l^C(\cdot)$, $m_l^G(\cdot,x) = t_l^G(\cdot) - r_l$ and $m_l^X(\cdot,x) = t_l^X(\cdot)$. The slopes of these cutting planes are s_C^l, s_G^l and s_X^l, respectively. Further, they underestimate $C(\cdot)$, $\phi^G(\cdot,x)$ and $P_X(\cdot)$, respectively, so that the following subgradient inequalities hold for every $l < k$ and any z:

$$m_l^C(x^k,x) + s_C^l(z - x^k) \leq C(z), \tag{31a}$$

$$m_l^G(x^k,x) + s_G^l(z - x^k) \leq \phi^G(z,x), \tag{31b}$$

$$m_l^X(x^k,x) + s_X^l(z - x^k) \leq P_X(z). \tag{31c}$$

From this, we derive subgradient inequalities for the functions $C(\cdot)$, $\phi^G(\cdot,x)$ and $P_X(\cdot)$ as follows: We multiply the inequalities by the corresponding factors λ_l^k, sum over $l < k$ and add a zero-term to the left-hand side. It follows for any z that

$$C(x^k) - C(x^k) + \sum_{l=0}^{k-1} \lambda_l^k m_l^C(x^k,x) + \sum_{l=0}^{k-1} \lambda_l^k s_C^l(z - x^k) \leq C(z), \tag{32a}$$

$$\phi^G(x^k,x) - \phi^G(x^k,x) + \sum_{l=0}^{k-1} \lambda_l^k m_l^G(x^k,x) + \sum_{l=0}^{k-1} \lambda_l^k s_G^l(z - x^k) \leq \phi^G(z,x), \tag{32b}$$

$$P_X(x^k) - P_X(x^k) + \sum_{l=0}^{k-1} \lambda_l^k m_l^X(x^k,x) + \sum_{l=0}^{k-1} \lambda_l^k s_X^l(z - x^k) \leq P_X(z). \tag{32c}$$

Using the convex ϵ-subdifferential (see, e.g., [5]) and (21), we have

$$\alpha_j^k \in \partial^{[\epsilon_C^k]} C(x^k), \quad \epsilon_C^k = C(x^k) - \sum_{l=0}^{k-1} \lambda_l^k m_l^C(x^k,x), \tag{33a}$$

$$\xi_j^k \in \partial^{[\epsilon_G^k]} \phi^G(x^k,x), \quad \epsilon_G^k = \phi^G(x^k,x) - \sum_{l=0}^{k-1} \lambda_l^k m_l^G(x^k,x), \tag{33b}$$

$$\zeta_j^k \in \partial^{[\epsilon_X^k]} P_X(x^k), \quad \epsilon_X^k = P_X(x^k) - \sum_{l=0}^{k-1} \lambda_l^k m_l^X(x^k,x). \tag{33c}$$

Further, we have for every $l < k$ with $\lambda_l^k > 0$ that $\phi_k(x^k,x) = m_l^C(x^k) + \psi m_l^G(x^k) + \psi_X m_l^X(x^k)$.

By (30), it thus holds that

$$\phi(x^k,x) - \phi_k(x^k,x) = \epsilon_C^k + \psi \epsilon_G^k + \psi_X \epsilon_X^k. \tag{34}$$

From Lemma 4.1, (1), we have that $x^k \to x$. Further, from Lemma 4.1, (1), and Lemma 4.2, (1), it follows that $\phi_k(x^k,x) \to f_a(x) = \phi(x,x)$. Thus, $\phi(x^k,x) -$

$\phi_k(x^k, x) \to 0$. As the summands on the right side of (34) are non-negative, they also converge to 0.

We thus have convergence of $\epsilon_C^k, \epsilon_G^k$ and ϵ_X^k to 0, convergence of $x^k \to x$, and convergence of $\alpha_j^k, \tilde{\xi}_j^k$ and ζ_j^k to $\alpha, \tilde{\xi}$ and ζ, respectively. With (33), it thus follows that $\alpha \in \partial C(x), \tilde{\xi} \in \partial\phi^G(x, x)$ and $\zeta \in \partial P_X(x)$ (see, e.g., [20, Proposition 4.1.1]). The claim $\tilde{\xi} \in \tilde{\partial}_a G(x)$ follows by the result $\phi(x, x) \subseteq \tilde{\partial}_a f(x)$ from Lemma 4.2, (2), which holds analogously for ϕ^G. □

We note that the proof for the case of finite termination (i.e., (23)) follows as a special case of the proof of Claim 4.1. From the definition of the aggregate subgradient g_k^*, we see that $g_k^* = 0$ implies that the trial iterate x^k and the serious iterate x coincide. With this, we can follow the proof of Claim 4.1 until we arrive at formula (33). Now, using $x^k = x$, it is straight-forward to show that $\epsilon_C^k = \epsilon_G^k = \epsilon_X^k = 0$ without passing to the limit.

Claim 4.2 *If the outer loop does not terminate finitely with serious iterates x_j and a cluster point $\bar{x}$, then there is a choice of a subsequence of inner loop indices $k(j)$ such that the limits of subsequences $\alpha := \lim_{j\to\infty} \alpha_j^{k(j)}$, $\tilde{\xi} := \lim_{j\to\infty} \tilde{\xi}_j^{k(j)}$ and $\zeta := \lim_{j\to\infty} \zeta_j^{k(j)}$ exist and fulfill*

$$\alpha \in \partial C(\bar{x}),\ \tilde{\xi} \in \tilde{\partial}_a G(\bar{x}),\ \zeta \in \partial P_X(\bar{x}),\ 0 = \alpha + \psi\tilde{\xi} + \psi_X\zeta.$$

Proof The proof can be conducted by following the proof of Claim 4.1. Minor changes have to be made as follows: We do not consider a fixed serious iterate but rather a sequence (x_j) of serious iterates with a cluster point $\bar{x}$. As a sequence of trial iterates, we consider a sequence $(x^{k(j)})_j$ that has the properties as ensured by Lemma 4.1, (2). This sequence has $\bar{x}$ as a cluster point and it holds by Lemma 4.1, (2), and Lemma 4.2, (1), that $\phi_{k(j)}(x^{k(j)}, x_j) \to f_a(\bar{x}) = \phi(\bar{x}, \bar{x})$. Finally, the claim follows analogously to the proof of Claim 4.1 □

With this, we have handled all cases in Theorem 4.1. □

Remark 4.1 From convexity of C and V_i $(i \in [n])$ in x, we infer that the approximate subdifferential $\tilde{\partial}_a f(\cdot)$ is contained in a convex ϵ-subdifferential (see, e.g., [29]): with $g_x \in \partial_x v(x, u_x)$ and denoting the error in the evaluation of $G(x)$ by $\epsilon_G^p = f(x) - v(x, u_x)$, we have

$$v(x, u_x) + g_x^T(z - x) \le v(z, u_x)\ \forall z \in \mathbb{R}^{n_x} \Rightarrow\ v(x, u_x) + g_x^T(z - x) \le f(z)\ \forall z \in \mathbb{R}^{n_x}$$
$$\Rightarrow\ f(x) + g_x^T(z - x) \le f(z) + \epsilon_G^p\ \forall z \in \mathbb{R}^{n_x} \Rightarrow\ g_x \in \partial^{[\epsilon_G^p]} f(x).$$

This shows $\tilde{\partial}_a f(x) \subset \partial^{[\epsilon_G^p]} f(x)$. Thus, the approximate optimality condition $0 \in \tilde{\partial}_a f(x)$ already implies $f(x) \le \min_x f(x) + \epsilon_G^p$, i.e., that the point x is ϵ_G^p-optimal.

We further note that in practice, we may not have access to ϵ_G^p, but we have the *required* precision of the corresponding function evaluation ϵ_G^K, determined by (14) and specified in (23)-(25) for the three different termination scenarios, which is an upper bound for the exact error ϵ_G^p.

Subgradients w.r.t. Integer Variables We now provide formulas for subgradients with respect to integer variables. Having generated $\tilde{\xi}_K$ via the adaptive bundle method as described above, we have $\tilde{\xi}_K = \sum_{l=0}^{k-1} \lambda^k s_G^l$ with k the bundle iteration index at which we stopped and $s_G^l \in \partial_x V(x^l, y_K, u_{x^l})$. Now, we compute in every bundle iteration a subgradient $s_{G,y}^l$ such that $(s_G^l, s_{G,y}^l)$ fulfills the subgradient inequality (7) for $V(\cdot, \cdot, u_{x^l})$ at (x^l, y_K) and choose

$$\tilde{\eta}_K = \sum_{l=0}^{k-1} \lambda_l^k s_{G,y}^l,$$

with weights λ_l^k from (19) as an integer subgradient. Analogously, we compute

$$\beta_K = \sum_{l=0}^{k-1} \lambda_l^k s_{C,y}^l.$$

These subgradients, β_K and $\tilde{\eta}_K$, fulfill the subgradient inequality (7) for $C(\cdot, \cdot)$ at (x_K, y_K) and for $G(\cdot, \cdot)$ at (x_K, y_K) up to the error ϵ_G^p in $\tilde{G}(x_K, y_K)$. For a theoretical justification, the proof of Theorem 4.1 can be extended in a straight-forward way. In particular, the required subgradient inequalities can be derived from an appropriate extension of (31) and (32).

Inexactness in Solution Output We finally comment on the inexactness in the solution output (23)-(25) and possibilities to enhance its quality. In particular, it is in an OA approach of interest to decide on the feasibility of the continuous subproblem $(NLP(y_K))$.

If we approximately solve the feasibility problem $(F(y_K))$ with the adaptive bundle method and obtain an output $\tilde{G}(x_K, y_K) > 0$, then we know that $(NLP(y_K))$ is indeed infeasible. Otherwise, if we solve $(F(y_K))$ with output $\tilde{G}(x_K, y_K) = 0$, or $(NLP(y_K))$ with any value for $\tilde{G}(x_K, y_K)$, we do not know if $(NLP(y_K))$ is feasible or not. For the latter case, we have the following three options. As preferred option, we can always simply increase ϵ_G^K by $\tilde{G}(x_K, y_K)$, then set, in (23)-(25), $\tilde{G}(x_K, y_K)$ to 0 and correctly label the problem as feasible with tolerance ϵ_G^K. If this leads to a larger feasibility error ϵ_G^K than desired, we can re-run the bundle method with increased penalty parameter and we can use a refinement strategy in the bundle method that has been proposed in [23, Corollary 5]. This strategy involves re-evaluations of function values at serious iterates and leads to an exact convergence result, i.e., to $0 \in \partial f(\bar{x})$ in Corollary 4.1. This option can however be rather expensive. We note that Assumption 3.2 ensures that, if $(NLP(y_K))$ is feasible, then this strategy of exactly solving the penalized problem $(NLP_\psi(y_K))$ leads to a feasible and optimal solution to $(NLP(y_K))$.

In contrast to inexactness in the feasibility with respect to $G(x_K, y_K)$, we require exact feasibility with respect to the constraint $x \in X$. To achieve this, we can rely on the options above and hence can assume that $x_K \in X$ for the solution output. We note that in our computational experiments, we never needed to employ these options to achieve feasibility with respect to X.

For $x_K \in X$, the subdifferential of the penalty term $P_X(x_K)$ is contained in the normal cone of X at x_K, which we use in the following and which we denote by

$$N(X, x_K) = \{z \in \mathbb{R}^{n_x} : z^T(x - x_K) \leq 0 \quad \forall x \in X\}.$$

As we assume $x_K \in X$, our results thus hold for $\partial P_X(x_K)$ replaced by $N(X, x_K)$.

5 Inexactness in Function Values and Subgradients

In the previous section, we have seen that inexact worst-case evaluations lead to the following situation: for every iterate (x_K, y_K), an ϵ_G^K-optimal solution to the adversarial problem with a known tolerance ϵ_G^K is available. Denoting the approximation of the function value $G(x_K, y_K)$ by $\tilde{G}(x_K, y_K)$, we thus have

$$\tilde{G}(x_K, y_K) = V(x_K, y_K, \tilde{u}) \geq \max_{u \in \mathcal{U}} V(x_K, y_K, u) - \epsilon_G^K.$$

Having access to the approximate worst case $\tilde{u}$, one has natural access to a subgradient $\xi \in \partial_x V(x_K, y_K, \tilde{u})$, which fulfills

$$\tilde{G}(x_K, y_K) + \xi^T(z - x_K) \leq G(z, y_K) \quad \forall z \in X.$$

Such an approximate subgradient then lies in the convex ϵ_G^p-subdifferential of G at x_K, where ϵ_G^p denotes the exact error in the evaluation of G. It will be shown later in this section that any such subgradient can be used to generate a cutting plane, which cuts off a point only if it is infeasible or if it does not improve the current best objective value. However, in order to guarantee that our OA algorithm converges in a finite number of iterations, it is also required that every integer assignment should be visited only once. For this, we need the following stronger assumption.

Assumption 5.1 At an arbitrary iteration K, the problems ($NLP(y_K)$) and ($F(y_K)$) can be solved with output $\big(x_K, C(x_K, y_K), \epsilon_G^K, \tilde{G}(x_K, y_K), \alpha_K, \tilde{\xi}_K, \zeta_K\big)$, such that $x_K \in X$,

$$\exists 0 \leq \epsilon_G^p \leq \epsilon_G^K \colon \tilde{G}(x_K, y_K) = G(x_K, y_K) - \epsilon_G^p, \tag{35a}$$

$$\alpha_K \in \partial_x C(x_K, y_K),\ \tilde{\xi}_K \in \partial_x^{[\epsilon_G^p]} G(x_K, y_K),\ \zeta_K \in N(X, x_K), \tag{35b}$$

$$\exists \psi, \psi_X \colon 0 = \alpha_K + \psi \tilde{\xi}_K + \psi_X \zeta_K. \tag{35c}$$

For the feasibility problem ($F(y_K)$), we simply set $C \equiv 0$ in (35b) and $\psi = 1$ in (35c). Further, if ($NLP(y_K)$) is feasible, then it can be solved with $\tilde{G}(x_K, y_K) = 0$.

In Sect. 4, we have shown that Assumption 5.1 is satisfied when applying the adaptive bundle method from [23] with the modifications outlined in Algorithm 2 to the subproblems ($NLP(y_K)$) and ($F(y_K)$). In this case, Assumption 5.1 does not need to be checked. Nevertheless, the OA method we develop also allows the integration of any other method, as long as it fulfills Assumption 5.1.

To generate the linearized constraints of the master problem, we also need suitable subgradients with respect to the integer variables. More precisely, when we have subgradients $\alpha_K \in \partial_x C(x_K, y_K)$, $\tilde{\xi}_K \in \partial_x^{[\epsilon_G^p]} G(x_K, y_K)$ at hand, we use subgradients β_K and $\tilde{\eta}_K$ with respect to the integer variables, such that

$$C(x_K, y_K) + \begin{pmatrix} \alpha_K \\ \beta_K \end{pmatrix}^T \begin{pmatrix} \bar{x} - x_K \\ \bar{y} - y_K \end{pmatrix} \le C(\bar{x}, \bar{y}) \qquad \forall (\bar{x}, \bar{y}) \in X \times Y \cap \mathbb{Z}^{n_y}, \tag{36}$$

$$G(x_K, y_K) + \begin{pmatrix} \tilde{\xi}_K \\ \tilde{\eta}_K \end{pmatrix}^T \begin{pmatrix} \bar{x} - x_K \\ \bar{y} - y_K \end{pmatrix} \le G(\bar{x}, \bar{y}) + \epsilon_G^p \qquad \forall (\bar{x}, \bar{y}) \in X \times Y \cap \mathbb{Z}^{n_y}. \tag{37}$$

We have detailed in Sect. 4, how such subgradients β_K and $\tilde{\eta}_K$ can also be generated in the course of the adaptive bundle method.

As we evaluate G only inexactly, we need an according notion of inexactness for the sets S^K and T^K, which are in an exact setting subsets of S and T, respectively (see Sect. 3). At an integer point that we label as infeasible, we underestimate the minimum value of G (see Assumption 5.1). These integer points are hence indeed infeasible and we collect them in a set $\tilde{S}^K \subseteq S$ with

$$\tilde{S}^K := \{y_L \mid L \le K,\ \text{Assumption 5.1 is fulfilled for } (F(y_K)) \text{ with } \tilde{G}(x_K, y_K) > 0\}.$$

In contrast, integer points we label as feasible may in reality be infeasible. We collect them in the following set, which is an inexact version of visited feasible integer assignments.

$$\tilde{T}^K := \{y_J \mid J \le K,\ \text{Assumption 5.1 is fulfilled for } (NLP(y_K)) \text{ with } \tilde{G}(x_K, y_K) = 0\}.$$

If $\tilde{G}(x_K, y_K) = 0$ for the obtained solution to $(NLP(y_K))$, then we set $y_K \in \tilde{T}^K$. In this case, it holds that $\tilde{G}(x_K, y_K) = 0$ with $\tilde{G}(x_K, y_K) \ge G(x_K, y_K) - \epsilon_G^K$ and we say that (y_K, x_K) is ϵ_G^K-feasible.

In an iteration K, we obtain as inexact master problem (i.e., an inexact version of (MP^K)), with approximate function values $\tilde{G}(\cdot, \cdot)$ and subgradients $\tilde{\xi}$, $\tilde{\eta}$,

$$\begin{aligned} \min_{x,y,\theta} \quad & \theta && \qquad (\widetilde{MP}^K) \\ \text{s.t.} \quad & \theta \le \Theta^K - \epsilon_{oa} \\ & C(x_J, y_J) + (\alpha_J^T, \beta_J^T) \begin{pmatrix} x - x_J \\ y - y_J \end{pmatrix} \le \theta && \forall y_J \in \tilde{T}^K \\ & \tilde{G}(x_J, y_J) + (\tilde{\xi}_J^T, \tilde{\eta}_J^T) \begin{pmatrix} x - x_J \\ y - y_J \end{pmatrix} \le 0 && \forall y_J \in \tilde{T}^K \\ & \tilde{G}(x_L, y_L) + (\tilde{\xi}_L^T, \tilde{\eta}_L^T) \begin{pmatrix} x - x_L \\ y - y_L \end{pmatrix} \le 0 && \forall y_L \in \tilde{S}^K \\ & x \in X,\ y \in Y \cap \mathbb{Z}^{n_y}. \end{aligned}$$

5.1 Algorithm

With these algorithmic concepts at hand, we now state the OA method for mixed-integer nonlinear robust optimization with the notion of inexactness formalized in Assumption 5.1. We use the adaptive bundle method from Sect. 4 for the solution of the continuous subproblems. The corresponding steps in the outer approximation algorithm outlined below are marked by "bundle." Moreover, if the bundle method is used, the computations in line 6 and line 12 can be realized as detailed in the paragraph on *subgradients w.r.t. integer variables* and the condition in line 16 can be realized by the strategies in the paragraph on *inexactness in the solution output*, both at the end of Sect. 4. However, we note that the OA method does not rely on a specific method for these steps and is thus not restricted to the bundle method suggested here. For the subproblems, any method that leads to solutions fulfilling Assumption 5.1 can be used instead.

Algorithm 1 OA for mixed-integer nonlinear robust optimization with inexactness

```
1:  Fix parameter $\epsilon_{oa} > 0$.
2:  Choose initial values: $y^1 \in Y$, $\tilde{S}^0 = \tilde{T}^0 = \emptyset$, $\Theta^0 = \infty$, $K = 1$.
3:  while $\widetilde{MP}^{K-1}$ feasible do
4:      Solve ($NLP(y_K)$), fulfilling Assumption 5.1, with output
          denoted by $\big(x_K, C(x_K, y_K), \epsilon_G^K, \tilde{G}(x_K, y_K), \alpha_K, \tilde{\xi}_K, \zeta_K\big)$.          ▷ bundle
5:      if $\tilde{G}(x_K, y_K) = 0$ then
6:          Compute $\beta_K, \tilde{\eta}_K$ fulfilling (36)-(37).
7:          $\tilde{T}^K \leftarrow \tilde{T}^{K-1} \cup \{y_K\}$, $\tilde{S}^K \leftarrow \tilde{S}^{K-1}$.
8:          $\Theta^K \leftarrow \min\{\Theta^{K-1}, C(x_K, y_K)\}$.
9:      else
10:         Solve ($F(y_K)$) fulfilling Assumption 5.1, with output
            denoted by $\big(x_K, C(x_K, y_K), \epsilon_G^K, \tilde{G}(x_K, y_K), \alpha_K, \tilde{\xi}_K, \zeta_K\big)$.          ▷ bundle
11:         if $\tilde{G}(x_K, y_K) > 0$ then
12:             Compute $\tilde{\eta}_K$ fulfilling (37).
13:             $\tilde{T}^K \leftarrow \tilde{T}^{K-1}$, $\tilde{S}^K \leftarrow \tilde{S}^{K-1} \cup \{y_K\}$.
14:             $\Theta^K \leftarrow \Theta^{K-1}$.
15:         else
16:             Go to Step 4 and enforce $\tilde{G}(x_K, y_K) = 0$.
17:         end if
18:     end if
19:     Solve ($\widetilde{MP}^K$) and denote the solution's integer part by $y_{K+1}$.
20:     Increase $K$ by 1.
21: end while
22: Set $K^* \in \{J \mid J \leq K, \Theta^K = C(x_J, y_J)\}$.
23: Return $(x_{K^*}, y_{K^*})$.
```

5.2 Inexact Cutting Planes

We now prove correctness and finite convergence of the proposed OA method. Therefore, we closer examine the cutting planes generated by the two types of continuous problems.

Cutting Planes Generated by the Continuous Subproblem We show that the linear constraints with respect to $\tilde{T}^K$ in ($\widetilde{MP}^K$) are valid and cut off the current integer solution. In an iteration K, let x_K be an approximate solution to the continuous subproblem ($NLP(y_K)$) that fulfills Assumption 5.1. We consider the following inexact cutting planes:

$$\begin{aligned} \tilde{G}(x_K, y_K) + \tilde{\xi}_K^T(x - x_K) + \tilde{\eta}_K^T(y - y_K) &\leq 0 &\quad (38a)\\ C(x_K, y_K) + \alpha_K^T(x - x_K) + \beta_K^T(y - y_K) &\leq \theta &\quad (38b)\\ \theta &\leq \Theta^K - \epsilon_{oa} &\quad (38c)\\ \Theta^K &= \min\{\Theta_{K-1}, C(x_K, y_K)\}. &\quad (38d) \end{aligned}$$

The constraints are valid in the sense that they cut off infeasible solutions, and feasible solutions only if they do not improve the current best objective value by more than ϵ_{oa}. We prove this and further show that (38) cuts off the current integer solution:

Lemma 5.1 *If $(\bar{x}, \bar{y}) \in \mathbb{R}^{n_x+n_y}$ is feasible for (P) and infeasible for (38), then $C(\bar{x}, \bar{y}) > \Theta^K - \epsilon_{oa}$. Furthermore, for any $\bar{x} \in X$, $(\bar{x}, y_K)$ is infeasible for (38).*

Proof We prove the first claim: Let $(\bar{x}, \bar{y}) \in \mathbb{R}^{n_x+n_y}$ be feasible for (P) and infeasible for (38). We show feasibility of $(\bar{x}, \bar{y})$ for (38a). There exists an $\epsilon_G^p > 0$ with $\tilde{G}(x_K, y_K) = G(x_K, y_K) - \epsilon_G^p$ and with (37), we have

$$\tilde{G}(x_K, y_K) + \tilde{\xi}_K^T(\bar{x} - x_K) + \tilde{\eta}_K^T(\bar{y} - y_K) \leq G(\bar{x}, \bar{y}).$$

As $G(\bar{x}, \bar{y}) \leq 0$ due to feasibility of $(\bar{x}, \bar{y})$ for (P), it follows that $(\bar{x}, \bar{y})$ fulfills (38a). It hence violates (38b)–(38c). This implies that $C(\bar{x}, \bar{y}) > \Theta^K - \epsilon_{oa}$.

We prove the second claim by contradiction: We assume that there exists an $\bar{x} \in X$ such that $(\bar{x}, y_K)$ is feasible for (38). Then,

$$\begin{aligned} \tilde{G}(x_K, y_K) + \tilde{\xi}_K^T(\bar{x} - x_K) &\leq 0 &\quad (39a)\\ C(x_K, y_K) + \alpha_K^T(\bar{x} - x_K) &\leq \theta &\quad (39b)\\ \theta &\leq \Theta^K - \epsilon_{oa} &\quad (39c)\\ \Theta^K &= \min\{\Theta^{K-1}, C(x_K, y_K)\}. &\quad (39d) \end{aligned}$$

As $\tilde{G}(x_K, y_K) = 0$, we have $\tilde{\xi}_K^T(\bar{x} - x_K) \leq 0$. It follows from Assumption 5.1, (35b), (35c), that $\alpha_K^T(\bar{x} - x_K) \geq 0$. Hence, due to (39b), $C(x_K, y_K) \leq \theta$, which contradicts (39c),(39d). □

Cutting Planes Generated by the Feasibility Problem Let x_K be an approximate solution to the feasibility problem ($F(y_K)$) fulfilling Assumption 5.1. We consider the cutting plane

$$\tilde{G}(x_K, y_K) + \tilde{\xi}_K^T(x - x_K) + \tilde{\eta}_K^T(y - y_K) \leq 0. \tag{40}$$

We note that we have no inexactness with respect to the claim that the nonlinear subproblem is infeasible: if the underestimated optimal value of the feasibility problem indicates infeasibility, the subproblem is indeed infeasible. We now prove that the cutting plane (40) cuts off the current integer solution without cutting off any feasible solution.

Lemma 5.2 *If $(\bar{x}, \bar{y}) \in \mathbb{R}^{n_x+n_y}$ is feasible for (P), then it is feasible for (40). Furthermore, for any $\bar{x} \in X$, $(\bar{x}, y_K)$ is infeasible for (40).*

Proof From feasibility of $(\bar{x}, \bar{y})$ for (P), it follows that $G(\bar{x}, \bar{y}) \leq 0$. As there is an $\epsilon_G^p > 0$ such that $\tilde{G}(x_K, y_K) = G(x_K, y_K) - \epsilon_G^p$ and by (37), it holds that

$$\tilde{G}(x_K, y_K) + \tilde{\xi}_K^T(\bar{x} - x_K) + \tilde{\eta}_K^T(\bar{y} - y_K) \leq G(\bar{x}, \bar{y}),$$

so that $(\bar{x}, \bar{y})$ fulfills the constraint (40).

We prove the second claim by contradiction: We assume that there exists an $\bar{x} \in X$ with

$$\tilde{G}(x_K, y_K) + \tilde{\xi}_K^T(\bar{x} - x_K) + \tilde{\eta}_K^T(y_K - y_K) \leq 0.$$

From Assumption 5.1, (35b), (35c), it follows that $\tilde{\xi}_K^T(\bar{x} - x_K) \geq 0$. As $\tilde{\eta}_K^T(y_K - y_K) = 0$, this is is a contradiction to $\tilde{G}(x_K, y_K) > 0$. □

5.3 Finite Convergence of the Outer Approximation Method

We now combine the results from the preceding section to show that Algorithm 1 terminates after finitely many steps and that a solution (x_{K^*}, y_{K^*}) found by the algorithm is $\epsilon_G^{K^*}$-feasible and ϵ_{oa}-optimal. The proof uses similar arguments as, e.g., [35] and [14].

Theorem 5.1 *If (P) is feasible, then Algorithm 1 terminates after finitely many iterations with a solution (x_{K^*}, y_{K^*}) that is $\epsilon_G^{K^*}$-feasible and ϵ_{oa}-optimal for (P). If (P) is infeasible, Algorithm 1 either outputs a solution (x_{K^*}, y_{K^*}) that is $\epsilon_G^{K^*}$-feasible and ϵ_{oa}-optimal for (P) or detects infeasibility, after finitely many iterations.*

Proof It follows from Lemma 5.1 and Lemma 5.2 that, if ($\widetilde{MP}^K$) is infeasible, then Algorithm 1 either correctly detects infeasibility of (P) or outputs an ϵ_{oa}-optimal solution. Further, any candidate solution (x_K, y_K) with feasibility tolerance ϵ_G^K is ϵ_G^K-feasible. Finite convergence follows from the fact that, by Lemma 5.1 and Lemma 5.2, each integer point in Y is visited only once. □

The OA method thus is applicable to mixed-integer nonlinear robust optimization with the notion of inexactness specified in Assumption 5.1. We use inexact worst-case evaluations with precision ϵ_G^K and therefore may accept solutions that are only ϵ_G^K-feasible. Consequently, we cannot achieve a better result than the approximate feasibility in Theorem 5.1.

In an iteration K of the OA method, the feasibility tolerance ϵ_G^K is not specified before the subproblem is solved. Whether or not this can be specified in advance depends on the method used for the subproblem. In any case, if the subproblem's solution happens to be the final solution of the OA method, the algorithm outputs this solution, which is ϵ_G^K-feasible.

6 Numerical Results for the Gas Transport Problem

We implemented the OA approach with the adaptive bundle method in MATLAB and Python with Gurobi 8.1 [18]. We approximated the adversarial maximization problem via piecewise linear relaxation, for which we used the delta method [15]. This was implemented in Python with Gurobi 8.1. The experiments were done on a machine with an Intel Core i7-8550U (4 cores) and 16GB of RAM.

We used instances from the GasLib library, which contains realistic gas network instances [31]. We slightly modified the instances such that they fulfill our assumptions. The modified instances are publicly available as online supplement to this paper. We evaluated our methods for networks with up to 103 nodes. The two smallest GasLib instances are defined on networks with 11 and 24 nodes, respectively, and the robust gas transport problem is solved by our method within only a few seconds. The computational results get more interesting for the larger GasLib instances with 40 and with 103 nodes, on which we focus here. These networks are of the sizes of real networks.

In the adaptive bundle method, we used the stopping criterion $g_k^* \leq 10^{-7}$ along with heuristic stopping criteria from [23, Section 4.3]. Nevertheless, as it can be seen from the following tables, the required precision for the aggregate subgradient is met in almost all cases. In the tables, we list the computational times spent within the bundle method as 'runtime bundle.' The main part of the OA method's running time is spent for the subproblems. For the OA iterations in which no approximate feasible solution could be found, we list the accumulated running times for the runs of the bundle method working on ($F(y_K)$) and ($NLP(y_K)$). We did not need to resolve ($NLP(y_K)$) (line 16 in Algorithm 1) in our experiments. The bundle method's running time is mainly spent for solving the adversarial problems up to the required precisions. In order to reach a solution within reasonable running time, we bounded the required precision by 10^{-3} for the 40-node instance, i.e., $\epsilon_f^k \geq 10^{-3}$, and by 10^{-1} for the 103-node instance, i.e., $\epsilon_f^k \geq 10^{-1}$. As it can be seen from the tables, where $\bar{\epsilon}_G^p$ denotes an upper bound on the exact a posteriori error, this did not prevent this error from becoming small. As cost function w in (P_{gas}), we used compressor costs, determined by the achieved difference of squared pressures. We internally scaled these costs by a factor of 10^{-2}. For the use of valves, we did not charge any costs.

In Tables 1 and 2, results are presented for a slightly modified version of GasLib-40 with 40 nodes, 5 compressors and 2 valves. In detail, we removed a compressor on a cycle in the original instance in order to fulfill the assumption of Lemma 2.1. We replaced it by a valve and added another valve on a cycle. As a benchmark result, we first applied our method to the nominal problem, for which we obtained—within a few seconds—optimal compressor costs of 1148. Then, we solved the robust problems with different uncertainty sets, namely once with 5% and once with 10% deviation from the nominal value : $[0.975{\cdot}d,\ 1.025{\cdot}d]$, $[\lambda,\ 1.05{\cdot}\lambda]$ and $[0.95{\cdot}d,\ 1.05{\cdot}d]$, $[\lambda,\ 1.1{\cdot}\lambda]$. These sets yield a robust protection against a reasonable amount of parameter deviation. The corresponding results are presented in Table 1 and 2, respectively. From these results, we compute the price of robustness, which is the relative increase of compressor costs caused by the robust treatment of uncertainties. For the first uncertainty set, it amounts to 45% and for the second to 93%. The larger uncertainty set thus leads to almost twice the nominal compressor costs.

In Table 3, we present results for a modified version of GasLib-135 that has 103 nodes, 21 compressors, which are not on cycles, and 3 valves. As uncertainty sets for the demand and for pressure loss coefficients d and λ, we used the set of balanced demands in $[0.975 \cdot d, 1.025 \cdot d]$ and the set $[\lambda, 1.05 \cdot \lambda]$, respectively. Typically, the running time for the adversarial problems, and thus for the whole method, increases when we enlarge the uncertainty set. In order to keep the adversarial problems solvable within a reasonable amount of time, we restricted ourselves to an uncertainty set of 5% deviation for this network.

For the nominal problem, we encountered—within less than one minute—an optimal objective value of 704.2, so that the price of robustness amounted to 30% for the chosen uncertainty sets, which is in the same order of magnitude as in the case of the smaller instance.

Table 1 GasLib-40 with uncertain parameters in $[0.975 \cdot d, 1.025 \cdot d]$ and $[\lambda, 1.05 \cdot \lambda]$ and 7 binary decision variables

	Θ	Penalty	$\bar{\epsilon}_G^p$	Runtime bundle	$\|g^*\|$	Active compressors	Open valves
0	Inf	4821.1	2.41e+01	–	–	2, 3, 4	1, 2
1	Inf	4821.1	8.48e–04	42.56	0	1, 2, 3, 4, 5	1, 2
2	2064.0	0	0	22.79	6.34e–08	5	
3	1664.0	0	0	1.36	1.81e–08		

Table 2 GasLib-40 with uncertain parameters in $[0.95 \cdot d, 1.05 \cdot d]$ and $[\lambda, 1.1 \cdot \lambda]$ and 7 binary decision variables

	Θ	Penalty	$\bar{\epsilon}_G^p$	Runtime bundle	$\|g^*\|$	Active compressors	Open valves
0	Inf	6450.4	1.03e+02	–	–	2, 3, 4	1, 2
1	Inf	6450.4	9.40e–04	343.68	0	1, 2, 3, 4, 5	1, 2
2	2615.7	0	0	7.81	1.35e–08	5	
3	2615.7	327.4	5.02e–03	537.29	1.60e–07	5	2
4	2215.4	0	0	4.45	2.53e–08		

Table 3 103-nodes instance with uncertain parameters in $[0.975 \cdot d, 1.025 \cdot d]$ and $[\lambda, 1.05 \cdot \lambda]$ and 24 binary decision variables

	Θ	Penalty	$\bar{\epsilon}_G^p$	Runtime bundle	$\|g^*\|$	Active compressors	Open valves
0	Inf	635.8	8.40e–07	–	–	1–21	1, 2
1	Inf	128.7	3.94e–07	90.95	1.25e–07		
2	Inf	21404.4	8.81e+01	0.00	0	2, 3, 6–8, 11–15, 17, 20, 21	1, 3
3	1917.4	0	4.90e–04	63.68	9.77e–08	4	1, 3
4	1917.4	608.0	1.36e–09	22.87	9.82e–12	4, 14	1, 3
5	917.4	0	0	10.50	1.56e–14	6	1, 3
6	917.4	0	0	12.52	7.02e–08	2, 14	1, 3
7	917.4	0	0	18.02	7.38e–08	14	1, 3
8	917.4	0	0	7.64	2.07e–09	6, 14	1, 3
9	917.4	0	0	8.35	1.33e–07	4, 6	1, 3
10	917.4	0	0	9.02	9.52e–08	2, 6	1, 3
11	917.4	0	6.87e–05	4.16	9.07e–11	2, 4, 14	1, 3
12	917.4	0	0	11.11	2.65e–08	2, 4, 6	1, 3
13	917.4	0	0	12.08	2.33e–08	4, 6, 14	1, 3
14	917.4	0	0	1.64	1.26e–10	2, 4, 6, 14	1, 3
15	917.4	0	0	10.62	8.99e–08	2, 6, 14	1, 3
16	917.4	0	5.57e–05	5.09	1.07e–08		

We care to mention that the considered robust setting that allows for discrete-continuous decisions has not been solved in the literature so far. The case of only continuous decisions is roughly comparable to one iteration within our OA method. This simpler case has been treated by a decomposition approach specifically designed for robust gas networks in [3]. There, the instance GasLib-40 could be solved within a few seconds or a few minutes—depending on an error in the relaxation of non-convex constraints. We have observed that the discrete-continuous robust gas transport problem on instances of size 40-nodes and 103-nodes are solvable within a few minutes by our method where we obtained—up to a small tolerance in the last case—robust feasible solutions. Thus, although our method is a general approach for mixed-integer robust optimization that is applicable in wider contexts as [3], it solves a challenging and more complex robust optimization task within a similar order of running time as could be obtained in [3].

7 Conclusion

We proposed an outer approximation approach for nonlinear robust optimization with mixed-integer decisions and inexact worst-case evaluations. In the core of this, an adaptive bundle method was used to solve the continuous subproblems. In general, the method can be applied to robust problems, in which uncertain parameters enter the problem in a non-concave way and in which only approximate worst cases are computationally accessible. This setting is extremely challenging, and no general solution approach exists. According to our numerical results, it performs very well on an example application in robust gas transport and can solve relevant real-world problems.

There are possibilities to improve the performance of the method. As proposed in [9], the bundle method in an iteration of the OA method could be initialized by the use of cutting planes from earlier runs. One could thereby think of an appropriate downshift mechanism of recycled cutting planes, as used in bundle methods to recycle cutting planes from former outer loops. Another idea would be to exchange cutting planes between the bundle method's cutting plane model and the master problem in the OA approach. Also, to accelerate the master problems' solution, one could employ a so-called single-tree approach, as proposed in [32], where the branch-and-bound tree for the MIP's solution is not re-built in every iteration. Further, one could employ regularization strategies in order to avoid large step-sizes between the master problems' solutions [1, 9, 22, 30].

Apart from accelerating the proposed approach, there are possibilities of extending the scope of applicability of our method. Probably the most interesting case would be a relaxation of the convexity assumption. One possible avenue here would be to resort to concepts of pseudo- and quasi-convexity, as has been done for the related extended cutting plane methods [11] and extended supporting hyperplane methods [36]. As pointed out in [13], a suitable framework for the OA method could be the one by [19], which requires only quasi-convexity. Such an integration could be a challenging subject of future research and requires a substantial extension of our results.

Acknowledgements The authors thank the DFG for their support within Project B06 in CRC TRR 154, as well as within Project-ID 416229255 - SFB 1411.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability All data generated or analyzed during this study are included in this published article and its supplementary information files.

Appendix A. Algorithm: The Adaptive Bundle Method

Algorithm 2 Adaptive bundle method with subgradient generation

```
1: Fix parameters: $0 < \gamma < \Gamma < 1, \gamma < \tilde{\gamma} < 1, 0 < q < T < \infty, 0 < \nu < T - q, \epsilon'' > 0, \epsilon_f^{x_1} > 0, \epsilon^* \geq 0$.
2: Choose initial values: Starting point $x_1$, $Q_1 = Q_1^T$ s.t. $-q\mathcal{I} \preceq Q_1 \preceq q\mathcal{I}$, $\tau_1^\#$ s.t. $Q_1 + \tau_1^\# \mathcal{I} \succ 0$. Set $j = 1$.
3: while True do                                   ▷ removed stopping test "dist(0, $\tilde{\partial}_a f(x_j)) \leq \tilde{\epsilon}$"
4:     Initialize inner loop with serious iterate $x_j$: $\tau_1 = \tau_j^\#$. Set $k = 1$.
5:     Solve the program                                                    ▷ Trial step generation
           $$\min_{x^k \in \mathbb{R}^n} \Phi_k(x^k, x_j) + \frac{\tau_k}{2} \|x^k - x_j\|^2.$$
6:     Compute $f_a(x^k)$ with $\epsilon_f^k = \epsilon'' \|x_j - x^k\|^2$.      ▷ Adaptive function value approximation
7:     Set $g_k^* = (Q_j + \tau_k \mathcal{I})(x_j - x^k)$.
8:     if $\|g_k^*\| \leq \epsilon^*$ then
9:         Stop.                                                            ▷ Stopping test
10:    end if
11:    Compute $\lambda_l^k$ s.t. $\sum_{l<k} \lambda_l^k g_l = g_k^* \wedge \sum_{l<k} \lambda_l^k = 1 \wedge \lambda_l^k > 0 \Rightarrow m_l(x^k, x_j) = \phi_k(x^k, x_j)$.
12:    Set $\alpha_k^j = \sum_{l<k} \lambda_l^k s_C^l$, $\tilde{\xi}_k^j = \sum_{l<k} \lambda_l^k s_G^l$.
                                                                            ▷ Subgradient generation
13:    if $\rho_k = \frac{f_a(x_j) - f_a(x^k)}{f_a(x_j) - \Phi_k(x^k, x_j)} \geq \gamma$ then             ▷ Acceptance test
14:        $x_{j+1} \leftarrow x^k$.
15:        $\tau_{j+1}^\# \leftarrow \begin{cases} \tau_k & \rho_k < \Gamma, \\ 1/2\tau_k & \rho_k \geq \Gamma. \end{cases}$      ▷ Update $\tau^\#$
16:        Set $Q_{j+1}$ s.t. $Q_{j+1} = Q_{j+1}^T, -q\mathcal{I} \preceq Q_{j+1} \preceq q\mathcal{I}$.     ▷ Update $Q_j$
17:        If $Q_{j+1} + \tau_{j+1}^\# \mathcal{I} \not\succ \nu\mathcal{I}$: Increase $\tau_{j+1}^\#$ s.t. $Q_{j+1} + \tau_{j+1}^\# \mathcal{I} \succ \nu\mathcal{I}$.
18:        $\tau_{j+1}^\# \leftarrow \min\{\tau_{j+1}^\#, T\}$.
19:    else
20:        Compute $s_C^k$, $s_G^k$ and $s_X^k$
               $$s_C^k \in \partial_x C(x^k), s_G^k \in \tilde{\partial}_a G(x^k), s_X^k \in \partial P_X(x^k).$$
21:        Set $g_k = s_C^k + \psi s_G^k + \psi_X s_X^k$.
22:        Generate a cutting plane $m_k(\cdot, x_j)$ with $g_k \in \tilde{\partial}_a f(x^k)$.
23:        if $\tilde{\rho}_k = \frac{f_a(x_j) - M_k(x^k, x_j)}{f_a(x_j) - \Phi_k(x^k, x_j)} \geq \tilde{\gamma}$ then        ▷ Update $\tau_k$
24:            $\tau_{k+1} \leftarrow 2\tau_k$.
25:        end if
26:        Build new working model $\phi_{k+1}$: add $m_k(\cdot, x_j)$ to $\phi_k$.        ▷ Update $\phi_k$
27:        Increase $k$ and go to Step 5: Trial step generation.
28:    end if
29:    Increase $j$.
30: end while
```

References

1. Ackooij, W., van Frangioni, A., de Oliveira, W.: Inexact stabilized Benders' decomposition approaches with application to chance-constrained problems with finite support. Comput. Optim. Appl. **65**(3), 637–669 (2016). https://doi.org/10.1007/s10589-016-9851-z
2. Aßmann, D.: Exact Methods for Two-Stage Robust Optimization with Applications in Gas Networks. PhD thesis. FAU University Press (2019). https://doi.org/10.25593/978-3-96147-234-5
3. Aßmann, D., Liers, F., Stingl, M.: Decomposable robust two-stage optimization: an application to gas network operations under uncertainty. Networks **74**(1), 40–61 (2019). https://doi.org/10.1002/net.21871
4. Aßmann, D., Liers, F., Stingl, M., Vera, J.C.: Deciding robust feasibility and infeasibility using a set containment approach: an application to stationary passive gas network operations. SIAM J. Optim. **28**(3), 2489–2517 (2018). https://doi.org/10.1137/17M112470X
5. Bagirov, A., Karmitsa, N., Mäkelä, M.M.: Introduction to Nonsmooth Optimization, Theory, Practice and Software. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08114-4
6. Ben-Tal, A., den Hertog, D., Vial, J.-P.: Deriving robust counterparts of nonlinear uncertain inequalities. Math. Program. Ser. A **149**(1), 265–299 (2015). https://doi.org/10.1007/s10107-014-0750-8
7. Bixby, R.E., Fenelon, M., Gu, Z., Rothberg, E., Wunderling, R.: Mixed-integer programming: a progress report. M. Grötschel (Eds.) The Sharpest Cut, pp. 309–325. MPS/SIAM Ser. Optim. SIAM, Philadelphia, PA (2004)
8. Clarke, F. H. (1990) Optimization and Nonsmooth Analysis. Classics in Applied Mathematics,2nd ed., vol. 5. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. https://doi.org/10.1137/1.9781611971309
9. Delfino, A., de Oliveira, W.: Outer-approximation algorithms for nonsmooth convex MINLP problems. Optimization **67**(6), 797–819 (2018). https://doi.org/10.1080/02331934.2018.1434173
10. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. **36**(3), 307–339 (1986). https://doi.org/10.1007/BF02592064
11. Eronen, V.-P., Mäkelä, M.M., Westerlund, T.: On the generalization of ECP and OA methods to nonsmooth convex MINLP problems. Optimization **63**(7), 1057–1073 (2014). https://doi.org/10.1080/02331934.2012.712118
12. Eronen, V.-P., Mäkelä, M.M., Westerlund, T.: Extended cutting plane method for a class of nonsmooth nonconvex MINLP problems. Optimization **64**(3), 641–661 (2015). https://doi.org/10.1080/02331934.2013.796473
13. Eronen, V.-P., Westerlund, T., Mäkelä, M. M.: On Mixed integer nonsmooth optimization. In: A. M. Bagirov, M. Gaudioso, N. Karmitsa, M. M. Mäkelä, and S. Taheri (Eds.) Numerical Nonsmooth Optimization: State of the Art Algorithms, pp. 549–578. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-34910-3_16
14. Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation. Math. Program. Ser. A **66**(3), 327–349 (1994). https://doi.org/10.1007/BF01581153
15. Geißler, B., Martin, A., Morsi, A., and Schewe, L.: Using piecewise linear functions for solving MINLPs. In: J. Lee, S. Leyffer (Eds.) Mixed Integer Nonlinear Programming, vol. 154, pp. 87–314. IMA Vol. Math. Appl. Springer, New York (2012). https://doi.org/10.1007/978-1-4614-1927-3_10
16. Gotzes, C., Heitsch, H., Henrion, R., Schultz, R.: On the quantification of nomination feasibility in stationary gas networks with random load. Math. Methods Oper. Res. **84**(2), 427–457 (2016). https://doi.org/10.1007/s00186-016-0564-y
17. Grossmann, I.E.: MINLP: Outer Approximation Algorithm. In: C. A. Floudas and P. M. Pardalos (Eds.) Encyclopedia of Optimization, pp. 2179–2183. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-74759-0_385
18. Gurobi Optimization, L.: Gurobi Optimizer Reference Manual (2019)
19. Hamzeei, M., Luedtke, J.: Linearization-based algorithms for mixed-integer nonlinear programs with convex continuous relaxation. J. Glob. Optim. **59**(2), 343–365 (2014). https://doi.org/10.1007/s10898-014-0172-4
20. Hiriart-Urruty, J.-B., Lemaréchal, C.: Convex Analysis and Minimization Al- gorithms II., Vol. 306. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], pp. xviii+346. Springer, Berlin (1993). https://doi.org/10.1007/978-3-662-06409-2

21. Koch, T., Hiller, B., Pfetsch, M.E., Schewe, L. (eds.).: Evaluating Gas Network Capacities. Vol. 21. MOS-SIAM Series on Optimization, pp. xv+364. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. (2015). https://doi.org/10.1137/1.9781611973693
22. Kronqvist, J., Bernal, D.E., Grossmann, I.E.: Using regularization and second order information in outer approximation for convex MINLP. Math. Program. Ser. A **180**(1), 285–310 (2020). https://doi.org/10.1007/s10107-018-1356-3
23. Kuchlbauer, M., Liers, F., and Stingl, M. (2022) Adaptive Bundle Methods for Nonlinear Robust Optimization. INFORMS J. Comput. **34**(4), 2106–2124 (2022). https://doi.org/10.1287/ijoc.2021.1122Published online
24. Lasserre, J.B.: Min–max and robust polynomial optimization. J. Glob. Optim. **51**(1), 1–10 (2011). https://doi.org/10.1007/s10898-010-9628-3
25. Lasserre, J.B.: Robust global optimization with polynomials. Math. Program. Ser. B **107**(1), 275–293 (2006). https://doi.org/10.1007/s10107-005-0687-z
26. Leyffer, S., Menickelly, M., Munson, T., Vanaret, C., Wild, S.M.: A survey of nonlinear robust optimization. INFOR Inf. Syst. Oper. Res. **58**(2), 342–373 (2020). https://doi.org/10.1080/03155986.2020.1730676
27. Li, M., Vicente, L.N.: Inexact solution of NLP subproblems in MINLP. J. Global Optim. **55**(4), 877–899 (2013). https://doi.org/10.1007/s10898-012-0010-5
28. Noll, D.: Bundle method for non-convex minimization with inexact subgradients and function values. In: D.H. Bailey, H.H. Bauschke, P. Borwein, F. Garvan, M. Théra, J.D. Vanderwerff, H. Wolkowicz Computational and Analytical Mathematics, vol. 50, Springer Proc. Math. Stat., pp. 555–592. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-7621-4_26
29. de Oliveira, W., Sagastizábal, C.: Level bundle methods for oracles with ondemand accuracy. Optim. Methods Softw. **29**(6), 1180–1209 (2014). https://doi.org/10.1080/10556788.2013.871282
30. de Oliveira, W.: Regularized optimization methods for convex MINLP problems. TOP **24**(3), 665–692 (2016). https://doi.org/10.1007/s11750-016-0413-4
31. Pfetsch, M.E., Fügenschuh, A., Geißler, B., Geißler, N., Gollmer, R., Hiller, B., Humpola, J., Koch, T., Lehmann, T., Martin, A., Morsi, A., Rövekamp, J., Schewe, L., Schmidt, M., Schultz, R., Schwarz, R., Schweiger, J., Stangl, C., Steinbach, M.C., Vigerske, S., Willert, B.M.: Validation of nominations in gas network optimization: models, methods, and solutions. Optim. Methods Softw. **30**(1), 15–53 (2015). https://doi.org/10.1080/10556788.2014.888426
32. Quesada, I., Grossmann, I.E.: An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. Comput. Chem. Eng. **16**(10–11), 937–947 (1992). https://doi.org/10.1016/0098-1354(92)80028-8
33. Ruszczyński, A.: Nonlinear optimization. Princeton University Press, Princeton, NJ, pp. xiv+448 (2006)
34. Wei, Z., Ali, M.M.: Convex mixed integer nonlinear programming problems and an outer approximation algorithm. J. Glob. Optim. **63**(2), 213–227 (2015). https://doi.org/10.1007/s10898-015-0284-5
35. Wei, Z., Ali, M.M., Xu, L., Zeng, B., Yao, J.-C.: On solving nonsmooth mixedinteger nonlinear programming problems by outer approximation and generalized Benders decomposition. J. Optim. Theory Appl. **181**(3), 840–863 (2019). https://doi.org/10.1007/s10957-019-01499-7
36. Westerlund, T., Eronen, V.-P., Mäkelä, M.M.: On solving generalized convex MINLP problems using supporting hyperplane techniques. J. Glob. Optim. **71**(4), 987–1011 (2018). https://doi.org/10.1007/s10898-018-0644-z