A SEQUENTIAL ALGORITHM FOR SOLVING NONLINEAR OPTIMIZATION PROBLEMS WITH CHANCE CONSTRAINTS

FRANK E. CURTIS*, ANDREAS WÄCHTER†, AND VICTOR M. ZAVALA‡

Dedicated to Roger Fletcher, a great pioneer for the field of nonlinear optimization.

Abstract. An algorithm is presented for solving nonlinear optimization problems with chance constraints, i.e., those in which a constraint involving an uncertain parameter must be satisfied with at least a minimum probability. In particular, the algorithm is designed to solve cardinality-constrained nonlinear optimization problems that arise in sample average approximations of chance-constrained problems, as well as in other applications in which it is only desired to enforce a minimum number of constraints. The algorithm employs a novel penalty function, which is minimized sequentially by solving quadratic optimization subproblems with linear cardinality constraints. Properties of minimizers of the penalty function in relation to minimizers of the corresponding nonlinear optimization problem are presented, and convergence of the proposed algorithm to a stationary point of the penalty function is proved. The effectiveness of the algorithm is demonstrated through numerical experiments with a nonlinear cash flow problem.

Key words. nonlinear optimization, chance constraints, cardinality constraints, sample average approximation, exact penalization, sequential quadratic optimization, trust region methods

AMS subject classifications. 90C15, 90C30, 90C55

1. Introduction.

The focus of this paper is the proposal of an algorithm for solving nonlinear optimization problems with *cardinality constraints*. Such problems are those in which only a subset of constraints is desired to be enforced. Similar to penalty-function-based sequential quadratic optimization (penalty-SQP) methods for solving standard nonlinear optimization problems, the algorithm that we propose is based on the sequential minimization of a penalty function. Each search direction is computed by solving a subproblem involving Taylor approximations of the problem objective and constraint functions. An important application area in which such cardinality-constrained problems arise—and the area in which we believe that our proposed algorithm can be particularly effective—is that of *chance-constrained optimization*. Hence, to put the contribution and ideas presented in this paper in context, we briefly review the notion of a chance-constrained optimization problem and the numerical methods that have been proposed for solving such problems. In short, our proposed algorithm can be seen as a technique for extending, to nonlinear settings, the ideas originally proposed by Luedtke et al. [34, 35, 36] and others (e.g., see [29]) for solving linear chance-constrained problems.

Broadly, the subject of interest in this paper is that of optimization problems involving uncertain parameters, which are pervasive throughout science and engineering. Modern approaches to handling uncertainty in optimization involve formulating and solving *stochastic optimization* problems. For example, if the uncertain parameters appear in an objective function related to a system performance measure, then

*Department of Industrial and Systems Engineering, Lehigh University. This author was partially supported by U.S. Department of Energy grant DE-SC0010615 and U.S. National Science Foundation grant DMS-1319356. E-mail: frank.e.curtis@gmail.com

†Department of Industrial Engineering and Management Sciences, Northwestern University. This author was partially supported by U.S. National Science Foundation grant DMS-1522747. E-mail: waechter@iems.northwestern.edu

‡Department of Chemical and Biological Engineering, University of Wisconsin-Madison. This author was partially supported by U.S. Department of Energy grant DE-SC0014114. E-mail: victor.zavala@wisc.edu

a standard technique has been to optimize the expected system performance. In practice, this can be done either by assuming certain probability distributions for the uncertain parameters, or approximately by optimizing over a discrete distribution defined by a known set of scenarios (or realizations). In any case, an issue with such a strategy is that it does not safeguard against potentially high variability of the system performance; moreover, it does not protect against poor worst-case performance. Hence, popular alternative methodologies have focused on optimization over all or worst-case scenarios. In *robust optimization* [4, 6], one formulates and solves a problem over the worst-case when the uncertain parameters are presumed to lie in a tractable uncertainty set. Alternatively, one can formulate a problem in which the constraints are to be satisfied *almost surely*, i.e., they must hold with probability one. Again, in practice, this can be done with presumed knowledge of distributions of the uncertain parameters, or by satisfying all constraints defined by a sufficiently large set of scenarios of the uncertain parameters [8]. The downside of all of these techniques is that they can lead to quite conservative decisions in many practical situations.

A modeling paradigm that offers more flexibility is that of optimization with *chance constraints* (or *probabilistic constraints*). Chance-constrained optimization problems (CCPs) are notable in that they offer the decision maker the ability to dictate the probability of dissatisfaction that they are willing to accept. Formally, given an objective function $f : \mathbb{R}^n \to \mathbb{R}$, a vector constraint function $c : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$, and a probability of dissatisfaction $\alpha \in [0, 1]$, a basic CCP has the form

$$\text{(CCP)} \qquad \min_{x \in \mathbb{R}^n} \; f(x) \quad \text{s.t.} \quad \mathbb{P}[c(x, \Xi) \leq 0] \geq 1 - \alpha.$$

Here, $\Xi : \Omega \to \mathbb{R}^p$ is a random variable with associated probability space $(\Omega, \mathcal{F}, P)$. Contained within this paradigm are optimization models in which the constraints must be satisfied almost surely, i.e., when the probability of dissatisaction is $\alpha = 0$.

Starting with the first publications over 60 years ago [13, 14], a large body of research has evolved that addresses the properties and numerical solution of CCPs. Since then, chance-constrained optimization problem formulations have been studied in a variety of application areas including supply chain management [31], power systems [7], production planning [37], finance [9, 19, 42], water management [1, 20, 33, 46], chemical processes [24], telecommunications [2, 32, 45], and mining [12].

In terms of numerical solution methods for solving CCPs, most approaches have aimed at computing a global minimizer, at least in some sense. For example, when the CCP itself is convex, one can apply standard convex optimization techniques to find a global minimizer [11, 13, 26, 43, 44]. On the other hand, if the CCP is not convex, then one may instead be satisfied by finding a global minimizer of an approximate problem constructed so that the feasible region of the chance constraint is convex [9, 15, 38, 40, 41]. Global solutions of nonconvex CCPs can also be found using combinatorial techniques, which may be effective when one can exploit problem-specific structures such as separability of the chance constraint, linearity (or at least convexity) of the almost-sure constraint, and/or explicit knowledge of an α-concave distribution of the random variable Ξ; e.g., see [5, 18, 30].

Additional opportunities arise when Ξ is discrete (i.e., when $\Omega := \{\xi_1, \xi_2, \ldots, \xi_N\}$ is finite) and the associated realizations have equal probability of $1/N$. Such a situation might arise naturally, but also arises when one approximates problem (CCP) by drawing a number of samples of the random variable (either from a presumed distribution or from a set of historical data) to construct a *sample average approximation*

(SAA) [28, 44]. This leads to a cardinality-constrained problem

$$\text{(P)} \qquad \min_{x\in\mathbb{R}^n} f(x) \quad \text{s.t.} \quad |\{i \in \mathcal{I} : c_i(x) \le 0\}| \ge M,$$

where $|\cdot|$ denotes the cardinality of a set, $c_i : \mathbb{R}^n \to \mathbb{R}^m$ for all $i \in \mathcal{I} := \{1, \ldots, N\}$, and $M \in \{0, 1, \ldots, N\}$. We denote the elements of the vector function c_i as c_{ij} with $j = 1, ..., m$. In an SAA of problem (CCP), one obtains (P) with $c_i := c(\cdot, \xi_i)$ for all $i \in \mathcal{I}$ and $M := \lceil (1-\alpha)N \rceil$. Note that the cardinality of the set of constraints that are allowed to be violated in (P) is $N - M$ (corresponding to the probability level α).

Optimization problems of the form (P) also arise in other decision-making settings. For example, each constraint $c_i(x) \le 0$ might represent the satisfaction of demand for a particular set of customers (or interested parties), and one may be interested in making an optimal decision subject to satisfying the demands of most, if not all, sets of customers. This situation is found, for instance, in power grid systems where the system operator might need to select a subset of demands to curtail or shed to prevent a blackout [27]. Instances of (P) also arises in model predictive control when the controller has flexibility to violate constraints at certain times. These so-called soft constraints are often dealt with using penalization schemes. However, these can be difficult to tune to achieve a desired behavior [16]. Cardinality-constrained formulations offer the ability to achieve the desired effect directly.

In this paper, we propose, analyze, and provide the results of numerical experiments for a new method for solving problems of the form (P) when the objective function f and constraint functions $\{c_1, \ldots, c_N\}$ are continuously differentiable. Our algorithm and analysis can readily be extended for situations in which the codomain of c_i varies with $i \in \mathcal{I}$, there are multiple cardinality constraints, and/or the problem also involves other smooth nonlinear equality and inequality constraints. However, for ease of exposition, we present our approach in the context of problem (P) and return to discuss extensions in the description of our software implementation. For cases when f and $\{c_1, \ldots, c_N\}$ are linear, recent methods have been proposed for solving such problems using mixed-integer linear optimization techniques [29, 34, 35, 36]. In our method, we allow these problems functions to be nonlinear and even nonconvex.

In summary, problems (CCP) and (P) represent powerful classes of optimization models, and the purpose of this paper is to propose an algorithm for solving instances of (P), which may arise as approximations of (CCP). We stress that solving such instances is extremely challenging. Even when f and each c_i are linear, the feasible region of (P) is nonconvex and the problem typically has many local minima. Rather than resorting to nonlinear mixed-integer optimization (MINLP) techniques, the method that we propose performs a local search based on first-order and, if desired, second-order derivatives of the problem functions. In this manner, we are only able to prove that our algorithm guarantees that a stationarity measure will vanish, not necessarily that a global minimizer will be found. That being said, we claim that our approach is of interest for a least a few reasons. First, we claim that numerical methods based on convex relaxations of nonlinear chance-constrained problems can lead to conservative decisions in many practical situations and that, in such cases, solving nonlinear cardinality-constrained problems can offer improved results. Second, for large-scale problems, applying MINLP techniques to solve instances of (P) may be intractable, in which case a local search method such as ours might represent the only viable alternative. Indeed, we demonstrate in our numerical experiments that our approach can find high quality (i.e., nearly globally optimal) solutions.

It is worth emphasizing that if one were to employ our approach for solving (P) with the underlying goal of solving an instance of (CCP), then one would also need to consider advanced scenario sampling schemes, such as those discussed in [25]. While this is an important consideration, our focus in this paper is on a technique for solving a given instance of (P), including its convergence properties and practical performance. This represents a critical first step toward the design of numerical methods for solving large-scale instances of (CCP). We leave the remaining issues toward solving (CCP), such as sample set size and selection, as future work.

The paper is organized as follows. In §2, we introduce preliminary definitions and notation. Specifically, since our algorithm is based on a technique of *exact penalization*, we introduce concepts and notation related to constraint violation measures and penalty functions that will be used in the design of our algorithm and its analysis. We then present our algorithm in §3 and a global convergence theory for it in §4. In §5, we present the results of numerical experiments with an implementation of our method when applied to solve a simple illustrative example as well as a nonlinear cash flow optimization problem. These results show that our method obtains high quality solutions to nonconvex test instances. We end with concluding remarks in §6.

2. Exact Penalty Framework.

Our problem of interest (P) requires that a subset of the constraints $\{i \in \mathcal{I} : c_i(x) \leq 0\}$ be satisfied. In particular, the cardinality of the subset of constraints that are satisfied must be at least M. Hence, we define $\mathfrak{S}_M \subseteq \mathcal{I}$ as the set of all lexicographically ordered subsets of the constraint index set $\mathcal{I}$ with cardinality M. With problem (CCP) in mind, we refer to each subset in $\mathfrak{S}_M$ as a scenario selection. For any lexicographically ordered subset $\mathcal{S} \subseteq \mathcal{I}$ (not necessarily in $\mathfrak{S}_M$), let $c_{\mathcal{S}}(x)$ denote the $(m|\mathcal{S}|)$-dimensional vector consisting of a concatenation of all $c_i(x)$ with $i \in \mathcal{S}$. For convenience, we also let $c(x) := c_{\mathcal{I}}(x)$.

With this notation and letting $[\cdot]_+ := \max\{\cdot, 0\}$ (componentwise), we measure the constraint violation for (P) through the function $\langle\!\langle \cdot \rangle\!\rangle_M : \mathbb{R}^{mN} \to \mathbb{R}$ defined by

$$\langle\!\langle w \rangle\!\rangle_M := \min_{\mathcal{S} \in \mathfrak{S}_M} \|[w_{\mathcal{S}}]_+\|_1. \tag{1}$$

In particular, given $x \in \mathbb{R}^n$, its constraint violation with respect to (P) is $\langle\!\langle c(x) \rangle\!\rangle_M$, which is equal to the minimum value of the ℓ_1-norm constraint violation $\|[c_{\mathcal{S}}(x)]_+\|_1$ over all possible choices of the scenario selection $\mathcal{S}$ from $\mathfrak{S}_M$. Note that $\langle\!\langle \cdot \rangle\!\rangle_M$ is Lipschitz continuous, since it is the pointwise minimum of Lipschitz continuous functions. Also note that when $M = N$, we have that $\langle\!\langle c(x) \rangle\!\rangle_M = \|[c(x)]_+\|_1$, which corresponds to the ℓ_1-norm constraint violation of the standard nonlinear optimization problem:

$$\min_{x \in \mathbb{R}^n} \ f(x) \quad \text{s.t.} \quad c_i(x) \leq 0 \ \text{ for all } \ i \in \mathcal{I}. \tag{RP}$$

This is a "robust" counterpart of (P) in which all constraints are enforced.

A useful and intuitive alternative definition of the violation measure (1) which also takes into account the amount by which the relevant constraints are satisfied is derived in the following manner. First, let $v : \mathbb{R}^{mN} \times \mathcal{I} \to \mathbb{R}$ be defined by

$$v(w, i) := \begin{cases} \|[w_i]_+\|_1 & \text{if } \|[w_i]_+\|_1 > 0 \\ \max_{j \in \{1,\ldots,m\}} \{w_{i,j}\} & \text{otherwise.} \end{cases} \tag{2}$$

The value $v(c(x), i)$ measures the ℓ_1-norm violation of the constraint $c_i(x) \leq 0$ if this violation is positive; however, if this constraint is satisfied, then it provides the element of (the vector) $c_i(x)$ that is closest to the threshold of zero. The function v

also induces an ordering of the entire set of constraint function indices. In particular, for a given $w \in \mathbb{R}^{mN}$, let us define $\{i_{w,1}, i_{w,2}, \ldots, i_{w,N}\}$ such that

$$v(w, i_{w,1}) \leq v(w, i_{w,2}) \leq \cdots \leq v(w, i_{w,N}). \tag{3}$$

To make this ordering well-defined, we assume that ties are broken lexicographically; i.e., if $v(w, i_{w,j_1}) = v(w, i_{w,j_2})$ and $j_1 \leq j_2$, then $i_{w,j_1} \leq i_{w,j_2}$. It can now be easily verified that our constraint violation measure is given by

$$\langle\langle c(x) \rangle\rangle_M = \sum_{j=1}^{M} \max\{0, v(c(x), i_{c(x),j})\} = \|[c_{\mathcal{S}(x)}(x)]_+\|_1, \tag{4}$$

where $\mathcal{S}(x) := \{i_{c(x),1}, \ldots, i_{c(x),M}\} \in \mathfrak{S}_M$.

With these definitions of constraint violation measures in place, we now define a penalty function for problem (P) as $\phi(\cdot; \rho) : \mathbb{R}^n \to \mathbb{R}$ such that, for any $x \in \mathbb{R}^n$,

$$\phi(x; \rho) := \rho f(x) + \langle\langle c(x) \rangle\rangle_M.$$

This penalty function can be expressed in terms of the standard exact ℓ_1-norm penalty function $\phi_{\mathcal{S}}(\cdot; \rho) : \mathbb{R}^n \to \mathbb{R}$ given by

$$\phi_{\mathcal{S}}(x; \rho) := \rho f(x) + \|[c_{\mathcal{S}}(x)]_+\|_1, \tag{5}$$

which takes into account only those constraints with indices in $\mathcal{S} \subseteq \mathcal{I}$. In particular, from (1), it follows that, for any $x \in \mathbb{R}^n$,

$$\phi(x; \rho) = \min_{\mathcal{S} \in \mathfrak{S}_M} \phi_{\mathcal{S}}(x; \rho). \tag{6}$$

Importantly, we can characterize local minima of the penalty function ϕ by means of local minima of $\phi_{\mathcal{S}}$. In order to do this, we define a few additional quantities. First, for a given $x \in \mathbb{R}^n$, we define the critical value $v_M(x) := v(c(x), i_{c(x),M})$. This can be viewed as the Mth smallest ℓ_1-norm constraint violation or, if at least M constraints are satisfied, it can be interpreted as the distance to the zero threshold corresponding to the Mth most satisfied constraint. One can view this as the *value-at-risk* of $\{v(c(x), i_{c(x),j})\}_{j=1}^N$ corresponding to the confidence level α. The problem (P) can be written as $\min f(x)$ s.t. $v_M(x) \leq 0$.

For a given $\epsilon \geq 0$, we define a partition of the indices in $\mathcal{I}$ through the subsets

$$\begin{aligned}
\mathcal{P}(x; \epsilon) &:= \{i \in \mathcal{I} : v(c(x), i) - v_M(x) > \epsilon\}, \\
\mathcal{N}(x; \epsilon) &:= \{i \in \mathcal{I} : v(c(x), i) - v_M(x) < -\epsilon\}, \\
\text{and} \ \ \mathcal{C}(x; \epsilon) &:= \{i \in \mathcal{I} : |v(c(x), i) - v_M(x)| \leq \epsilon\}.
\end{aligned}$$

One may think of $\mathcal{P}$ as determining the indices corresponding to constraint violations that are sufficiently larger than the critical value $v_M(x)$, $\mathcal{N}$ as determining indices that are sufficiently smaller than $v_M(x)$, and $\mathcal{C}$ as determining constraint violations that are close to the critical value. We now define the set of critical scenario selections

$$\mathfrak{S}_M(x; \epsilon) := \{\mathcal{S} \in \mathfrak{S}_M : \mathcal{N}(x; \epsilon) \subseteq \mathcal{S} \text{ and } \mathcal{S} \subseteq \mathcal{N}(x; \epsilon) \cup \mathcal{C}(x; \epsilon)\}. \tag{7}$$

Importantly, this set includes those scenario selections that are ϵ-critical in the sense that their violation measure is within ϵ of the critical value $v_M(x)$.

Using these definitions, it follows as in (6) that

$$\phi(x;\rho) = \min_{\mathcal{S}\in\mathfrak{S}_M} \phi_{\mathcal{S}}(x;\rho) = \phi_{\tilde{\mathcal{S}}}(x;\rho) \ \text{ for all } \ \tilde{\mathcal{S}} \in \mathfrak{S}_M(x;0), \tag{8}$$

where we have used that the scenario selections in $\mathfrak{S}_M(x,0)$ each correspond to minimizers of the constraint violation measure for a given $x \in \mathbb{R}^n$. We also have that

$$\mathfrak{S}_M(x;\epsilon_1) \subseteq \mathfrak{S}_M(x;\epsilon_2) \ \text{ if } \ \epsilon_1 \in [0,\epsilon_2], \tag{9}$$

from which it follows that

$$\phi(x;\rho) = \min_{\mathcal{S}\in\mathfrak{S}_M(x;\epsilon)} \phi_{\mathcal{S}}(x;\rho) \ \text{ for all } \ \epsilon \geq 0, \tag{10}$$

where again we have used that the scenario selections in $\mathfrak{S}_M(x,0) \subseteq \mathfrak{S}_M(x,\epsilon)$ (for any $\epsilon \geq 0$) correspond to minimizers of the constraint violation measure.

We now prove the following lemma relating minimizers of the penalty function ϕ with those of the exact ℓ_1-norm penalty function $\phi_{\mathcal{S}}$ for certain $\mathcal{S}$.

Lemma 1. *Suppose the functions f and c_i for all $i \in \mathcal{I}$ are continuous. For any $\rho \geq 0$, a point $x_* \in \mathbb{R}^n$ is a local minimizer of $\phi(\cdot;\rho)$ if and only if it is a local minimizer of $\phi_{\mathcal{S}}(\cdot;\rho)$ for all $\mathcal{S} \in \mathfrak{S}_M(x_*;0)$.*

Proof. Suppose that x_* is not a local minimizer of $\phi_{\mathcal{S}}(\cdot;\rho)$ for all $\mathcal{S} \in \mathfrak{S}_M(x_*;0)$. That is, suppose that there exists $\bar{\mathcal{S}} \in \mathfrak{S}_M(x_*;0)$ such that x_* is not a local minimizer of $\phi_{\bar{\mathcal{S}}}(\cdot;\rho)$. Then, there exists a sequence $\{\overline{x}_j\}_{j=0}^{\infty} \subset \mathbb{R}^n$ converging to x_* such that $\phi_{\bar{\mathcal{S}}}(\overline{x}_j;\rho) < \phi_{\bar{\mathcal{S}}}(x_*;\rho)$ for all $j \in \{0,1,2,\dots\}$. Along with (8), this means that

$$\phi(\overline{x}_j;\rho) \leq \phi_{\bar{\mathcal{S}}}(\overline{x}_j;\rho) < \phi_{\bar{\mathcal{S}}}(x_*;\rho) = \phi(x_*;\rho) \ \text{ for all } \ j \in \{0,1,2,\dots\},$$

implying that x_* is not a local minimizer of $\phi(\cdot;\rho)$.

Now suppose that x_* is not a local minimizer of $\phi(\cdot;\rho)$, meaning that there exists a sequence $\{\hat{x}_j\}_{j=0}^{\infty} \subset \mathbb{R}^n$ converging to x_* such that $\phi(\hat{x}_j;\rho) < \phi(x_*;\rho)$ for all $j \in \{0,1,2,\dots\}$. By the definition of $\mathfrak{S}_M(x_*;0)$, we have from (8) that

$$\langle\!\langle c(x_*)\rangle\!\rangle_M = \min_{\mathcal{S}\in\mathfrak{S}_M} \|[c_{\mathcal{S}}(x_*)]_+\|_1 = \|[c_{\hat{\mathcal{S}}}(x_*)]_+\|_1 \ \text{ for all } \ \hat{\mathcal{S}} \in \mathfrak{S}_M(x_*;0).$$

Let $\overline{\mathfrak{S}_M^*} := \mathfrak{S}_M \setminus \mathfrak{S}_M(x_*;0)$ and $\bar{c} := \min_{\bar{\mathcal{S}}\in\overline{\mathfrak{S}_M^*}} \|[c_{\bar{\mathcal{S}}}(x_*)]_+\|_1$. We then have, for all $\bar{\mathcal{S}} \in \overline{\mathfrak{S}_M^*}$, that $\|[c_{\bar{\mathcal{S}}}(x_*)]_+\|_1 \geq \bar{c} > \langle\!\langle c(x_*)\rangle\!\rangle_M$. From continuity of the constraint functions and finiteness of the set $\overline{\mathfrak{S}_M^*}$, there exists $\hat{j} \in \{0,1,2,\dots\}$ such that

$$\|[c_{\bar{\mathcal{S}}}(\hat{x}_j)]_+\|_1 > \frac{\bar{c} + \langle\!\langle c(x_*)\rangle\!\rangle_M}{2} > \langle\!\langle c(\hat{x}_j)\rangle\!\rangle_M \ \text{ for all } \ j \geq \hat{j} \ \text{ and } \ \bar{\mathcal{S}} \in \overline{\mathfrak{S}_M^*}. \tag{11}$$

For any $j \geq \hat{j}$, choose some $\hat{\mathcal{S}}_j \in \mathfrak{S}_M(\hat{x}_j;0)$ (i.e., so that $\|[c_{\hat{\mathcal{S}}_j}(\hat{x}_j)]_+\|_1 = \langle\!\langle c(\hat{x}_j)\rangle\!\rangle_M$). By (11), we must have that $\hat{\mathcal{S}}_j \notin \overline{\mathfrak{S}_M^*}$, meaning that $\hat{\mathcal{S}}_j \in \mathfrak{S}_M(x_*;0)$. Since $\mathfrak{S}_M(x_*;0)$ is finite, we may assume without loss of generality that $\hat{\mathcal{S}}_j = \hat{\mathcal{S}}$ for some $\hat{\mathcal{S}} \in \mathfrak{S}_M(x_*;0)$. Finally, we obtain from our supposition and (8) that, for all $j \geq \hat{j}$,

$$\phi_{\hat{\mathcal{S}}}(\hat{x}_j;\rho) \overset{\hat{\mathcal{S}}\in\mathfrak{S}_M(\hat{x}_j,0)}{=} \phi(\hat{x}_j;\rho) < \phi(x_*;\rho) \overset{\hat{\mathcal{S}}\in\mathfrak{S}_M(x_*,0)}{=} \phi_{\hat{\mathcal{S}}}(x_*;\rho).$$

Therefore, x_* is not a local minimizer of $\phi_{\hat{\mathcal{S}}}(\cdot;\rho)$ where $\hat{\mathcal{S}} \in \mathfrak{S}_M(x_*;0)$. ◻

We also have the following lemmas which justify our choice of penalty function. For similar classical results for nonlinear optimization, see, e.g. [23, Thm. 4.1].

LEMMA 2. *Suppose the functions f and c_i for all $i \in \mathcal{I}$ are continuous, and let $x_* \in \mathbb{R}^n$. Then, the following hold true.*

(*i*) *If x_* is feasible for problem* (P) *and is a local minimizer of $\phi(\cdot;\rho)$ for some $\rho > 0$, then x_* is a local minimizer of* (P).

(*ii*) *If x_* is infeasible for problem* (P) *and there exists an open ball $\mathcal{B}(x_*, \theta)$ about x_* with radius $\theta > 0$ and some $\bar{\rho} > 0$ such that x^* is a minimizer of $\phi(\cdot;\rho)$ in $\mathcal{B}(x_*, \theta)$ for all $\rho \in (0, \bar{\rho}]$, then x_* is a local minimizer of the constraint violation measure $\langle\langle c(\cdot)\rangle\rangle_M$.*

Proof. Suppose x_* is feasible for (P) and a local minimizer of $\phi(\cdot;\rho)$ for some $\rho > 0$. Then, there exists an open ball $\mathcal{B}(x_*, \theta)$ about x_* with radius $\theta > 0$ such that $\phi(x_*;\rho) \leq \phi(x;\rho)$ for all $x \in \mathcal{B}(x_*, \theta)$. Let $\bar{x}$ be any point in $\mathcal{B}(x_*, \theta)$ that is feasible for problem (P), i.e., such that $\langle\langle c(\bar{x})\rangle\rangle_M = 0$. Then, $f(x_*) = \phi(x_*;\rho) \leq \phi(\bar{x};\rho) = f(\bar{x})$. Since $\bar{x}$ was chosen as an arbitrary feasible point in $\mathcal{B}(x_*, \theta)$, this implies that x_* is a minimizer of (P) in $\mathcal{B}(x_*, \theta)$, i.e., it is a local minimizer of problem (P).

Now suppose that x_* is infeasible for problem (P), i.e., suppose $\langle\langle c(x_*)\rangle\rangle_M > 0$, and that there exists $\theta > 0$ and $\bar{\rho} > 0$ such that x_* is a minimizer of $\phi(\cdot;\rho)$ in an open ball $\mathcal{B}(x_*, \theta)$ about x_* with radius θ for all $\rho \in (0, \bar{\rho}]$. In order to derive a contradiction, suppose that x_* is not a minimizer of $\langle\langle c(\cdot)\rangle\rangle_M$ in $\mathcal{B}(x_*, \theta)$. Then, there exists a sequence $\{\hat{x}_j\}_{j=0}^{\infty} \subset \mathcal{B}(x_*, \theta)$ converging to x_* such that $\langle\langle c(\hat{x}_j)\rangle\rangle_M < \langle\langle c(x_*)\rangle\rangle_M$ for all $j \in \{0, 1, 2, \dots\}$. This, along with the fact that the properties of x_* imply that $\phi(\hat{x}_j;\bar{\rho}) \geq \phi(x_*;\bar{\rho})$ for all $j \in \{0, 1, 2, \dots\}$, means that

$$f(\hat{x}_j) - f(x_*) \geq (\langle\langle c(x_*)\rangle\rangle_M - \langle\langle c(\hat{x}_j)\rangle\rangle_M)/\bar{\rho} > 0$$

for all $j \in \{0, 1, 2, \dots\}$. Then, for any fixed $\rho \in (0, \bar{\rho}]$ with

$$\rho < (\langle\langle c(x_*)\rangle\rangle_M - \langle\langle c(\hat{x}_j)\rangle\rangle_M)/(f(\hat{x}_j) - f(x_*)),$$

it follows that

$$\phi(\hat{x}_j;\rho) = \rho f(\hat{x}_j) + \langle\langle c(\hat{x}_j)\rangle\rangle_M < \rho f(x_*) + \langle\langle c(x_*)\rangle\rangle_M = \phi(x_*;\rho),$$

contradicting that x_* is a minimizer of $\phi(\cdot;\rho)$ in $\mathcal{B}(x_*, \theta)$. We may conclude that x_* is a minimizer of $\langle\langle c(\cdot)\rangle\rangle_M$ in $\mathcal{B}(x_*, \theta)$, i.e., it is a local minimizer of $\langle\langle c(\cdot)\rangle\rangle_M$. ∎

LEMMA 3. *Suppose the functions f and c_i for all $i \in \mathcal{I}$ are continuously differentiable, and let $x_* \in \mathbb{R}^n$ be a local minimizer of* (P). *Furthermore, suppose that, for each $\mathcal{S} \in \mathfrak{S}_M(x_*;0)$, a constraint qualification holds for the optimization problem $\min_{x\in\mathbb{R}^n} f(x)$ such that $c_{\mathcal{S}}(x) \leq 0$ in that there exists $\bar{\rho}_{\mathcal{S}} > 0$ such that x_* is a local minimizer of $\phi_{\mathcal{S}}(\cdot;\rho)$ for all $\rho \in (0, \bar{\rho}_{\mathcal{S}}]$. Then, there exists $\bar{\rho} > 0$ such that x_* is a local minimizer of $\phi(\cdot;\rho)$ for all $\rho \in (0, \bar{\rho}]$.*

Proof. The proof follows easily for $\bar{\rho} = \min\{\bar{\rho}_{\mathcal{S}} : \mathcal{S} \in \mathfrak{S}_M(x_*, 0)\} > 0$. ∎

An example of a constraint qualification that implies the conclusion of Lemma 3 is the linear independence constraint qualification (LICQ); however, many other less restrictive constraint qualifications are also sufficient [3, 39].

3. Algorithm Description.

Our algorithm is inspired by sequential quadratic optimization (commonly known as SQP) methods for solving nonlinear optimization problems that use a trust region (TR) mechanism for promoting global convergence. In particular, following standard penalty-SQP techniques [21, 22], in this section we describe a sequential method in which trial steps are computed through piecewise-quadratic penalty function models. The subproblems in this approach have quadratic objective functions and linear cardinality constraints that can be solved with tailored methods such as those proposed in [34, 35, 36].

We focus here on a method that employs a fixed penalty parameter $\rho \in (0, \infty)$ and a fixed $\epsilon \in (0, \infty)$ to define sets of scenario selections as in (7). It is this method with fixed (ρ, ϵ) for which we provide global convergence guarantees in §4. To simplify our notation here and in §4, we drop ρ from function expressions in which they appear; e.g., we use $\phi(\cdot) \equiv \phi(\cdot; \rho)$, $\phi_{\mathcal{S}}(\cdot) \equiv \phi_{\mathcal{S}}(\cdot; \rho)$, etc.

Let $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ be the gradient function of $f(\cdot)$ and, for a given $\mathcal{S} \subseteq \mathcal{I}$, let $\nabla c_{\mathcal{S}} : \mathbb{R}^n \to \mathbb{R}^{n \times (m|\mathcal{S}|)}$ be the transpose of the Jacobian function of $c_{\mathcal{S}}(\cdot)$. For a given iterate $x_k \in \mathbb{R}^n$, scenario selection $\mathcal{S} \subseteq \mathcal{I}$, and symmetric positive semidefinite matrix $H_k \in \mathbb{R}^{n \times n}$, we define convex piecewise-linear and piecewise-quadratic local models of $\phi_{\mathcal{S}}$ at x_k as the functions $l_{\mathcal{S},k} : \mathbb{R}^n \to \mathbb{R}$ and $q_{\mathcal{S},k} : \mathbb{R}^n \to \mathbb{R}$, respectively, where

$$l_{\mathcal{S},k}(d) := \rho(f(x_k) + \nabla f(x_k)^T d) + \|[c_{\mathcal{S}}(x_k) + \nabla c_{\mathcal{S}}(x_k)^T d]_+\|_1 \tag{12a}$$

$$\text{and} \quad q_{\mathcal{S},k}(d) := l_{\mathcal{S},k}(d) + \tfrac{1}{2} d^T H_k d. \tag{12b}$$

We then define the reductions in these models corresponding to a given $d \in \mathbb{R}^n$ as

$$\Delta l_{\mathcal{S},k}(d) := l_{\mathcal{S},k}(0) - l_{\mathcal{S},k}(d) \quad \text{and} \quad \Delta q_{\mathcal{S},k}(d) := q_{\mathcal{S},k}(0) - q_{\mathcal{S},k}(d).$$

Within a standard TR penalty-SQP methodology, a trial step toward minimizing $\phi_{\mathcal{S}}(\cdot)$ from x_k is computed as a minimizer of $q_{\mathcal{S},k}(\cdot)$ within a bounded region. Extending this for the minimization of ϕ (recall (8)), it is natural to consider a trial step as a minimizer of a local quadratic-linear model of ϕ, defined as $q_k : \mathbb{R}^n \to \mathbb{R}$ where

$$q_k(d) := \min_{\mathcal{S} \in \mathfrak{S}_M} q_{\mathcal{S},k}(d_k). \tag{13}$$

A method based around this idea is stated formally as Algorithm 1 below. As is common in TR methods, the ratio r_k, which essentially measures the accuracy of the predicted progress in the penalty function, is used to decide whether a trial point should be accepted and how the trust-region radius should be updated.

A critically important feature of Algorithm 1 is that the scenario selections $\mathfrak{S}_{M,k}$ considered in subproblem (14) do not need to contain all possible scenario selections in $\mathfrak{S}_M$, in contrast to what (13) suggests. Choosing a smaller subset makes it possible to reduce the computational effort of the step computation, which is the most computationally-intensive subroutine of the algorithm. In particular, it requires a solution of a cardinality-constrained problem (with linear constraints and a quadratic objective). For example, if $\mathfrak{S}_{M,k} = \mathfrak{S}_M(x_k; \epsilon)$, then subproblem (14) is equivalent to

$$\begin{aligned} \min_{\substack{d \in \mathbb{R}^n \\ s_i \in \mathbb{R}^m \\ \forall i \in \mathcal{N}_k \cup \mathcal{C}_k}} \quad & \rho(f(x_k) + \nabla f(x)^T d) + \tfrac{1}{2} d^T H_k d + \sum_{i \in \mathcal{C}_k \cup \mathcal{N}_k} e_m^T s_i \\ \text{s.t.} \quad & \begin{cases} |\{i \in \mathcal{C}_k : c_i(x_k) + \nabla c_i(x_k)^T d \le s_i\}| \ge M - |\mathcal{N}_k|, \\ c_i(x_k) + \nabla c_i(x_k)^T d \le s_i \quad \text{for all} \quad i \in \mathcal{N}_k, \\ -\delta_k e_n \le d \le \delta_k e_n, \\ s_i \ge 0 \quad \text{for all} \quad i \in \mathcal{C}_k \cup \mathcal{N}_k, \end{cases} \end{aligned} \tag{18}$$

Algorithm 1 SQP Algorithm for Problem (P) (for fixed (ρ, ϵ))

Require: penalty parameter $\rho \in (0, \infty)$, criticality parameter $\epsilon \in (0, \infty)$, initial point $x_0 \in \mathbb{R}^n$, initial TR radius $\delta_0 \in (0, \infty)$, sufficient decrease constant $\mu \in (0, \infty)$, TR norm $\|\cdot\|$, and TR update constants $\beta_1 \in (1, \infty)$, $\beta_2 \in (0, 1)$, and $\delta_{\text{reset}} \in (0, \delta_0]$

for $k \in \mathbb{N} := \{0, 1, 2, \dots\}$ **do**

  Choose $\mathfrak{S}_{M,k} \supseteq \mathfrak{S}_M(x_k; \epsilon)$.

  Compute a trial step d_k by solving the TR subproblem

$$\min_{d \in \mathbb{R}^n} \min_{\mathcal{S} \in \mathfrak{S}_{M,k}} q_{\mathcal{S},k}(d) \quad \text{s.t.} \quad \|d\| \le \delta_k. \tag{14}$$

  Compute the actual-to-prediction reduction ratio

$$r_k \leftarrow \frac{\min_{\mathcal{S} \in \mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k) - \min_{\mathcal{S} \in \mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k + d_k)}{\min_{\mathcal{S} \in \mathfrak{S}_{M,k}} q_{\mathcal{S},k}(0) - \min_{\mathcal{S} \in \mathfrak{S}_{M,k}} q_{\mathcal{S},k}(d_k)}. \tag{15}$$

  Update the iterate and trust region radius by

$$x_{k+1} \leftarrow \begin{cases} x_k + d_k & \text{if } r_k \ge \mu \\ x_k & \text{otherwise} \end{cases} \tag{16}$$

$$\text{and} \quad \delta_{k+1} \leftarrow \begin{cases} \max\{\beta_1 \delta_k, \delta_{\text{reset}}\} & \text{if } r_k \ge \mu \\ \beta_2 \|d_k\| & \text{otherwise.} \end{cases} \tag{17}$$

end for

where $(e_m, e_n) \in \mathbb{R}^m \times \mathbb{R}^n$ are vectors of ones, $\mathcal{N}_k := \mathcal{N}(x_k, \epsilon)$, and $\mathcal{C}_k := \mathcal{C}(x_k, \epsilon)$. Clearly, the complexity of solving this problem depends on $|\mathcal{C}(x_k; \epsilon)|$.

This being said, to ensure convergence, it is necessary to include in $\mathfrak{S}_{M,k}$ at least all ϵ-critical scenario selections $\mathfrak{S}_M(x_k; \epsilon)$ for some fixed $\epsilon > 0$, since otherwise the algorithm might have a myopic view of the feasible region around x_k, causing it to ignore constraints that are asymptotically relevant to characterize the local behavior of ϕ at a limit point x_{lim}. More precisely, consider a situation in which there is a constraint $i \in \mathcal{I}$ that has $v(c(x_k), i) > v_M(x_k)$ for all $k \in \mathbb{N}$, but $v(c(x_{\text{lim}}), i) = v_M(x_{\text{lim}})$. In such a situation, one finds $i \notin \mathcal{C}(x_k, 0)$ for all $k \in \mathbb{N}$. Therefore, no $\mathcal{S} \in \mathfrak{S}_M(x_k; 0)$ includes i, meaning that the method with $\epsilon = 0$ would compute trial steps via (14) that completely ignore the constraint function c_i. This is problematic since, at x_{lim}, one finds $i \in \mathcal{C}(x_{\text{lim}}, 0)$. Recalling Lemma 1, we see that the constraint function c_i needs to be considered to draw conclusions about the relevance of x_* as a minimizer—or even a stationary point (see (20) below)—of the penalty function ϕ.

Overall, the parameter $\epsilon > 0$ plays a critical role. If ϵ is large, then the method considers a good approximation of q_k (and, hence, of ϕ) when computing each trial step, but the computational cost of computing each step might be large. On the other hand, if ϵ is small (near zero), then the objective function in (14) might be a poor approximation of q_k (and ϕ). In such cases, since only a small subset of scenario selections is considered in (14), the method might be unable to "see" local minima of ϕ that correspond to scenario selections that are ignored, meaning that it might converge to an inferior local minimum, even if better ones are "close by". In the remainder of this section, we shall see that our analysis holds for any $\epsilon > 0$, but we will explore the crucial practical balance in the choice of ϵ in §5.2.

4. Analysis.

The main purpose of this section is to prove a global convergence result for Algorithm 1 in terms of driving a stationarity measure for the (nonconvex and nonsmooth) penalty function ϕ to zero. We also provide some commentary on the usefulness of our convergence result, particularly as it relates to the potential convergence of the method to a "poor" local minimizer; see §4.2.

Our analysis generally follows that for the penalty-SQP method in [10], which we have extended to account for our unique constraint violation measure. We stress that the analysis is not straightforward, especially since—in contrast to standard penalty-SQP—even our linear and quadratic models of the penalty function are nonconvex.

4.1. Global convergence.

We present our analysis under the following assumption. In this assumption and throughout the remainder of the section, $\|\cdot\|$ refers to the trust region norm used in Algorithm 1.

ASSUMPTION 4. *The problem functions $f : \mathbb{R}^n \to \mathbb{R}$ and $c_i : \mathbb{R}^n \to \mathbb{R}^m$ for all $i \in \mathcal{I}$ are Lipschitz continuous with Lipschitz continuous first-order derivatives over a bounded convex set whose interior contains the closure of the iterate sequence $\{x_k\}$. In addition, the sequence $\{H_k\}$ is bounded in norm in the sense that there exists a scalar $H_{\max} > 0$ such that $d^T H_k d \le H_{\max}\|d\|^2$ for all k and any $d \in \mathbb{R}^n$.*

To state the global convergence theorem that we prove, we first need to define a valid stationarity measure for ϕ. In order to do this, let us first draw from standard exact penalty function theory to define a valid measure for $\phi_{\mathcal{S}}$ for a given $\mathcal{S} \subseteq \mathcal{I}$. Similar to our piecewise-linear model $l_{\mathcal{S},k}$ for $\phi_{\mathcal{S}}$ at x_k (recall (12a)), let us define a local piecewise-linear model of $\phi_{\mathcal{S}}$ at x as $l_{\mathcal{S}}(\cdot; x) : \mathbb{R}^n \to \mathbb{R}$, as defined by

$$l_{\mathcal{S}}(d; x) = \rho(f(x) + \nabla f(x)^T d) + \|[c_{\mathcal{S}}(x) + \nabla c_{\mathcal{S}}(x)^T d]_+\|_1.$$

We denote the reduction in this model corresponding to $d \in \mathbb{R}^n$ as $\Delta l_{\mathcal{S}}(d; x) := l_{\mathcal{S}}(0; x) - l_{\mathcal{S}}(d; x)$. Letting $d^{\mathrm{L}}_{\mathcal{S}}(x)$ denote a minimizer of $l_{\mathcal{S}}(\cdot; x)$ within the $\|\cdot\|$ unit ball, or equivalently a maximizer of the reduction within the ball, i.e.,

$$d^{\mathrm{L}}_{\mathcal{S}}(x) \in \arg\max_{d \in \mathbb{R}^n} \{\Delta l_{\mathcal{S}}(d; x) : \|d\| \le 1\}, \tag{19}$$

we define the measure $\chi_{\mathcal{S}}(x) := \Delta l_{\mathcal{S}}(d^{\mathrm{L}}_{\mathcal{S}}(x); x)$. The following lemma confirms that $\chi_{\mathcal{S}}(x)$ is a valid criticality measure for $\phi_{\mathcal{S}}$.

LEMMA 5. *Let $\mathcal{S} \subseteq \mathcal{I}$. Then, $\chi_{\mathcal{S}} : \mathbb{R}^n \to \mathbb{R}$ is continuous. In addition, one has $\chi_{\mathcal{S}}(x_*) = 0$ if and only if x_* is stationary for $\phi_{\mathcal{S}}$ in the sense that $0 \in \partial\phi_{\mathcal{S}}(x_*)$.*

Proof. See, e.g., Lemma 2.1 in [48]. ∎

In light of Lemmas 1 and 5, it is now natural to define a criticality measure for ϕ in terms of the criticality measures for $\phi_{\mathcal{S}}$ for each scenario selection $\mathcal{S} \in \mathfrak{S}_M(\cdot; 0)$. Specifically, for the penalty function ϕ, we define the measure $\chi : \mathbb{R}^n \to \mathbb{R}$ by

$$\chi(x) := \max_{\mathcal{S} \in \mathfrak{S}_M(x;0)} \chi_{\mathcal{S}}(x). \tag{20}$$

We now state our global convergence result.

THEOREM 6. *Under Assumption 4, one of the following outcomes will occur from any run of Algorithm 1 for given scalars $\rho > 0$ and $\epsilon > 0$:*

(i) $\chi(x_k) = 0$ for some (finite) $k \in \mathbb{N}$ or

(ii) the set $\mathcal{K} := \{k \in \mathbb{N} : r_k \ge \mu\}$ of successful iterations has infinite cardinality and any limit point x_* of $\{x_k\}$ has $\chi(x_*) = 0$.

Remark 7. *We recover a standard $S\ell_1QP$ method [21] if all constraints are considered, i.e., if $M = N$. In such a case, one finds that $\mathfrak{S}_M = \mathcal{I}$, (P) reduces to (RP), and Theorem 6 implies known convergence properties of a standard $S\ell_1QP$ method.*

We prove this theorem after proving a sequence of lemmas. A first observation is that the actual reduction and predicted reduction as defined in the ratio in Step 4 of Algorithm 1 are close for sufficiently small trial steps.

Lemma 8. *There exist $L_f > 0$ and $L_c > 0$ independent of $k \in \mathbb{N}$ such that*

$$\left| \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(d_k) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k + d_k) \right| \leq (\rho L_f + L_c + H_{\max})\|d_k\|^2 \quad \textit{for any } k.$$

Proof. Under Assumption 4, there exist Lipschitz constants $L_f > 0$ and $\tilde{L}_c > 0$ independent of $k \in \mathbb{N}$ such that, for any $k \in \mathbb{N}$,

$$\left|\rho\left(f(x_k) + \nabla f(x_k)^T d_k\right) + \tfrac{1}{2} d_k^T H_k d_k - \rho f(x_k + d_k)\right| \leq (\rho L_f + H_{\max})\|d_k\|^2$$
$$\text{and} \quad \left\|c(x_k) + \nabla c(x_k)^T d_k - c(x_k + d_k)\right\| \leq \tilde{L}_c \|d_k\|^2.$$

Because the mapping $w_{\mathcal{S}} \mapsto \|[w_{\mathcal{S}}]_+\|_1$ is Lipschitz continuous for any $\mathcal{S} \subseteq \mathcal{I}$ and the pointwise minimum of a set of Lipschitz continuous functions is Lipschitz continuous, the mapping $w \mapsto \min_{\mathcal{S}\in\mathfrak{S}_M} \|[w_{\mathcal{S}}]_+\|_1$ is Lipschitz continuous. Thus, there exists a constant $L_c > 0$ independent of $k \in \mathbb{N}$ such that, for any $k \in \mathbb{N}$,

$$\begin{aligned}
&\left| \left(\min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(d_k) \right) - \left(\min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k + d_k) \right) \right| \\
&= \left| \left(\rho(f(x_k) + \nabla f(x_k)^T d_k) + \tfrac{1}{2} d_k^T H_k d_k + \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \|[c_{\mathcal{S}}(x_k) + \nabla c_{\mathcal{S}}(x_k)^T d_k]_+\|_1 \right) \right. \\
&\quad \left. - \left(\rho f(x_k + d_k) + \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \|[c_{\mathcal{S}}(x_k + d_k)]_+\|_1 \right) \right| \leq (\rho L_f + L_c + H_{\max})\|d_k\|^2,
\end{aligned}$$

which is the desired conclusion. ∎

We now establish lower bounds for the reduction in the piecewise-quadratic model of the penalty functions corresponding to each trial step. Following standard trust region terminology, we quantify this reduction in terms of a *Cauchy point* from x_k corresponding to a model of $\phi_{\mathcal{S}}$ for each $\mathcal{S} \in \mathfrak{S}_M$. Specifically, we define the Cauchy point in iteration $k \in \mathbb{N}$ for a given scenario selection $\mathcal{S} \in \mathfrak{S}_M$ as $d^{\mathrm{C}}_{\mathcal{S},k} := \alpha^{\mathrm{C}}_{\mathcal{S},k} d^{\mathrm{L}}_{\mathcal{S},k}$, where $d^{\mathrm{L}}_{\mathcal{S},k} := d^{\mathrm{L}}_{\mathcal{S}}(x_k)$ (recall (19)) and, for some $\alpha \in (0,1)$ and $\eta \in (0,1)$ that are fixed for the remainder of this section, the value $\alpha^{\mathrm{C}}_{\mathcal{S},k} > 0$ is the largest element in the sequence $\{\alpha^j \min\{1, \delta_k/\|d^{\mathrm{L}}_{\mathcal{S},k}\|\}\}_{j\in\mathbb{N}}$ such that

$$\Delta q_{\mathcal{S},k}(\alpha^{\mathrm{C}}_{\mathcal{S},k} d^{\mathrm{L}}_{\mathcal{S},k}) \geq \eta \Delta l_{\mathcal{S},k}(\alpha^{\mathrm{C}}_{\mathcal{S},k} d^{\mathrm{L}}_{\mathcal{S},k}). \tag{21}$$

Our next lemma reveals that the reduction the Cauchy point yields in the piecewise-quadratic model of $\phi_{\mathcal{S}}$ is proportional to the criticality measure $\chi_{\mathcal{S},k} := \chi_{\mathcal{S}}(x_k)$.

Lemma 9. *For any $k \in \mathbb{N}$ and $\mathcal{S} \in \mathfrak{S}_M$, it follows that*

$$\Delta q_{\mathcal{S},k}(d^{\mathrm{C}}_{\mathcal{S},k}) \geq \eta\, \alpha^{\mathrm{C}}_{\mathcal{S},k}\, \chi_{\mathcal{S},k}. \tag{22}$$

Proof. For any $k \in \mathbb{N}$ and $\mathcal{S} \in \mathfrak{S}_M$, convexity of $l_{\mathcal{S},k}$ ensures that, corresponding to any step $d \in \mathbb{R}^n$ and stepsize $\tau \in [0,1]$, one has

$$\Delta l_{\mathcal{S},k}(\tau d) = l_{\mathcal{S},k}(0) - l_{\mathcal{S},k}(\tau d) \geq \tau(l_{\mathcal{S},k}(0) - l_{\mathcal{S},k}(d)) = \tau \Delta l_{\mathcal{S},k}(d). \tag{23}$$

Hence, by the definitions of $d^{\rm C}_{\mathcal{S},k}$ and $\alpha^{\rm C}_{\mathcal{S},k}$ (specifically, (21)), it follows that

$$\Delta q_{\mathcal{S},k}(d^{\rm C}_{\mathcal{S},k}) \geq \eta \Delta l_{\mathcal{S},k}(\alpha^{\rm C}_{\mathcal{S},k} d^{\rm L}_{\mathcal{S},k}) \geq \eta \alpha^{\rm C}_{\mathcal{S},k} \Delta l_{\mathcal{S},k}(d^{\rm L}_{\mathcal{S},k}) = \eta \alpha^{\rm C}_{\mathcal{S},k} \chi_{\mathcal{S},k},$$

as desired. ◻

To accompany Lemma 9, we now establish a lower bound on the stepsize defining any Cauchy step. To do this, note that Assumption 4 ensures that each model $l_{\mathcal{S},k}$ is Lipschitz continuous with Lipschitz constant independent of k and $\mathcal{S}$; in particular, under Assumption 4, there exists $L_l > 0$ such that, for any $k \in \mathbb{N}$ and $\mathcal{S} \in \mathfrak{S}_M$,

$$\chi_{\mathcal{S},k} = \Delta l_{\mathcal{S},k}(d^{\rm L}_{\mathcal{S},k}) = l_{\mathcal{S},k}(0) - l_{\mathcal{S},k}(d^{\rm L}_{\mathcal{S},k}) \leq L_l \|d^{\rm L}_{\mathcal{S},k}\|. \tag{24}$$

We now prove the following result.

LEMMA 10. *For any $k \in \mathbb{N}$ and $\mathcal{S} \in \mathfrak{S}_M$, the Cauchy stepsize satisfies*

$$\alpha^{\rm C}_{\mathcal{S},k} \geq \|d^{\rm C}_{\mathcal{S},k}\| \geq \min\left\{\frac{\chi_{\mathcal{S},k}}{L_l}, \delta_k, \frac{2(1-\eta)\alpha\chi_{\mathcal{S},k}}{H_{\max}}\right\}. \tag{25}$$

Proof. The first inequality follows since (19) involves $\|d^{\rm L}_{\mathcal{S},k}\| \leq 1$, from which it follows that $\|d^{\rm C}_{\mathcal{S},k}\| = \alpha^{\rm C}_{\mathcal{S},k}\|d^{\rm L}_{\mathcal{S},k}\| \leq \alpha^{\rm C}_{\mathcal{S},k}$. To establish the second inequality, consider two cases. First, if $\alpha^{\rm C}_{\mathcal{S},k} = \min\{1, \delta_k/\|d^{\rm L}_{\mathcal{S},k}\|\}$ (i.e., if the Cauchy condition (21) is satisfied by the stepsize corresponding to $j = 0$), then (24) yields

$$\|d^{\rm C}_{\mathcal{S},k}\| = \alpha^{\rm C}_{\mathcal{S},k}\|d^{\rm L}_{\mathcal{S},k}\| = \min\left\{1, \frac{\delta_k}{\|d^{\rm L}_{\mathcal{S},k}\|}\right\}\|d^{\rm L}_{\mathcal{S},k}\| \geq \min\left\{\frac{\chi_{\mathcal{S},k}}{L_l}, \delta_k\right\},$$

as desired. Second, if $\alpha^{\rm C}_{\mathcal{S},k} < \min\{1, \delta_k/\|d^{\rm L}_{\mathcal{S},k}\|\}$, then the Cauchy condition (21) must be violated when $\alpha^{\rm C}_{\mathcal{S},k}$ is replaced by $\alpha^{\rm C}_{\mathcal{S},k}/\alpha$, from which it follows that

$$\begin{aligned} &l_{\mathcal{S},k}(0) - l_{\mathcal{S},k}(\alpha^{\rm C}_{\mathcal{S},k} d^{\rm L}_{\mathcal{S},k}/\alpha) - \tfrac{1}{2}(\alpha^{\rm C}_{\mathcal{S},k}/\alpha)^2 (d^{\rm L}_{\mathcal{S},k})^T H_k d^{\rm L}_{\mathcal{S},k} \\ &= \Delta q_{\mathcal{S},k}(\alpha^{\rm C}_{\mathcal{S},k} d^{\rm L}_{\mathcal{S},k}/\alpha) < \eta \Delta l_{\mathcal{S},k}(\alpha^{\rm C}_{\mathcal{S},k} d^{\rm L}_{\mathcal{S},k}/\alpha) = \eta(l_{\mathcal{S},k}(0) - l_{\mathcal{S},k}(\alpha^{\rm C}_{\mathcal{S},k} d^{\rm L}_{\mathcal{S},k}/\alpha)). \end{aligned}$$

This inequality and (23) then reveal that, under Assumption 4,

$$\begin{aligned} &\tfrac{1}{2}(\alpha^{\rm C}_{\mathcal{S},k}/\alpha)^2 H_{\max}\|d^{\rm L}_{\mathcal{S},k}\|^2 \geq \tfrac{1}{2}(\alpha^{\rm C}_{\mathcal{S},k}/\alpha)^2 (d^{\rm L}_{\mathcal{S},k})^T H_k d^{\rm L}_{\mathcal{S},k} \\ &> (1-\eta)(l_{\mathcal{S},k}(0) - l_{\mathcal{S},k}(\alpha^{\rm C}_{\mathcal{S},k} d^{\rm L}_{\mathcal{S},k}/\alpha)) \geq (1-\eta)(\alpha^{\rm C}_{\mathcal{S},k}/\alpha)\Delta l_{\mathcal{S},k}(d^{\rm L}_{\mathcal{S},k}). \end{aligned}$$

Since (19) involves $\|d^{\rm L}_{\mathcal{S},k}\| \leq 1$, it now follows by (24) that

$$\|d^{\rm C}_{\mathcal{S},k}\| = \alpha^{\rm C}_{\mathcal{S},k}\|d^{\rm L}_{\mathcal{S},k}\| \geq \frac{2(1-\eta)\alpha\Delta l_{\mathcal{S},k}(d^{\rm L}_{\mathcal{S},k})}{H_{\max}} = \frac{2(1-\eta)\alpha\chi_{\mathcal{S},k}}{H_{\max}},$$

as desired. ◻

The next lemma reveals that, in a sufficiently small neighborhood of any point $x \in \mathbb{R}^n$ representing either an element or a limit point of the iterate sequence $\{x_k\}$, the reduction obtained by the Cauchy point defined with respect to $\tilde{\mathcal{S}} \in \mathfrak{S}_M(x;0)$ in the piecewise-quadratic model of ϕ is sufficiently large whenever the trust region radius and criticality measure for $\phi_{\tilde{\mathcal{S}}}$ are sufficiently large.

LEMMA 11. *Let $x \in \mathbb{R}^n$ be an element or a limit point of the sequence $\{x_k\}$, let $\tilde{\mathcal{S}} \in \mathfrak{S}_M(x;0)$, and consider $\delta_{\min} > 0$ and $\chi_{\min} > 0$ such that*

$$\delta_{\min} \leq \min\left\{\frac{\chi_{\min}}{L_l}, \frac{2(1-\eta)\alpha\chi_{\min}}{H_{\max}}\right\}. \tag{26}$$

Then, there exists a neighborhood $\mathcal{X}$ of x such that

$$\min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} q_{\mathcal{S},k}(0) - \min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} q_{\mathcal{S},k}(d^{\mathrm{C}}_{\tilde{\mathcal{S}},k}) \geq \tfrac{1}{2}\eta\chi_{\min}\min\left\{\frac{\chi_{\min}}{L_l}, \delta_k, \frac{2(1-\eta)\alpha\chi_{\min}}{H_{\max}}\right\}$$

whenever $\delta_k \geq \delta_{\min}$, $x_k \in \mathcal{X}$, and $\chi_{\tilde{\mathcal{S}},k} \geq \chi_{\min}$.

Proof. As in (8), it follows that $\min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} \phi_{\mathcal{S}}(x) = \phi_{\tilde{\mathcal{S}}}(x)$ since $\tilde{\mathcal{S}} \in \mathfrak{S}_M(x;0)$. Hence, from continuity of both $\min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} \phi_{\mathcal{S}}(\cdot)$ and $\phi_{\tilde{\mathcal{S}}}(\cdot)$, there exists a neighborhood $\mathcal{X}$ of x such that, whenever $x_k \in \mathcal{X}$,

$$\phi_{\tilde{\mathcal{S}}}(x_k) - \min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} \phi_{\mathcal{S}}(x_k) \leq \tfrac{1}{2}\eta\delta_{\min}\chi_{\min}. \tag{27}$$

Now, under the conditions of the lemma, with $\delta_k \geq \delta_{\min}$, $x_k \in \mathcal{X}$, and $\chi_{\tilde{\mathcal{S}},k} \geq \chi_{\min}$, it follows from Lemma 10 and (26) that $\alpha^{\mathrm{C}}_{\tilde{\mathcal{S}},k} \geq \delta_{\min}$. One then obtains from (27) that

$$\tfrac{1}{2}\eta\alpha^{\mathrm{C}}_{\tilde{\mathcal{S}},k}\chi_{\min} \geq \phi_{\tilde{\mathcal{S}}}(x_k) - \min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} \phi_{\mathcal{S}}(x_k).$$

Thus, along with Lemmas 9 and 10,

$$\begin{aligned}
&\min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} q_{\mathcal{S},k}(0) - \min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} q_{\mathcal{S},k}(d^{\mathrm{C}}_{\tilde{\mathcal{S}},k}) \\
\geq\ &\min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} q_{\mathcal{S},k}(0) - q_{\tilde{\mathcal{S}},k}(d^{\mathrm{C}}_{\tilde{\mathcal{S}},k}) \\
=\ &\min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} \phi_{\mathcal{S}}(x_k) - q_{\tilde{\mathcal{S}},k}(d^{\mathrm{C}}_{\tilde{\mathcal{S}},k}) \\
=\ &\min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} \phi_{\mathcal{S}}(x_k) - \phi_{\tilde{\mathcal{S}}}(x_k) + \phi_{\tilde{\mathcal{S}}}(x_k) - q_{\tilde{\mathcal{S}},k}(d^{\mathrm{C}}_{\tilde{\mathcal{S}},k}) \\
\geq\ &-\tfrac{1}{2}\eta\alpha^{\mathrm{C}}_{\tilde{\mathcal{S}},k}\chi_{\min} + \Delta q_{\tilde{\mathcal{S}},k}(d^{\mathrm{C}}_{\tilde{\mathcal{S}},k}) \\
\geq\ &\tfrac{1}{2}\eta\alpha^{\mathrm{C}}_{\tilde{\mathcal{S}},k}\chi_{\min} \\
\geq\ &\tfrac{1}{2}\eta\chi_{\min}\min\left\{\frac{\chi_{\min}}{L_l}, \delta_k, \frac{2(1-\eta)\alpha\chi_{\min}}{H_{\max}}\right\},
\end{aligned}$$

as desired. ∎

We now prove that around any non-stationary point representing an iterate or a limit point of the iterate sequence, there exists a neighborhood such that the trust region radius must be set sufficiently large. For obtaining this result, a key role is played by the trust region reset value $\delta_{\text{reset}} > 0$.

LEMMA 12. *Let $x \in \mathbb{R}^n$ be an element or a limit point of the sequence $\{x_k\}$ and suppose that $\chi(x) > 0$. In addition, let $\chi_{\min} := \frac{1}{2}\chi(x)$ and*

$$\delta_{\min} := \beta_2 \min\left\{\frac{\chi_{\min}}{L_l}, \delta_{\text{reset}}, \frac{2(1-\eta)\alpha\chi_{\min}}{H_{\max}}, \frac{(1-\mu)\eta\chi_{\min}}{2L}\right\}. \tag{28}$$

Then, there exists a neighborhood $\mathcal{X}$ of x such that $x_k \in \mathcal{X}$ implies $\delta_k \geq \delta_{\min}$.

Proof. From (20) and the definition of $\chi_{\min} > 0$, there exists $\mathcal{S} \in \mathfrak{S}_M(x;0)$ with $\chi_{\mathcal{S}}(x) = 2\chi_{\min}$. Thus, by continuity of $\chi_{\mathcal{S}}$ (recall Lemma 5), it follows that $\chi_{\mathcal{S}}(x_k) \geq \chi_{\min} > 0$ for all x_k in a sufficiently small neighborhood $\mathcal{X}_1$ of x. One also has by continuity of the constraint functions that there exists $\tilde{\epsilon} \in (0,\epsilon]$ such that $\mathfrak{S}_M(x_k;\tilde{\epsilon}) = \mathfrak{S}_M(x;0)$ for all x_k in a sufficiently small neighborhood $\mathcal{X}_2 \subseteq \mathcal{X}_1$ of x. Since $\tilde{\epsilon} \in (0,\epsilon]$, it follows from (9) that $\mathfrak{S}_{M,k} \supseteq \mathfrak{S}_M(x_k;\epsilon) \supseteq \mathfrak{S}_M(x_k;\tilde{\epsilon}) = \mathfrak{S}_M(x;0)$.

Since (28) implies (26), the result of Lemma 11 holds for some neighborhood $\mathcal{X} \subseteq \mathcal{X}_2$. In particular, noting that d_k is the global minimizer of (14), $\|d^{\mathrm{C}}_{\tilde{\mathcal{S}},k}\| \leq \delta_k$, and $q_{\mathcal{S},k}(0) = \phi_{\mathcal{S}}(x_k)$, it follows that, for any $\tilde{\mathcal{S}} \in \mathfrak{S}_M(x;0) \subseteq \mathfrak{S}_{M,k}$ with $\delta_k \geq \delta_{\min}$ and $x_k \in \mathcal{X}$,

$$
\begin{aligned}
&\min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(0) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(d_k) \\
&\overset{(8)}{\geq} \phi_{\tilde{\mathcal{S}}}(x_k) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(d^{\mathrm{C}}_{\tilde{\mathcal{S}},k}) \\
&\overset{(10)}{=} \min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} \phi_{\mathcal{S}}(x_k) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(d^{\mathrm{C}}_{\tilde{\mathcal{S}},k}) \\
&\geq \min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} q_{\mathcal{S},k}(0) - \min_{\mathcal{S}\in\mathfrak{S}_M(x;0)} q_{\mathcal{S},k}(d^{\mathrm{C}}_{\tilde{\mathcal{S}},k}) \\
&\geq \tfrac{1}{2}\eta\chi_{\min}\min\left\{\frac{\chi_{\min}}{L_l}, \delta_k, \frac{2(1-\eta)\alpha\chi_{\min}}{H_{\max}}\right\}. \qquad (29)
\end{aligned}
$$

The desired result can now be proved by contradiction. For this purpose, suppose that for some $\tilde{k} \in \mathbb{N}$ with $x_{\tilde{k}} \in \mathcal{X}$ the algorithm has $\delta_{\tilde{k}} < \delta_{\min}$. Since $\delta_0 \geq \delta_{\text{reset}}$ and the trust region radius is reset to at least δ_{reset} after each accepted trial step, it follows from the fact that $\delta_{\min} < \delta_{\text{reset}}$ that $\tilde{k} > 0$ and there must be some $k \in \{0,\ldots,\tilde{k}-1\}$ with $x_k = x_{\tilde{k}}$ and

$$\delta_{\min}/\beta_2 \geq \delta_k \geq \delta_{\min} \qquad (30)$$

where d_k was rejected. However, since (28) and (30) imply that

$$\delta_k \leq \min\left\{\frac{\chi_{\min}}{L_l}, \frac{2(1-\eta)\alpha\chi_{\min}}{H_{\max}}, \frac{(1-\mu)\eta\chi_{\min}}{2(\rho L_f + L_c + H_{\max})}\right\},$$

it follows with (29) and Lemma 8 that

$$
\begin{aligned}
1 - r_k &= 1 - \frac{\min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k + d_k)}{\min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(0) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(d_k)} \\
&\leq \frac{|\min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k + d_k) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(d_k)|}{\min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(0) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{\mathcal{S},k}(d_k)} \\
&\leq \frac{2(\rho L_f + L_c + H_{\max})\|d_k\|^2}{\eta\chi_{\min}\min\left\{\frac{\chi_{\min}}{L_l}, \delta_k, \frac{2(1-\eta)\alpha\chi_{\min}}{H_{\max}}\right\}} \\
&\leq \frac{2(\rho L_f + L_c + H_{\max})\delta_k^2}{\eta\chi_{\min}\delta_k} = \frac{2(\rho L_f + L_c + H_{\max})\delta_k}{\eta\chi_{\min}} \leq 1 - \mu,
\end{aligned}
$$

contradicting the assertion that d_k was rejected. Hence, no such $\tilde{k} \in \mathbb{N}$ with $x_{\tilde{k}} \in \mathcal{X}$ and $\delta_{\tilde{k}} < \delta_{\min}$ may exist, meaning that $x_k \in \mathcal{X}$ implies $\delta_k \geq \delta_{\min}$, as desired. ∎

We now prove that if the number of accepted steps is finite, then the algorithm must have arrived at a point with the stationarity measure for ϕ equal to zero.

LEMMA 13. *If* $\mathcal{K} := \{k \in \mathbb{N} : r_k \geq \mu\}$ *has* $|\mathcal{K}| < \infty$*, then* $x_k = x_*$ *for some* $x_* \in \mathbb{R}^n$ *for all sufficiently large* $k \in \mathbb{N}$ *where* x_* *is stationary for* ϕ *in that* $\chi(x_*) = 0$.

Proof. If the iteration index set $\mathcal{K}$ is finite, then the iterate update (16) ensures that $x_k = x_*$ for all $k \geq k_*$ for some $x_* \in \mathbb{R}^n$ and $k_* \in \mathbb{N}$. Hence, $\chi(x_k) = \chi(x_*)$ for all $k \geq k_*$. If $\chi(x_*) = 0$, then the desired result holds. Otherwise, if $\chi(x_*) > 0$, then Lemma 12 implies the existence of $\delta_{\min} > 0$ such that $\delta_k \geq \delta_{\min}$ for all $k \geq k_*$. However, this contradicts the fact that $|\mathcal{K}| < \infty$ and (17) ensure $\{\delta_k\} \searrow 0$. ∎

We are now ready to prove our global convergence theorem.

Proof (of Theorem 6). If $\mathcal{K} := \{k \in \mathbb{N} : r_k \geq \mu\}$ has $|\mathcal{K}| < \infty$, then Lemma 13 implies that $\chi(x_k) = 0$ for some finite $k \in \mathbb{N}$, which is represented by outcome (i). Thus, for the remainder of the proof, suppose that $|\mathcal{K}| = \infty$. For the purpose of deriving a contradiction, suppose that there exists a limit point x_* of the iterate sequence $\{x_k\}$ that has $\chi(x_*) > 0$. Let $\{x_{k_i}\}_{i\in\mathbb{N}}$ be an infinite subsequence of $\{x_k\}$ that converges to x_*. Without loss of generality, it can be assumed that $k_i \in \mathcal{K}$ for all $i \in \mathbb{N}$, i.e., that all steps from the elements of $\{x_{k_i}\}$ are successful in that $x_{k_i+1} = x_{k_i} + d_{k_i}$ for all $i \in \mathbb{N}$. In addition, along the lines of Lemma 12, let $\chi_{\min} := \frac{1}{2}\chi(x_*)$ and $\delta_{\min}$ be defined as in (28). Since $\{x_{k_i}\}$ converges to x_*, it can be assumed without loss of generality that $x_{k_i} \in \mathcal{X}$ for all $i \in \mathbb{N}$ with $\mathcal{X}$ defined as in Lemma 12, from which it follows that $\delta_{k_i} \geq \delta_{\min}$.

As in the proof of Lemma 12 (recall (29)), it follows by (28) that

$$\begin{aligned}\min_{\mathcal{S}\in\mathfrak{S}_{M,k_i}} q_{\mathcal{S},k_i}(0) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k_i}} q_{\mathcal{S},k_i}(d_{k_i}) &\geq \tfrac{1}{2}\eta\chi_{\min}\min\left\{\frac{\chi_{\min}}{L_l}, \delta_{k_i}, \frac{2(1-\eta)\alpha\chi_{\min}}{H_{\max}}\right\} \\ &\geq \tfrac{1}{2}\eta\chi_{\min}\delta_{\min}. \end{aligned} \tag{31}$$

On the other hand, using (8) and (10), one has for all $k \in \mathbb{N}$ that

$$\begin{aligned} &\phi(x_k) = \min_{\mathcal{S}\in\mathfrak{S}_M} \phi_{\mathcal{S}}(x_k) = \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k) \\ \text{and}\quad &\phi(x_k + d_k) = \min_{\mathcal{S}\in\mathfrak{S}_M} \phi_{\mathcal{S}}(x_k + d_k) \leq \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k + d_k). \end{aligned}$$

Consequently, for any $k \in \mathcal{K}$ it follows that $r_k \geq \mu$ and, hence,

$$\begin{aligned}\phi(x_k) - \phi(x_{k+1}) &\geq \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} \phi_{\mathcal{S}}(x_k + d_k) \\ &\geq \mu\left(\min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{k,\mathcal{S}}(0) - \min_{\mathcal{S}\in\mathfrak{S}_{M,k}} q_{k,\mathcal{S}}(d_k)\right) \geq 0.\end{aligned}$$

For the subsequence $\{x_{k_i}\}$ we can strenghthen this, using (31), to

$$\phi(x_{k_i}) - \phi(x_{k_i+1}) \geq \tfrac{1}{2}\eta\mu\chi_{\min}\delta_{\min}.$$

Summing this inequality over all $k \in \mathbb{N}$ and noting that $\phi(x_k) - \phi(x_{k+1}) = 0$ for each unsuccessful $k \in \mathbb{N} \setminus \mathcal{K}$ implies that $\{\phi(x_k)\} \searrow -\infty$. However, this contradicts Assumption 4. Consequently, a limit point x_* with $\chi(x_*) > 0$ cannot exist. ∎

4.2. Avoiding "poor" local minimizers. The analysis presented in the previous subsection concerns convergence to a local minimizer (or at least a stationary point) of the penalty function ϕ, which under nice conditions (recall Lemmas 2 and 3) corresponds to a local minimizer (or stationary point) of the cardinality-constrained problem (P). It is important to note, however, that such convergence is of little meaning if the algorithm is likely to get attracted to a "poor" local minimizer. In our setting of cardinality-constrained optimization, this is of particular concern since the feasible region of such a problem is often very jagged; see, e.g., Figure 1 for the two-dimensional example studied in §5.2. As a consequence, these problems have many local minimizers (in fact, increasingly more if N grows, such as in SAA approximations of chance-constrained problems), many of which can be considered "poor" in the sense that there are better local minima in a small neighborhood.

Fortunately, there is good reason to believe that our method will not get trapped at a local minimizer that is particularly poor. The important feature that guarantees this desirable behavior is that our method employs subproblem (18), which is itself a cardinality-constrained problem involving local Taylor models of the original problem functions, meaning that it inherits the jagged structure of problem. This allows our method to, at least locally, be aware of better local minimizers that are nearby.

To make this claim more precise, suppose that the algorithm has reached an iterate $x_k \in \mathbb{R}^n$ such that subproblem (14) with $\mathfrak{S}_{M,k} = \mathfrak{S}_M$ yields $d_k = 0$ for any sufficiently small $\delta_k > 0$. In addition, suppose that there exists another local minimizer $x \in \mathbb{R}^n$ of the penalty function such that $\phi(x_k) > \phi(x)$. Let the distance between the minimizers be $\delta := \|x - x_k\|$. By the same reasoning as in the proof of Lemma 8, it follows that

$$\begin{aligned} & \left| \min_{\mathcal{S}\in\mathfrak{S}_M} q_{\mathcal{S},k}(x - x_k) - \min_{\mathcal{S}\in\mathfrak{S}_M} \phi_{\mathcal{S}}(x) \right| \leq (\rho L_f + L_c + H_{\max})\delta^2 \\ \implies \quad & \phi(x) \geq \min_{\mathcal{S}\in\mathfrak{S}_M} q_{\mathcal{S},k}(x - x_k) - (\rho L_f + L_c + H_{\max})\delta^2. \end{aligned} \tag{579}$$

In other words, the penalty function value at x cannot be less than that predicted by the quadratic model at x_k minus a term involving only the penalty parameter, Lipschitz constants of f and $\{c_i\}$, the upper bound on the Hessian, and the distance between x_k and x. If this term is small in that $\rho L_f + L_c + H_{\max}$ and/or δ is small, then a step from x_k to x yields a predicted reduction that is sufficiently close to the actual reduction $\phi(x_k) - \phi(x)$, meaning that our algorithm would accept such a step.

There are few observations to make as a result of this discussion. First, we have argued that our algorithm will not get "stuck" due to a wrong scenario selection. This can be seen in the fact that any discrepancy between the actual and prediction reduction in our penalty function is independent of the particular $\mathcal{S} \in \mathfrak{S}_M$ that minimizes the constraint violation. Second, we have shown that if a better local minimizer is nearby, then our algorithm would compute and accept a step toward such a point unless the problem functions are highly nonlinear (i.e., if L_f and/or L_c is large) and/or if $H_{\max}$ is large. Finally, if the problem functions are linear, $H_{\max} = 0$, and our algorithm considers a sufficiently large trust region radius, then our algorithm will compute a global minimizer of the penalty function.

5. Numerical Experiments. We now describe a prototype implementation of Algorithm 1 that we have written in `Matlab B2015b`, pose two interesting test problems, and present the results of numerical experiments when the implementation is tasked to solve various instances of the problems. The primary purpose of this investigation is to demonstrate that, while Algorithm 1 has only the convergence guarantees of Theorem 6, in practice it has the potential to generate high-quality solutions to realistic problems. We also illustrate that the algorithm is flexible and allows for trade-offs between speed and quality of the solution through the tuning of algorithmic parameters. All computations were executed on a Ubuntu 12.04 Linux workstation with 256GB RAM and Intel Xeon CPU with 20 cores, running at 3.10GHz.

5.1. Implementation. Our implementation solves general instances of the form

$$\text{(Q)} \qquad \min_{x\in\mathbb{R}^n} \ f(x) \quad \text{s.t.} \quad |\{i \in \mathcal{I} : c_i(x) \le 0\}| \ge M, \quad \bar{c}(x) \le 0,$$

where the objective and cardinality constraint quantities are defined as in problem (P) while $\bar{c} : \mathbb{R}^n \to \mathbb{R}^{\overline{m}}$ represents an additional inequality constraint function. These additional constraints are handled through the addition of a standard ℓ_1-norm penalty term of the form $\|[\bar{c}(x)]_+\|_1$ in the definition of $\phi_{\mathcal{S}}$ in (5). The local models (12) and all related quantities are modified accordingly. (For ease of exposition, we continue to use the notation and labels of all of these quantities with these modifications presumed.) One can verify that an analysis as in §4 still applies.

Each iteration requires the solution of (14), which, using an ℓ_∞-norm trust region, can equivalently be rewritten in terms of a cardinality-constrained problem with linear functions in the constraints and a quadratic objective function. Specifically, with $e_m \in \mathbb{R}^m$, $e_{\overline{m}} \in \mathbb{R}^{\overline{m}}$, and $e_n \in \mathbb{R}^n$ representing vectors of ones, $\mathcal{N}_k$ representing indices of constraints to be enforced, and $\mathcal{C}_k$ representing indices of constraints *that might be* enforced (such that $\mathfrak{S}_{M,k} = \mathcal{N}_k \cup \mathcal{C}_k$), our implementation considers

$$\text{(32)} \qquad \begin{aligned} &\min_{\substack{d\in\mathbb{R}^n\\ t\in\mathbb{R}^{\overline{m}}\\ s_i\in\mathbb{R}^m\\ \forall i\in\mathcal{N}_k\cup\mathcal{C}_k}} \ \rho\nabla f(x)^Td + \tfrac{1}{2}d^TH_kd + \sum_{i\in\mathcal{C}_k\cup\mathcal{N}_k} e_m^Ts_i + e_{\overline{m}}^Tt \\ &\text{s.t.} \ \begin{cases} |\{i\in\mathcal{C}_k : c_i(x_k) + \nabla c_i(x_k)^Td \le s_i\}| \ge M - |\mathcal{N}_k|, \\ c_i(x_k) + \nabla c_i(x_k)^Td \le s_i \quad \text{for all} \quad i\in\mathcal{N}_k, \\ \bar{c}(x_k) + \nabla\bar{c}(x_k)^Td \le t, \\ -\delta_ke_n \le d \le \delta_ke_n, \\ s_i \ge 0 \quad \text{for all} \quad i\in\mathcal{N}_k\cup\mathcal{C}_k, \\ t \ge 0. \end{cases} \end{aligned}$$

Further details about our method for choosing $\mathcal{N}_k$ and $\mathcal{C}_k$ are given at the end of this subsection. To solve subproblem (32), the cardinality constraint is replaced using a "big-M" approach, leading to the mixed-integer constraint set

$$\text{(33)} \qquad \begin{aligned} c_i(x_k) + \nabla c_i(x_k)^Td \le s_i + \overline{M}(1-z_i) \quad &\text{for all} \quad i\in\mathcal{C}_k, \\ \sum_{i\in\mathcal{C}_k} z_i = M - |\mathcal{N}_k|, \quad & \\ z_i \in \{0,1\} \quad &\text{for all} \quad i\in\mathcal{C}_k. \end{aligned}$$

Here, for each $i \in \mathcal{C}_k$, the binary variable z_i indicates whether the constraint corresponding to the index i is enforced and the parameter $\overline{M}$ is chosen large enough so that the constraint is inactive at the optimal solution of (32) whenever $z_i = 0$. We stress that this "big-M" approach, while valid, likely leads to vastly inferior performance compared to a specialized branch-and-cut decomposition algorithm for solving these types of problems, such as the one proposed by Luedtke [34]. However, since adopting such a specialized approach would require a significant programming effort beyond the scope of this paper, our implementation simply employs the "big-M" approach above, invoking `Cplex` (version 12.6.2) to solve the resulting subproblem.

For the quadratic objective term in (32), our implementation uses

$$H_k = \rho \nabla^2 f(x_k) + \sum_{i=1}^{N} \sum_{j=1}^{m} [\lambda_k^c]_{ij} \nabla^2 c_{ij}(x_k) + \sum_{j=1}^{\overline{m}} [\lambda_k^{\bar{c}}]_j \nabla^2 \bar{c}_j(x_k) + (\zeta + 10^{-8}) I.$$

The quantities involved in this expression require some explanation. Corresponding to the constraint functions c_{ij} (i.e., the jth element of the vector function c_i) and $\bar{c}_j$ (i.e., the jth element of the vector function $\bar{c}$), the quantities $[\lambda_k^c]_{ij}$ and $[\lambda_k^{\bar{c}}]_j$ can be interpreted as Lagrange multipliers. Thus, the first three terms defining H_k can be viewed as the Hessian of the Lagrangian corresponding to problem (Q) involving all elements of the cardinality constraints. In all iterations after the first one—the choice of the initial values is described below—the values for these Lagrange multipliers are set to the optimal multipliers for the constraints in (32)–(33) when this subproblem is solved at the previous iterate with all binary variables fixed at their optimal values. For any $i \notin \mathfrak{S}_{M,k}$ and for any $i \in \mathcal{C}_k$ such that the optimal binary variable is $z_i^* = 0$, the corresponding multipliers are set to zero. As for the scalar $\zeta > 0$, it is chosen to ensure that H_k is positive definite so that the objective of (32)–(33) is convex. Specifically, its value is chosen using "Algorithm IC (Inertia Correction)" in [47] with the parameters $\bar{\delta}_w^0 = 10^{-4}$, $\bar{\delta}_w^{\min} = 10^{-12}$, $\bar{\delta}_w^{\max} = 10^{10}$, $\bar{\kappa}_w^+ = \kappa_w^+ = 8$, and $\kappa_w^- = 1/3$. This method first checks if H_k is positive definite with $\zeta = 0$. If this is the case, then the choice $\zeta = 0$ is used; otherwise, increasing positive values of ζ are tried until H_k is positive definite. In our implementation, to determine whether a trial matrix is positive definite, the smallest eigenvalue of H_k is computed using Matlab's built-in `eigs` function. The small shift of the eigenvalues by 10^{-8} (regardless of the computed value for ζ) is included as a regularization term since it made the numerical performance of the `Cplex` solver more stable in our experiments.

The values for the parameters in Algorithm 1 that we employ in our implementation are $\rho = 0.01$, $\epsilon = 10^{-3}$, $\delta_0 = 1000$, $\mu = 10^{-8}$, and $\beta_1 = \beta_2 = 10^{-8}$. The remaining parameters are given in subsequent subsections for each experiment. The algorithm terminates when $\|d_k\|_\infty \leq 10^{-6}$. We also note that, to account for numerical inaccuracies in the subproblem solutions, our implementation relaxes the ratio test in (15) by adding 10^{-8} to both the numerator and denominator.

Significant tuning of the `Cplex` solver was required to achieve reliably performance, both when solving instances of the mixed-integer QP (32)–(33) and the related QP for computing the Lagrange multipliers in the definition of H_k. Feasibility and optimality tolerances were decreased to 10^{-8} and the `emphasis.numerical` option was set to 1, asking `Cplex` to emphasize "extreme numerical caution". When solving the QP for computing Lagrange multipliers, we set the `solutiontarget` option to 1, thereby telling `Cplex` that we are seeking an optimal solution to a convex QP.

When solving (32)–(33), we used the dual simplex solver with integrality gap tolerance set to 10^{-7} and absolute optimality gap tolerance set to 10^{-8}. Furthermore, we specified that the interior point QP solver is used for solving the root node in the branch-and-bound methods for the MIQP. This was necessary because otherwise `Cplex` claimed in some instances that the root node was infeasible (which, in theory, is not possible due to the presence of the slack variables). We used `Cplex`'s `MipStart` feature to provide the zero-step (setting $d = 0$, and choosing the remaining variables to minimize the objective in (32)–(33)) as a feasible incumbent. We imposed a time limit of 300 seconds for each subproblem. If the time limit was exceeded when solving an instance of (32)–(33), then we took the final incumbent solution as the step d_k. In some cases, this incumbent did not lead to a reduction in the model, and the run was reported as a failure (even though this says nothing about the performance of Algorithm 1, except that the subproblem was numerically difficult to solve).

Finally, we recall the critical role in Algorithm 1 played by the set $\mathfrak{S}_{M,k}$, which must satisfy $\mathfrak{S}_{M,k} \supseteq \mathfrak{S}_M(x_k;\epsilon)$. As expected, in our implementation and experiments, the manner in which this set is chosen offers a trade-off between speed and quality of the ultimate solution found. If one were to set $\mathfrak{S}_{M,k} = \mathfrak{S}_M(x_k;\epsilon)$ for a relatively small ϵ (as would typically describe our default value of $\epsilon = 10^{-3}$), then one would often find the subproblem solved relatively quickly due to the smaller numbers of elements in the cardinality constraints (and, hence, smaller number of binary variables) in the generated instances of (32)–(33). On the other hand, if one were to choose $\mathfrak{S}_{M,k} = \mathcal{I}$ (corresponding to $\epsilon = \infty$), then one might find that the algorithm attains a better quality solution at the expense of a *much* higher computing time. Since one of the goals of our experiments is to illustrate this trade-off, our implementation employs a fractional scenario selection parameter $\gamma \in [0,1]$ in the choice of $\mathcal{C}_k$ which, in contrast to ϵ, is not dependent on the scale of the constraint function c. Specifically, with $M^- := \max\{1, M - \lceil \gamma N \rceil\}$ and $M^+ := \min\{N, M + \lceil \gamma N \rceil\}$ and recalling the ordering scheme defined by the function v in (2), our implementation chooses

$$\mathcal{C}_k \leftarrow \left\{ i_{c(x_k),M^-}, \ldots, i_{c(x_k),M^+} \right\} \cup \mathcal{C}(x_k,\epsilon) \ \text{ and } \ \mathcal{N}_k \leftarrow \mathcal{N}(x_k,\epsilon) \setminus \mathcal{C}_k.$$

In this manner, to follow our convergence analysis in §4, the set $\mathcal{C}_k$ always includes the set $\mathcal{C}(x_k,\epsilon)$, but it might also include indices corresponding to additional constraints whose violations are above and/or below the critical value $v_M(x_k)$. We explore the effects of various values of γ in our numerical experiments.

5.2. A Simple Nonconvex Example. This experiment explores how likely the algorithm gets stuck at a "poor" local minimizer (recall §4.2) using an example with $n = 2$, $f(x) = x_2$, $c_i(x) = 0.25x_1^4 - 1/3x_1^3 - x_1^2 + 0.2x_1 - 19.5 + \xi_{i,1}x_1 + \xi_{i,1}\xi_{i,0}$, $N = 5{,}000$, and $\alpha = 0.05$. The random values $\{\xi_{i,1}\}_{i\in\mathcal{I}}$ were drawn uniformly from $[-3,3]$ while $\{\xi_{i,0}\}_{i\in\mathcal{I}}$ were drawn uniformly from $[-12,12]$. Figure 1a depicts the feasible region of (P). By inspection, we see that this function has a "true" local minimizer around $x_1 = -1$, in the sense that this is the best local minimizer in a large neighborhood. The global minimizer lies around $x_1 = 2$.

This problem was solved with $\overline{M} = 20$ and $\delta_{\text{reset}} = 1$, using different starting points, different values of the penalty parameter ρ, and different values of the fractional parameter γ. The initial multipliers for the computation of H_0 were set to zero. The starting point was computed by choosing $[x_0]_1$ (the first component of x_0), then determining $[x_0]_2$ so that x_0 lies on the boundary of the feasible region.

Table 1 gives the initial points along with their corresponding function values. The starting point with $[x^{\text{bad}}]_1 = 0.08524989$ is included as one from which some poor performance was observed. This point is a "poor" local minimizer with a large objective value, away from the two "true" local minimizers (see Figure 1d). We include this starting point to explore whether the method can escape such a local solution.

$[x_0]_1$	-1.500	-1.000	-0.500	0.000	0.500	1.000	1.500	2.000	0.085
$f(x_0)$	2.299	1.823	2.127	2.357	2.562	1.657	0.880	0.548	2.310

Table 1: Values of the initial points and the corresponding objective functions.

Table 2 shows the objective function values for the points returned by the algorithm. Function values that are within 1% of one of the "true" minima are highlighted in bold face, and those within 5% are given in italics. The markers in Figures 1a–1d depict the locations of all the points returned by Algorithm 1. We see that each such point appears to be a local minimizer.

	γ							
$[x_0]_1$	0.001	0.002	0.005	0.010	0.050	0.100	0.200	1.000
-1.5	*1.776*	*1.776*	**1.754**	**1.754**	**1.754**	**1.754**	**1.754**	**1.754**
-1.0	**1.754**	**1.760**	**1.754**	**1.754**	**1.754**	**1.754**	**1.754**	**1.754**
-0.5	2.049	1.951	1.814	**1.754**	**1.754**	**1.754**	**1.754**	**1.754**
0.0	2.289	2.289	1.814	**1.754**	**1.754**	**1.754**	**1.754**	**1.754**
0.5	2.382	2.382	**0.460**	**0.460**	**0.460**	**0.460**	**0.460**	**0.460**
1.0	0.912	**0.460**	**0.460**	**0.460**	**0.463**	**0.463**	**0.463**	**0.463**
1.5	0.579	**0.460**	**0.460**	**0.460**	**0.460**	**0.460**	**0.460**	**0.460**
2.0	**0.463**	**0.463**	**0.460**	**0.460**	**0.463**	**0.463**	**0.463**	**0.463**
x^{bad}	2.310	2.289	1.814	**1.754**	**1.754**	**1.754**	**1.754**	**1.754**

Results for $\rho = 1$

	γ							
$[x_0]_1$	0.001	0.002	0.005	0.010	0.050	0.100	0.200	1.000
-1.5	*1.784*	*1.784*	*1.784*	**1.754**	*1.776*	*1.776*	*1.784*	*1.784*
-1.0	1.813	1.813	1.813	1.813	1.813	1.813	1.813	1.813
-0.5	2.049	1.951	1.951	1.951	1.814	1.814	1.814	1.814
0.0	2.289	2.289	1.951	1.951	*1.776*	*1.784*	*1.784*	*1.784*
0.5	2.382	2.382	0.579	0.579	0.624	**0.460**	**0.460**	**0.460**
1.0	0.912	0.579	0.624	0.624	0.579	0.579	0.579	0.579
1.5	0.624	0.579	**0.460**	*0.473*	**0.460**	**0.460**	**0.460**	**0.460**
2.0	**0.463**	**0.463**	**0.463**	**0.463**	**0.463**	**0.463**	**0.463**	**0.463**
x^{bad}	2.310	2.289	1.951	1.951	*1.776*	*1.784*	*1.784*	*1.784*

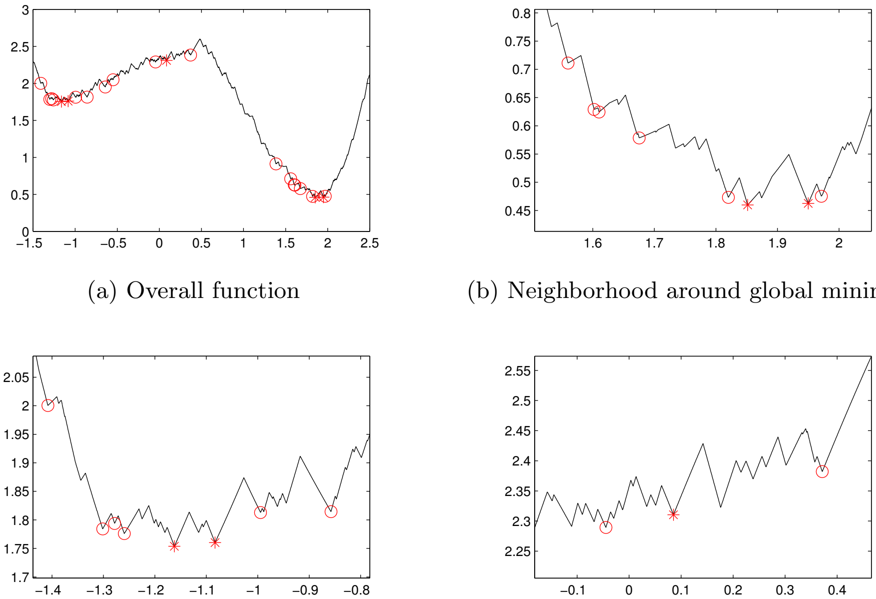
Results for $\rho = 0.1$

	γ							
$[x_0]_1$	0.001	0.002	0.005	0.010	0.050	0.100	0.200	1.000
-1.5	2.001	*1.784*	*1.784*	*1.784*	*1.784*	*1.784*	*1.784*	*1.784*
-1.0	1.813	1.813	1.813	1.813	1.813	1.813	1.813	1.813
-0.5	2.049	1.951	1.951	1.951	1.951	1.951	1.951	1.951
0.0	2.289	2.289	1.951	1.951	*1.784*	*1.794*	*1.794*	*1.794*
0.5	2.382	2.382	0.912	0.624	0.624	0.624	0.624	0.624
1.0	0.912	0.912	0.579	0.711	0.624	0.624	0.624	0.624
1.5	0.629	0.624	0.629	0.629	0.579	**0.460**	**0.460**	**0.460**
2.0	*0.475*	*0.475*	*0.475*	*0.475*	*0.475*	*0.475*	*0.475*	*0.475*
x^{bad}	2.310	2.289	2.289	1.951	*1.784*	*1.784*	*1.784*	*1.784*

Results for $\rho = 0.01$

Table 2: Objective function values of the final iterates returned by Algorithm 1 for a simple two-dimensional example problem, using different values for the penalty parameter ρ and scenario selection parameter γ. The most-left column shows the starting point component $[x_0]_2$ together with the corresponding objective values $f(x_0)$.

The penalty parameter ρ balances the influence of the objective function and the constraint violation. The subproblem model q_k is defined by linearizations of



(a) Overall function

(b) Neighborhood around global minimum

(c) Neighborhood around local minimum

(d) Neighborhood around bad starting point

Fig. 1: Final points returned for the noisy nonconvex test function. The markers indicate the different points returned by the algorithm in the different runs from Table 2. The stars are the locations of the points with objective values within 1% of one of the "true" minima in 1a–1c. In 1d, the star denotes the point x^{bad}.

the constraints, meaning that it approximates the constraint violation well near an iterate. However, in order to make progress towards a local minimizer that is not in the immediate neighborhood of the current iterate, it might be necessary to take a larger step. In that case, q_k might overestimate the constraint violation and not permit a sufficiently large trial step, unless a decrease in the objective function outweighs the increase in the constraint violation. A larger value of the penalty parameter gives more weight to the progress in the objective function, and makes it more likely that a larger step can be taken. We see this confirmed in Table 2, where better final objective values are obtained when ρ is large. The algorithm is then able to escape from an inferior local minimum more often. Of course, the penalty parameter must still be small enough to ensure that the limit point is feasible; see Lemma 2.

Another important factor that determines the solution quality is the number of scenarios that are considered in subproblem (14). We vary this by choosing different values of the scenario selection parameter γ. When γ is small, only scenarios are considered that are relevant to describe the feasible region locally in a small neighborhood around the current iterate. For instances, in our example, $\gamma = 0.001$ leads to a scenario selection $\mathfrak{S}_{M,k}$ of size 21 chosen out of 5000. As a consequence, the subproblem might again overestimate the constraint violation (recall that the model objective in (14) is defined as the minimum over the scenario selection $\mathfrak{S}_{M\,k}$) and not permit trial steps that are sufficiently large to escape a poor local minimizer. Indeed, we note in

Table 2 that inferior final objective values are obtained for small values of γ. In fact, the method is not able to escape the local minimizer x^{bad} when $\gamma = 0.001$. On the other hand, in the extreme case when $\gamma = 1$, the method typically achieves the best results for a given penalty parameter. However, the computational effort required to solve the subproblem can be significantly higher for large values of γ, and in practice a trade-off has to be made between solution quality and solution time. We also explore this trade-off in our larger example in §5.3.

The trust-region reset value δ_{reset} is another parameter that determines the final solution quality. Consider a situation where a new point is accepted with a small trust-region radius δ_k after a sequence of rejected trial points due to disagreement in (15) of the actual reduction and the predicted reduction. After this, the trust-region is increased to δ_{reset}. However, if this reset value is too small to reach a better local minimizer in the model objective in (14), the method might be restricted to stay close to an inferior local minimizer. Again, in practice a trade-off has to be made between solution quality and computation time, since a large value of δ_{reset} might require more rejected trial steps until an acceptable trial point is computed. A large value of δ_{reset} also makes the solution of the subproblem more difficult because of the increased feasible region. While the results above were found with $\delta_{\text{reset}} = 1$, we also ran the method with $\delta_{\text{reset}} \in \{0.1, 10\}$. This resulted in different final objective values for a few cases compared to Table 2, but none of the choices for δ_{reset} was clearly superior.

Overall, for this example, Algorithm 1 finds "true" local solutions in most cases when the penalty parameter is large and when γ is at least 0.05. This corresponds to 501 critical scenarios, chosen out of a total of 5,000 scenarios. These choices even allow the method to escape from the inferior local minimizer x^{bad}.

5.3. A Cash Flow Problem. Our second test problem is a nonlinear and nonconvex variation of the cash flow problem considered in [17] given by

$$\begin{aligned}
\max_{\substack{x\in\mathbb{R}^T\\ z\in\mathbb{R}^{T+1}}} \quad & z_{T+1} - \mathbb{E}\left[\sum_{t=1}^{T} L_t\right] \\
\text{s.t.} \quad & \left|\left\{k = 1,\dots,N : z_{t+1} \geq \sum_{\hat{t}=1}^{t} l_{\hat{t},k} \;\text{ for all }\; t = 1,\dots,T\right\}\right| \geq M, \\
& z_t = z_{t-1} - \sum_{j\in J: s_j = t} x_j + \sum_{j\in J: s_j + d_j = t} I_j(x_j)^{\frac{d_j}{2}} x_j \;\text{ for all }\; t = 2,\dots,T+1, \\
& z_1 = z^{\text{init}}, \quad z \geq 0, \quad x \geq 0.
\end{aligned}$$

Here, the task is to decide how much money x_j should be invested in each investment option $j \in J$ (in our experiments, $|J| = 75$) over 10 years, divided into $T = 20$ time periods. Option j can be bought at start time $s_j \in \{1,\dots,T\}$ and is sold after d_j periods. When sold, interest has been earned with an annual interest rate of $I_j(x_j)$. In contrast to [17], the interest rate depends on the amount of money invested and increases as the investment becomes larger, making the chance constraints nonlinear and nonconvex. We assume that the decision about how much to invest in the different options must be made upfront (with no recourse). In each time period t, an unknown random liability L_t is incurred and has to be paid from money that is not currently invested. The realizations of L_t are denoted $l_{t,k}$. The variables z_t keep track of the amount of cash available in period t, and we need to ensure that there is always enough cash on hand to pay off the liabilities $l_{t,k}$. For the objective, we maximize the expected amount of cash available at the end of the 10 years. We note that this problem resembles other important stochastic optimal control applications such as inventory and battery management where one seeks to balance profit with the risk of depleting stored resources and not being able to satisfy demands. The chance-constrained approach allows us to explore such trade-offs.

The initial budget is $z^{\text{init}} = 10$. The interest rate for investment $j \in J$ is given by

$$I_j(x_j) = \underline{I}_j + (\overline{I}_j - \underline{I}_j) \frac{\log(1 + \psi_j x_j)}{\log(1 + \psi_j z^{\text{init}})},$$

where $\underline{I}_j$ is the initial interest rate for very small investments, and $\underline{I}_j$ is the interest rate that would be earned if all of the initial cash z^{init} is invested in that option. This function is monotonically increasing and concave, with diminishing interest increase as the investment increases. Varying $\psi_j > 0$ changes the curvature of I_j; with this, we can explore the effects of nonconvexities on algorithm performance.

We report the performance of the proposed method averaged over five instances with randomly generated data. For generating each instance, the parameter d_j was drawn uniformly from $\{1, \dots, 10\}$ and s_j was drawn uniformly from $\{1, \dots, T - d_j\}$. The values $\underline{I}_j$ and $\overline{I}_j$ were drawn uniformly from $[0.01, 0.05]$ and $(\underline{I}_j + 0.005, \underline{I}_j + 0.015)$, respectively. The parameter ψ_j was chosen as 10^{-p_j}, where p_j was drawn uniformly from $[2, 6]$. The unknown liabilities L_j followed a normal distribution with mean z^{init}/T and variance $0.2 z^{\text{init}}/T$, i.e., 20% of the mean. With this, $\mathbb{E}[\sum_{t=1}^{T} L_t] = z^{\text{init}}$.

We chose $\delta_{\text{reset}} = 0.1$, $\overline{M} = 15$, and $\rho = 0.1$. We verified that the penalty parameter ρ was small enough so that, in each successful outcome, the infeasibility of the solution $\tilde{x}^*$ returned from our algorithm, i.e., $\langle\!\langle c(\tilde{x}^*) \rangle\!\rangle_M$, was at most 10^{-6}.

The initial point x_0 was set as the optimal solution of the "robust" counterpart of (P), i.e., problem (RP). This point was computed by the `Ipopt` solver [47] in negligible time. We highlight that the ability to initialize the search using the robust solution implicitly allows our algorithm to quickly identify subsets of constraints that can be relaxed to improve the objective. Furthermore, the optimal multipliers obtained by `Ipopt` are taken as the initial values of the multipliers $[\lambda_0^c]_{ij}$ and $[\lambda_0^{\bar{c}}]_j$.

Table 3 details the results of our numerical experiment, where we varied the number of scenarios N between 100 and 5,000 with $\alpha = 0.05$. The relevance of the size of the critical set $\mathcal{C}_k$ was assessed using values of the scenario selection parameter γ between 0.001 and 1. We count an instance as solved ("OK") if the algorithm terminated without an error. Unsuccessful outcomes were sometimes observed for one of the five randomly generated instances when $N \geq 2000$, where for some values of γ the QP for computing the multipliers could not be solved. We emphasize that these outcomes do not represent a failure of the proposed algorithm, but rather are a consequence of the inefficient manner in which we are solving the subproblems (recall the discussion surrounding (33)) in this preliminary implementation of the method.

Table 3 reports geometric averages over the successfully solved instances ("OK") for the number of iterations, the number the critical scenarios $\mathcal{C}_k$ per iteration, the total number of changes—from one iteration to the next—in the set of scenarios $\mathcal{W}_k = \mathcal{C}_k \cup \mathcal{N}_k$ considered in the subproblem (summed over all iterations after the first), and the wall clock time.

The quality of the solution returned by the algorithm is assessed using arithmetic averages of the final objective function values $f(x^{\text{rob}})$ and the relative improvement (a percentage) of the objective function compared to the robust solution x^{rob}, computed as $100 \frac{f(x^{\text{rob}}) - f(x_*)}{f(x^{\text{rob}})}$. To ensure consistency when comparing the influence of the choice of γ on the relative improvement, the averages were taken only over all instances that were solved for all values of γ. The numbers in parentheses indicate the number of instances over which the relative improvement was averaged.

Interestingly, the improvements achieved by the proposed algorithm over the robust formulation become more apparent as the number of scenarios increases. This is because the robust solution becomes increasingly conservative as N grows, which is reflected in smaller values of $f(x^{\text{rob}})$ (recall that this is a maximization problem). For reference, the (arithmetic) averages of the robust optima are given in Table 4.

N	100	200	500	1000	2000	3000	5000
$f(x^{\text{rob}})$	2.2456	2.2270	2.1852	2.1412	2.0844	2.0678	2.0441

Table 4: Average optimal objective value for robust formulation.

The main purpose of these experiments is the exploration of the trade-off between solution quality and computational effort. As in the experiments of §5.2, we observe that better objective function values are obtained when more scenarios are considered in the subproblem. The column "$|\mathcal{C}_k|$" shows how many discrete variables are in the MIQP formulation of (32). For this problem, we see that the best results are obtained if we choose γ as small as 0.05, so that about only 10% of all scenarios are critical in the subproblem. Even if we reduce γ further, the outcome is often very good.

Looking at the number of changes in the scenario set $\mathcal{W}_k$ considered in the subproblem from one iteration to the next, we see that this is not a trivial problem. Specifically for the large cases, the total number of changes is several hundreds. On the other hand, for $N = 100$, the subproblem in the first iteration finds the scenario selection that is considered relevant in the end. We also observed that the method seemed to converged at a superlinear rate once the final $\mathcal{W}_k$ was found, unless a suboptimal incumbent was returned by the subproblem solver due to the time limit.

Regarding the computational effort, we note that the number of iterations is moderate, indicating that the initial point provided by the robust solution is useful. Clearly, the computation times increase as the size of the subproblems grows. The reason that they are almost constant with $N = 5{,}000$ for varying γ is that, in most iterations, the 5 minutes time limit for the `Cplex` solver was exceeded, and the method continued with the best incumbent as the trial step in that iteration. As we can see, the final solution quality might even decline as γ increases to 1, because these incumbents are inferior solutions of the subproblem (14), and the method might then stall early.

We stress that our implementation is a proof-of-concept that has not attempted to reduce the overall computing time. Given that the MIQP subproblem can become quite large (the MIQP formulations for the last line in the table has 5,000 discrete variables, 100,213 continuous variables, and 100,097 constraints), the current times might be considered somewhat reasonable. However, the computation times can be expected to be significantly smaller in an implementation that uses the branch-and-cut method by Luedtke [34] to solve the subproblems.

6. Summary and Extensions. We presented an algorithm for solving nonlinear optimization problems with cardinality-constraints that arise from sample average approximations of chance constraints. Our analysis showed that, under standard assumptions, the method produces stationary points for an exact penalty function that coincide with stationary points for the optimization problem. A potential drawback of the proposed method is that it might converge to poor local minimizers resulting from the jaggedness of the feasible region caused by the SAA discretization of the chance constraint. Our numerical experiments demonstrate that this is only of a minor concern, even if only a fraction of all scenarios are considered in each subproblem.

In the paper, we assumed that the value of the penalty parameter ρ is fixed throughout the optimization. In practice, a suitable value is often not known *a priori*, and an update mechanism is necessary to adjust its value. As with standard nonlinear optimization problems, the penalty parameter has to be sufficiently small to ensure convergence to feasible points. On the other hand, as seen in §5.2, too small a value might result in convergence to inferior local solutions. This is in contrast to standard nonlinear optimization, where a small value usually affects only the convergence speed, not the quality of the solution.

The numerical experiments in the paper were carried out with a proof-of-concept Matlab implementation. A practical implementation requires that the subproblems are solved by a much more efficient method than the generic `Cplex` MIQP solver applied to a big-M formulation (33). Here, the decomposition algorithm by Luedtke [34] is a very promising option, since it is tailored specifically to chance constraints. It models the chance constraint for a given scenario i as the general implication

$$z_i \implies x \in P_i, \tag{34}$$

where z_i is a binary indicator variable that is one when scenario i is enforced, and P_i is a polyheron. Second-stage (cost-free) recourse actions can be expressed naturally in this manner, where the hidden recourse variables are part of the definition of P_i. The method proposed in [34] has shown tremendous improvement over a big-M formulation (like (33)) in this context. We expect that Algorithm 1 can reap these benefits for the solution of nonlinear chance-constrained problems with resource actions as well. Here, conceptually, the hidden second-stage variables are included explicitly in the optimization variables x in (P), but subproblem (32) can be posed using the implicit formulation (34) without recourse variables, thereby making it amenable to the decomposition method in [34].

REFERENCES

[1] L. Andrieu, R. Henrion, and W. Römisch, *A model for dynamic chance constraints in hydro power reservoir management*, European Journal of Operational Research, 207 (2010), pp. 579–589.

[2] J. Barrera, T. Homem-de Mello, E. Moreno, B. K. Pagnoncelli, and G. Canessa, *Chance-constrained problems and rare events: an importance sampling approach*, Mathematical Programming, 157 (2016), pp. 153–189, doi:10.1007/s10107-015-0942-x, http://dx.doi.org/10.1007/s10107-015-0942-x.

[3] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming*, John Wiley & Sons, Inc., 2005, doi:10.1002/0471787779, http://dx.doi.org/10.1002/0471787779.

[4] A. Ben-Tal and A. Nemirovski, *Robust optimization–methodology and applications*, Mathematical Programming, 92 (2002), pp. 453–480.

[5] P. Beraldi and A. Ruszczyński, *The probabilistic set-covering problem*, Operations Research, 50 (2002), pp. 956–967.

[6] D. Bertsimas, D. B. Brown, and C. Caramanis, *Theory and applications of robust optimization*, SIAM Review, 53 (2011), pp. 464–501.

[7] D. Bienstock, M. Chertkov, and S. Harnett, *Chance-constrained optimal power flow: Risk-aware network control under uncertainty*, SIAM Review, 56 (2014), pp. 461–495.

[8] J. R. Birge and F. Louveaux, *Introduction to stochastic programming*, Springer Science & Business Media, 2011.

[9] P. Bonami and M. A. Lejeune, *An exact solution approach for portfolio optimization problems under stochastic and integer constraints*, Operations Research, 57 (2009), pp. 650–670.

[10] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz, *On the convergence of successive linear-quadratic programming algorithms*, SIAM Journal on Optimization, 16 (2005), pp. 471–489, doi:10.1137/S1052623403426532, http://dx.doi.org/10.1137/S1052623403426532, arXiv:http://dx.doi.org/10.1137/S1052623403426532.

[11] G. C. Calafiore and L. El Ghaoui, *On distributionally robust chance-constrained linear programs*, Journal of Optimization Theory and Applications, 130 (2006), pp. 1–22.

[12] A. Carniato and E. Camponogara, *Integrated coal-mining operations planning: Modeling and case study*, International Journal of Coal Preparation and Utilization, 31 (2011), pp. 299–334.

[13] A. Charnes and W. W. Cooper, *Deterministic equivalents for optimizing and satisficing under chance constraints*, Operations Research, 11 (1963), pp. 18–39.

[14] A. Charnes, W. W. Cooper, and G. H. Symonds, *Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil*, Management Science, 4 (1958), pp. 235–263.

[15] W. Chen, M. Sim, J. Sun, and C.-P. Teo, *From CVaR to uncertainty set: Implications in joint chance-constrained optimization*, Operations Research, 58 (2010), pp. 470–485.

[16] N. de Oliveira and L. T. Biegler, *Constraint handing and stability properties of model-predictive control*, AIChE journal, 40 (1994), pp. 1138–1155.

[17] D. Dentcheva, B. Lai, and A. Ruszczyński, *Dual methods for probabilistic optimization problems*, Mathematical Methods of Operations Research, 60 (2004), pp. 331–346.

[18] D. Dentcheva, A. Prékopa, and A. Ruszczyński, *Concavity and efficient points of discrete distributions in probabilistic programming*, Mathematical Programming, 89 (2000), pp. 55–77.

[19] G. Dorfleitner and S. Utz, *Safety first portfolio choice based on financial and sustainability returns*, European Journal of Operational Research, 221 (2012), pp. 155–164.

[20] J. Dupačová, A. Gaivoronski, Z. Kos, and T. Szantai, *Stochastic programming in water management: A case study and a comparison of solution techniques*, European Journal of Operational Research, 52 (1991), pp. 28–44.

[21] R. Fletcher, *An ℓ_1 penalty method for nonlinear constraints*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, PA, USA, 1985, pp. 26–40.

[22] R. Fletcher, *Practical Methods of Optimization*, John Wiley and Sons, New York, USA, second ed., 1987.

[23] S. P. Han and O. L. Mangasarian, *Exact penalty functions in nonlinear programming*, Mathematical Programming, 17 (1979), pp. 251–269, doi:10.1007/BF01588250, http://dx.doi.org/10.1007/BF01588250.

[24] R. Henrion, P. Li, A. Möller, M. C. Steinbach, M. Wendt, and G. Wozny, *Stochastic optimization for operating chemical processes under uncertainty*, in Online Optimization of Large Scale Systems, M. Grötschel, S. O. Krumke, and J. Rambau, eds., Springer Berlin Heidelberg, 2001, pp. 457–478, doi:10.1007/978-3-662-04331-8_24, http://dx.doi.org/10.1007/978-3-662-04331-8_24.

[25] T. Homem-de Mello and G. Bayraksan, *Monte carlo sampling-based methods for stochastic optimization*, Surveys in Operations Research and Management Science, 19 (2014), pp. 56–85.

[26] L. J. Hong, Y. Yang, and L. Zhang, *Sequential convex approximations to joint chance constrained programs: A Monte Carlo approach*, Operations Research, 59 (2011), pp. 617–630.

[27] T. Kim and S. J. Wright, *An s\ ell _1lp-active set approach for feasibility restoration in power systems*, Optimization and Engineering, (2014), pp. 1–35.

[28] A. J. Kleywegt, A. Shapiro, and T. Homem-de-Mello, *The sample average approximation method for stochastic discrete optimization*, SIAM Journal on Optimization, 12 (2001), pp. 479–502.

[29] S. Küçükyavuz, *On mixing sets arising in chance-constrained programming*, Mathematical Programming, 132 (2012), pp. 31–56.

[30] M. A. Lejeune, *Pattern definition of the p-efficiency concept*, Annals of Operations Research, 200 (2012), pp. 23–36.

[31] M. A. Lejeune and A. Ruszczyński, *An efficient trajectory method for probabilistic production-inventory-distribution problems*, Operations Research, 55 (2007), pp. 378–394.

[32] W. Li, Y. Zhang, A. So, and M. Win, *Slow adaptive OFDMA systems through chance constrained programming*, IEEE Transactions on Signal Processing, 58 (2010), pp. 3858–3869.

[33] Y. Liu, H. Guo, F. Zhou, X. Qin, K. Huang, and Y. Yu, *Inexact chance-constrained linear programming model for optimal water pollution management at the watershed scale*, Journal of Water Resources Planning and Management, 134 (2008), pp. 347–356.

[34] J. Luedtke, *A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support*, Mathematical Programming, 146 (2014), pp. 219–244.

[35] J. Luedtke and S. Ahmed, *A sample approximation approach for optimization with probabilistic constraints*, SIAM Journal on Optimization, 19 (2008), pp. 674–699.

[36] J. Luedtke, S. Ahmed, and G. L. Nemhauser, *An integer programming approach for linear programs with probabilistic constraints*, Mathematical Programming, 122 (2010), pp. 247–272.

[37] M. R. Murr and A. Prékopa, *Solution of a product substitution problem using stochastic programming*, in Probabilistic Constrained Optimization, S. Uryasev, ed., Springer, 2000, pp. 252–271.

[38] A. Nemirovski and A. Shapiro, *Convex approximations of chance constrained programs*, SIAM Journal on Optimization, 17 (2006), pp. 969–996.

[39] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, NY, USA, second ed., 2006.

[40] J. Pintér, *Deterministic approximations of probability inequalities*, Zeitschrift für Operations Research, 33 (1989), pp. 219–239.

[41] R. T. Rockafellar and S. Uryasev, *Optimization of conditional value-at-risk*, Journal of Risk, 2 (2000), pp. 21–42.

[42] R. T. Rockafellar and S. Uryasev, *Conditional value-at-risk for general loss distributions*, Journal of banking & finance, 26 (2002), pp. 1443–1471.

[43] F. Shan, L. Zhang, and X. Xiao, *A smoothing function approach to joint chance-constrained programs*, Journal of Optimization Theory and Applications, (2014), pp. 1–19.

[44] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, 2009.

[45] Q.-T. Thieu and H.-Y. Hsieh, *Use of chance-constrained programming for solving the opportunistic spectrum sharing problem under Rayleigh fading*, in Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, IEEE, 2013, pp. 1792–1797.

[46] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati, *Joint chance constrained programming for hydro reservoir management*, Optimization and Engineering, 15 (2014), pp. 509–531.

[47] A. Wächter and L. T. Biegler, *On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming*, Mathematical Programming, 106 (2006), pp. 25–57.

[48] Y. Yuan, *Conditions for convergence of trust region algorithms for nonsmooth optimization*, Mathematical Programming, 31 (1985), pp. 220–228, doi:10.1007/BF02591750, http://dx.doi.org/10.1007/BF02591750.

N	γ	OK	iter	$\|\mathcal{C}_k\|$	$\Delta(\mathcal{W}_k)$	$f(x^*)$	rel. impr.	time sec
100	0.001	5	7.00	6.31	0.00	2.3306	3.61 (5)	3.43
100	0.002	5	7.00	6.31	0.00	2.3306	3.61 (5)	2.83
100	0.005	5	7.00	6.31	0.00	2.3306	3.61 (5)	2.96
100	0.010	5	7.00	6.31	0.00	2.3306	3.61 (5)	2.87
100	0.050	5	6.75	11.61	0.00	2.3312	3.64 (5)	4.15
100	0.200	5	6.75	26.00	0.00	2.3312	3.64 (5)	5.82
100	1.000	5	6.75	100.00	0.00	2.3312	3.64 (5)	6.97
200	0.001	5	7.42	7.91	0.00	2.3245	4.16 (5)	5.52
200	0.002	5	7.42	7.91	0.00	2.3245	4.16 (5)	5.33
200	0.005	5	7.42	7.91	0.00	2.3245	4.16 (5)	5.37
200	0.010	5	7.42	9.27	0.00	2.3280	4.32 (5)	6.30
200	0.050	5	7.42	21.00	2.46	2.3288	4.39 (5)	10.57
200	0.200	5	7.42	51.00	2.95	2.3288	4.39 (5)	14.66
200	1.000	5	7.42	200.00	2.95	2.3288	4.39 (5)	22.19
500	0.001	5	9.84	9.40	6.14	2.3103	5.72 (5)	25.95
500	0.002	5	9.84	9.40	6.14	2.3103	5.72 (5)	27.59
500	0.005	5	8.19	11.84	8.68	2.3156	5.96 (5)	37.45
500	0.010	5	7.69	14.27	9.58	2.3155	5.95 (5)	39.33
500	0.050	5	7.04	51.00	13.06	2.3171	6.03 (5)	52.96
500	0.200	5	7.04	126.00	13.66	2.3171	6.03 (5)	101.41
500	1.000	5	7.04	500.00	15.52	2.3171	6.03 (5)	260.08
1000	0.001	5	15.98	10.90	15.15	2.3023	7.40 (5)	117.13
1000	0.002	5	13.71	11.96	19.17	2.3059	7.57 (5)	130.13
1000	0.005	5	12.29	15.22	23.72	2.3092	7.73 (5)	147.30
1000	0.010	5	11.04	22.26	25.96	2.3104	7.79 (5)	176.23
1000	0.050	5	8.12	101.00	34.29	2.3126	7.91 (5)	313.17
1000	0.200	5	7.43	251.00	49.23	2.3126	7.91 (5)	411.45
1000	1.000	5	7.43	1000.00	57.14	2.3126	7.91 (5)	819.24
2000	0.001	5	21.94	13.84	43.67	2.2872	9.73 (4)	652.19
2000	0.002	4	17.22	15.88	51.58	2.2887	9.81 (4)	550.45
2000	0.005	4	12.85	23.01	56.61	2.2907	9.90 (4)	513.20
2000	0.010	4	13.39	41.00	67.10	2.2920	9.97 (4)	1013.33
2000	0.050	4	9.90	201.00	88.99	2.2946	10.10 (4)	1099.41
2000	0.200	4	9.90	501.00	113.26	2.2946	10.10 (4)	2208.06
2000	1.000	5	14.33	2000.00	129.18	2.2945	10.09 (4)	3519.53
3000	0.001	5	26.00	16.38	74.36	2.2895	10.74 (4)	1207.26
3000	0.002	5	19.86	19.24	79.64	2.2906	10.80 (4)	1093.91
3000	0.005	5	16.72	31.42	92.40	2.2921	10.86 (4)	1777.77
3000	0.010	4	14.56	61.00	111.11	2.2935	10.93 (4)	1736.09
3000	0.050	4	10.75	301.00	137.77	2.2945	10.98 (4)	2389.05
3000	0.200	5	9.14	751.00	183.12	2.2942	10.97 (4)	2251.02
3000	1.000	4	11.97	3000.00	227.11	2.2927	10.89 (4)	3124.97
5000	0.001	5	28.20	20.74	137.24	2.2866	11.92 (4)	2965.06
5000	0.002	5	21.49	25.84	144.26	2.2873	11.95 (4)	3469.57
5000	0.005	4	18.58	51.00	172.91	2.2897	12.06 (4)	4570.27
5000	0.010	4	15.90	101.00	192.78	2.2917	12.17 (4)	3954.34
5000	0.050	4	14.37	501.00	230.20	2.2917	12.16 (4)	3853.37
5000	0.200	4	11.66	1251.00	344.47	2.2912	12.14 (4)	3014.36
5000	1.000	4	8.99	5000.00	410.05	2.2676	10.89 (4)	2211.95

Table 3: Numerical results for cash flow problem with $\alpha = 0.05$.