Numer Algor (2014) 65:1–22
DOI 10.1007/s11075-012-9692-5

ORIGINAL PAPER

A feasible SQP-GS algorithm for nonconvex, nonsmooth constrained optimization

Chun-ming Tang · Shuai Liu · Jin-bao Jian · Jian-ling Li

Received: 17 September 2011 / Accepted: 28 December 2012 / Published online: 22 February 2013

Abstract The gradient sampling (GS) algorithm for minimizing a nonconvex, nonsmooth function was proposed by Burke et al. (SIAM J Optim 15:751–779, 2005), whose most interesting feature is the use of randomly sampled gradients instead of subgradients. In this paper, combining the GS technique with the sequential quadratic programming (SQP) method, we present a feasible SQP-GS algorithm that extends the GS algorithm to nonconvex, nonsmooth constrained optimization. The proposed algorithm generates a sequence of feasible iterates, and guarantees that the objective function is monotonically decreasing. Global convergence is proved in the sense that, with probability one, every cluster point of the iterative sequence is stationary for the improvement function. Finally, some preliminary numerical results show that the proposed algorithm is effective.

Project supported by the National Natural Science Foundation (No. 11271086), Tianyuan Fund for Mathematics (No. 11126341), Innovation Group of Talents Highland of Guangxi Higher School, and Science Foundation of Guangxi Education Department (No. 201102ZD002) of China.

C.-m. Tang · S. Liu · J.-l. Li
College of Mathematics and Information Science, Guangxi University, Nanning 530004,
People's Republic of China

J.-b. Jian (✉)
College of Mathematics and Information Science, Yulin Normal University, Yulin 537000,
People's Republic of China
e-mail: jianjb@gxu.edu.cn
URL: http://jians.gxu.edu.cn

J.-b. Jian
College of Mathematics and Information Science, Guangxi University, Nanning, Guangxi, 530004,
People's Republic of China

Keywords Nonsmooth optimization · Sequential quadratic programming · Gradient sampling · Feasible algorithm

Mathematics Subject Classifications (2010) 90C30 · 49M30

1 Introduction

We consider the following constrained optimization problem

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \ f(x) \\ &\text{s.t.} \ \ c_i(x) \le 0, \ i \in I \overset{\triangle}{=} \{1, \cdots, m\}, \end{aligned} \tag{1}$$

where the functions $f, \ c_i \ (i \in I) : \ \mathbb{R}^n \to \mathbb{R}$ are locally Lipschitz and continuously differentiable on open dense subsets of $\mathbb{R}^n$. Denote the feasible set for problem (1) by

$$\mathcal{F} = \left\{ x \in \mathbb{R}^n : \ c_i(x) \le 0, \ i \in I \right\}.$$

Problems of this type arise in many important applications, see [1–3] and references therein.

The *gradient sampling* (GS) algorithm was proposed by Burke, Lewis and Overton [2, 3] for solving the unconstrained version (i.e. $I = \emptyset$) of problem (1). For the current iterate x^k and a sample radius $\epsilon^k \ge 0$, the GS algorithm first samples independently and uniformly a finite set of points from $\mathbb{B}_{\epsilon^k}(x^k) = \{x : \ \|x - x^k\| \le \epsilon^k\}$ with 2-norm. If f is continuously differentiable at all the sample points, then a set G_k that is a convex hull of the gradients corresponding to these points is generated to approximate the Clarke ϵ^k-subdifferential (see [4]) of f at x^k. The minimum-norm element g^k of G_k is computed by solving a quadratic programming (QP) subproblem, and therefore an approximate ϵ^k-steepest descent direction d^k is obtained by setting $d^k = -g^k/\|g^k\|$. Finally, an Armijo line search along d^k produces the new iterate x^{k+1} (perturb if necessary to stay in the set $\mathcal{D}^f$ where f is differentiable). With probability one, the GS algorithm is globally convergent. The most interesting feature of the GS algorithm is the use of randomly sampled gradients instead of subgradients used in traditional nonsmooth methods, such as subgradient methods and bundle methods (see e.g. [5, 6]). Kiwiel [7] further improved the convergence results of [3], and presented several practical modifications. Kiwiel [8] proposed creatively a nonderivative version of the GS algorithm, such that the gradients can be approximated by f-values only. Most recently, Curtis and Que [9] proposed an adaptive GS algorithm that requires significantly fewer gradient evaluations. Thanks to the nice theoretical properties, the GS algorithm has been applied successfully to solve many practical problems, see e.g. [3, 10, 11].

The success of the GS algorithm in unconstrained optimization greatly stimulated researchers to extend it to solve constrained problem (1). It is well known that *sequential quadratic programming* (SQP) is among the most efficient methods for solving smooth constrained optimization [12, 13]. So combining SQP method with the GS technique, Curtis and Overton [1] proposed a penalty function based SQP-GS algorithm for solving problem (1). They used the ℓ_1 penalty function to balance between the decrease of the objective function and the feasibility of the constraints

$$\phi_\rho(x) = \rho f(x) + \sum_{i \in I} \max\{c_i(x), 0\}, \tag{2}$$

where $\rho > 0$ is a penalty parameter. The search direction is obtained by solving a QP subproblem (see (7) below), and penalty function (2) serves as a merit function to generate the next iterate. The penalty SQP-GS algorithm of [1] is globally convergent in the sense that, with probability one, every cluster point of the iterative sequence is stationary for $\phi_\rho(\cdot)$. It is also notable to mention that it is robust in practice and the efficient software SLQP-GS can be downloaded freely from the website of Curtis.

As far as we are aware, the penalty SQP-GS algorithm [1] is the first and the only constrained GS algorithm proposed to date. However, as is well known, the difficulty of the penalty function based methods is that estimating a suitable value of the penalty parameter is sometimes a delicate task, since excessive large or small parameters will bring difficulties either theoretically or numerically. In fact, the earlier version of the algorithm in [1] only used a fixed value (suitably small) of the penalty parameter ρ, while some practical updating rules without theoretical convergence guarantees were presented in numerical implementation. (As this paper was being revised, we learned that a penalty updating strategy with theoretical analysis was included in the latest version of the algorithm [1]).

As an alternative of the penalty SQP-GS algorithm [1], in this paper we present a feasible SQP-GS algorithm without a penalty for solving problem (1). The proposed algorithm, starting from a feasible point, generates a sequence of feasible iterates, and guarantees that the objective function is monotonically decreasing. It is worth pointing out that feasible algorithms are an important class of algorithms for solving both smooth and nonsmooth optimization problems (see [5, 14–19]), since they not only have nice theoretical properties, but also are widely applicable and robust in practice. Compared to the penalty function (2), we make use of the *improvement function*

$$\psi_x(y) = \max\{f(y) - f(x); C(y)\}, \tag{3}$$

where $x, y \in \mathbb{R}^n$ and $C(y) = \max\{c_i(y), i \in I\}$, which is one of the most effective tools to handle constraints in this context (see [5, 6] and Lemma 1 below) and plays a significant role in this paper. With the help of this function, we prove that the proposed algorithm is globally convergent in the sense that, with probability one, every cluster point of the iterative sequence is stationary for the improvement function.

We close this section by pointing out that a drawback of the class of feasible algorithms is the need for feasibility of the starting point. However, this drawback may be overcome relatively easily by some suitable approaches, such as the Phase I-Phase II methods [5, 23] and the strongly sub-feasible direction methods [20–22]. That is, we can extend the feasible SQP-GS algorithm to accept infeasible starting points through these methods, but we defer such extensions to future work. In the next section, we recall some basic facts that are relevant to our algorithm. In Section 3, we present the details of the proposed algorithm and discuss its properties. In Section 4, the global convergence is proved. Preliminary numerical results and conclusion are given in Sections 5 and 6, respectively. The symbol $\|\cdot\|$ stands for the Euclidean vector norm, and $\nabla f(x)$ denotes the gradient of $f(x)$ at x.

2 Preliminaries

In this section, we first describe a basic assumption and some notations used in the remainder of the paper. Then we give an overview of the unconstrained GS method [3] and the penalty SQP-GS method [1].

As in [1], a basic assumption about the problem data is required.

Assumption 1 *The functions f, c_i, $i \in I$ are locally Lipschitz on $\mathbb{R}^n$ and continuously differentiable on the open dense subsets $\mathcal{D}^f$, $\mathcal{D}^{c_i}$, $i \in I$, respectively, of $\mathbb{R}^n$.*

We recall some basic facts that are relevant to the GS algorithm [3]. The Clarke subdifferential [24] of f at any point x is defined by

$$\bar{\partial} f(x) = \mathrm{co}\left\{\lim_j \nabla f(y^j):\ y^j \to x,\ y^j \in \mathcal{D}^f\right\}, \tag{4}$$

where "co" denotes the convex hull. The Clarke ϵ-subdifferential [4] is given by

$$\bar{\partial}_\epsilon f(x) = \mathrm{co}\{\bar{\partial} f(y):\ y \in \mathbb{B}_\epsilon(x)\},$$

where $\mathbb{B}_\epsilon(x) = \{y:\ \|y - x\| \le \epsilon\}$ is the ball centered at x with radius $\epsilon \ge 0$. The Clarke ϵ-subdifferential $\bar{\partial}_\epsilon f(x)$ can be approximated by the set (see [3])

$$G_\epsilon(x) = \mathrm{cl}\ \mathrm{co}\left\{\nabla f(y):\ y \in \mathbb{B}_\epsilon(x) \cap \mathcal{D}^f\right\},$$

since $G_\epsilon(x) \subset \bar{\partial}_\epsilon f(x)$ and $\bar{\partial}_{\epsilon_1} f(x) \subset G_{\epsilon_2}(x)$ for $0 \le \epsilon_1 < \epsilon_2$, where "cl" denotes the closure. The Clarke subdifferential of f at a point x is given by (see e.g. [3])

$$\bar{\partial} f(x) = \bigcap_{\epsilon > 0} G_\epsilon(x). \tag{5}$$

For a given iterate x^k, the basic idea behind the GS algorithm [3] is that the set $G_{\epsilon^k}(x^k)$ (and thus $\bar{\partial}_{\epsilon^k} f(x^k)$) is approximated by the convex hull of a bundle of randomly sampled gradients, denoted by G_k as follows

$$G_k = \mathrm{co}\{\nabla f(x^k), \nabla f(x^{k1}), \cdots, \nabla f(x^{kp})\},$$

where $\{x^{ki}\}_{i=1}^p$ with $p \ge n+1$ are sampled independently and uniformly from $\mathbb{B}_{\epsilon^k}(x^k)$. The minimum-norm element g^k of G_k is computed by solving the subproblem

$$\min_{g \in G_k} \ \tfrac{1}{2}\|g\|^2 \tag{6}$$

and then the approximate ϵ^k-steepest descent direction d^k is obtained by setting $d^k = -g^k/\|g^k\|$. An Armijo type line search is performed at x^k along d^k to obtain a new iterate.

By analyzing the dual of the subproblem (6), Curtis and Overton [1] deduced that a search direction d^k possessing similar properties can also be obtained as the solution to

$$\begin{array}{ll} \displaystyle\min_{d \in \mathbb{R}^n,\ z \in \mathbb{R}} & z + \tfrac{1}{2} d^T H^k d \\ \text{s.t.} & f(x^k) + \nabla f(x)^T d \le z,\ \forall\, x \in \{x^k, x^{k1}, \cdots, x^{kp}\}, \end{array}$$

where H^k is a symmetric, sufficiently positive definite, and bounded matrix.

By appending constraints and combining the idea of penalty SQP method, Curtis and Overton [1] then considered obtaining such directions d^k by solving the QP subproblem

$$\begin{array}{ll} \min\limits_{d,z,r} & \rho z + \sum\limits_{i\in I} r_i + \frac{1}{2} d^T H^k d \\ \text{s.t.} & f(x^k) + \nabla f(x)^T d \le z, \ \ \forall\, x \in \mathcal{B}_f^k \\ & c_i(x^k) + \nabla c_i(x)^T d \le r_i, \ \ \forall\, x \in \mathcal{B}_{c_i}^k, \ r_i \ge 0, \ i \in I, \end{array} \tag{7}$$

where

$$\mathcal{B}_f^k = \left\{x^k, x_f^{k1}, \cdots, x_f^{kp}\right\} \ \text{ and } \ \mathcal{B}_{c_i}^k = \left\{x^k, x_{c_i}^{k1}, \cdots, x_{c_i}^{kp}\right\}, \ i \in I, \tag{8}$$

are sets of independent and identically distributed random points sampled uniformly from $\mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}^f$ and $\mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}^{c_i}$, $i \in I$, respectively. The ℓ_1 penalty function (2) is then used to generate a new iterate.

Remark 1

(i) The convergence analysis ([1, Lemma 3.8]) requires that different sample points must be used for functions f and c_i, $i \in I$, respectively.

(ii) The nominal point x^k is always included in the sample set, which guarantees that the computed direction is always (rather than almost always) appropriate. More precisely, it ensures that d^k is a descent direction of function $\phi_\rho(\cdot)$ at x^k ([1, Lemma 3.7]). In this paper, including x^k implies that d^k is a feasible descent direction of problem (1) at x^k, see Lemma 2 (iii) below.

3 The feasible SQP-GS algorithm

In this section, we present a feasible version of the SQP-GS algorithm. We first show that the improvement function (3) is well defined and is suitable to serve as an alternative of the penalty function (2).

Lemma 1 *Suppose that $\bar{x}$ is a local solution of problem* (1). *Then*

(i) *there exists a neighbourhood $\mathcal{N}(\bar{x})$, such that*

$$\min\{\psi_{\bar{x}}(y) : \ y \in \mathcal{N}(\bar{x})\} = \psi_{\bar{x}}(\bar{x}) = 0;$$

(ii) $0 \in \bar{\partial}\psi_{\bar{x}}(\bar{x})$.

Proof If $\bar{x}$ is a local solution of problem (1), then there exists a neighbourhood $\mathcal{N}(\bar{x})$ of $\bar{x}$, such that

$$f(y) \ge f(\bar{x}), \ \forall\, y \in \mathcal{N}(\bar{x}) \cap \mathcal{F},$$

which implies

$$\begin{aligned} \max_{y\in\mathcal{N}(\bar{x})} \{f(y) - f(\bar{x}); \ C(y)\} &\ge \max_{y\in\mathcal{N}(\bar{x})\cap\mathcal{F}} \{f(y) - f(\bar{x}); \ C(y)\} \\ &\ge 0 = \psi_{\bar{x}}(\bar{x}), \end{aligned}$$

so (i) holds.

From part (i), the claim (ii) is a well-known result (see e.g. [5, Lemma1.2.14]). □

Remark 2 If f and c_i $(i \in I)$ are convex and the Slater constraint qualification is satisfied for problem (1), i.e., there exists a vector $\tilde{x} \in \mathbb{R}^n$ satisfying $c_i(\tilde{x}) < 0,\ \forall\, i \in I$, then the above results are sufficient, i.e., the following statements are equivalent (see [5, Lemma 1.2.16]):

(i) $\bar{x}$ solves problem (1);
(ii) $\min\{\psi_{\bar{x}}(y) :\ y \in \mathbb{R}^n\} = \psi_{\bar{x}}(\bar{x}) = 0$;
(iii) $0 \in \bar{\partial}\psi_{\bar{x}}(\bar{x})$.

We are now in position to introduce our search direction finding subproblem. For the current feasible iterate $x^k \in \mathcal{F}$, combining subproblem (7) and the idea of feasible SQP method (see e.g. [18]), we consider the QP subproblem

$$\begin{array}{ll} \min\limits_{d\in\mathbb{R}^n,\ z\in\mathbb{R}} & z + \frac{1}{2} d^T H^k d \\ \text{s.t.} & \nabla f(x)^T d \leq z,\ \ \forall\, x \in \mathcal{B}_f^k \\ & c_i(x^k) + \nabla c_i(x)^T d \leq z,\ \ \forall\, x \in \mathcal{B}_{c_i}^k,\ i \in I. \end{array} \tag{9}$$

It is obvious that (9) is consistent, since $(d, z) = (0, 0)$ is a feasible point. Let (d^k, z^k) be an optimal solution of (9), then it is also a KKT (Karush-Kuhn-Tucker) point of (9), i.e., there exist multipliers $\mu_0^k(x) \in \mathbb{R},\ x \in \mathcal{B}_f^k,\ \mu_i^k(x) \in \mathbb{R},\ x \in \mathcal{B}_{c_i}^k,\ i \in I$ such that

$$\begin{aligned} & H^k d^k + \sum_{x\in\mathcal{B}_f^k} \mu_0^k(x)\nabla f(x) + \sum_{i\in I}\sum_{x\in\mathcal{B}_{c_i}^k} \mu_i^k(x)\nabla c_i(x) = 0, \\ & \sum_{x\in\mathcal{B}_f^k} \mu_0^k(x) + \sum_{i\in I}\sum_{x\in\mathcal{B}_{c_i}^k} \mu_i^k(x) = 1, \\ & 0 \leq \mu_0^k(x) \perp (\nabla f(x)^T d^k - z^k) \leq 0,\ x \in \mathcal{B}_f^k, \\ & 0 \leq \mu_i^k(x) \perp \left(c_i(x^k) + \nabla c_i(x)^T d^k - z^k\right) \leq 0,\ x \in \mathcal{B}_{c_i}^k,\ i \in I, \end{aligned} \tag{10}$$

where notation $a \perp b$ means $ab = 0$ for any $a, b \in \mathbb{R}$.

The following lemma describes some important properties of subproblem (9).

Lemma 2 *Suppose that* (d^k, z^k) *is an optimal solution of* (9)*. Then*

(i) $z^k \leq -(d^k)^T H^k d^k \leq 0$*;*
(ii) $z^k = 0$ *if and only if* $d^k = 0$*;*
(iii) *if* $z^k < 0$*, then* d^k *is a feasible descent direction of problem* (1) *at* x^k.

Proof

(i) From the KKT conditions (10), we can deduce that

$$z^k = -(d^k)^T H^k d^k + \sum_{i\in I}\sum_{x\in\mathcal{B}_{c_i}^k} c_i(x^k)\mu_i^k(x),$$

which together with $x^k \in \mathcal{F}$ and the nonnegativity of the multipliers shows claim (i).

(ii) If $z^k = 0$, from (i) and the positive definiteness of H^k, we have

$$0 = z^k \le -(d^k)^T H^k d^k \le 0,$$

which implies $d^k = 0$. Conversely, if $d^k = 0$, then $z^k = 0$ follows from

$$0 = \nabla f(x^k)^T d^k \le z^k \le 0.$$

(iii) If $z^k < 0$, in view of the fact that $x^k \in \mathcal{B}_f^k$ and $x^k \in \mathcal{B}_{c_i}^k,\ i \in I$, we have

$$\nabla f(x^k)^T d^k < 0 \text{ and } \nabla c_i(x^k)^T d^k < 0, \quad \text{for } c_i(x^k) = 0,\ i \in I.$$

So claim (iii) holds. □

In order to show the relation between the improvement function (3) and the direction finding subproblem (9), we define

$$\mathcal{B}^k := \left(\mathcal{B}_f^k, \mathcal{B}_{c_1}^k, \cdots, \mathcal{B}_{c_m}^k\right), \tag{11}$$

and

$$\begin{aligned} &q(d; x^k, \mathcal{B}^k, H^k) \\ &:= \max\left\{\max_{x\in\mathcal{B}_f^k}\{\nabla f(x)^T d\};\ \max\left\{\max_{x\in\mathcal{B}_{c_i}^k}\left\{c_i(x^k) + \nabla c_i(x)^T d\right\},\ i \in I\right\}\right\} \\ &\quad +\tfrac{1}{2} d^T H^k d. \end{aligned} \tag{12}$$

If (d^k, z^k) is a solution of (9), then d^k also solves

$$\min_d\ q(d; x^k, \mathcal{B}^k, H^k),$$

and the two subproblems have equal optimal objective values. It is obvious that $q(d; x^k, \mathcal{B}^k, H^k)$ is a local quadratic model of the improvement function $\psi_{x^k}(\cdot)$ at x^k, so the introduction of the improvement function (3) and the definition of the subproblem (9) are harmonious.

Define the reduction

$$\Delta q^k := q(0; x^k, \mathcal{B}^k, H^k) - q(d^k; x^k, \mathcal{B}^k, H^k).$$

In [1], such a reduction corresponding to the model of $\phi_\rho(x)$ plays a key role in both the algorithm design and the theoretical analysis.

The following lemma implies that $-z^k$ used in our algorithm plays essentially the same role as Δq^k used in [1], while the use of $-z^k$ instead of Δq^k will bring somewhat simpler presentation.

Lemma 3 *The model reduction satisfies*

$$\Delta q^k = -z^k - \frac{1}{2}(d^k)^T H^k d^k,$$

and $\Delta q^k = 0$ *if and only if* $z^k = 0$.

Proof From (12), we get

$$q(d^k; x^k, \mathcal{B}^k, H^k) = z^k + \frac{1}{2}(d^k)^T H^k d^k.$$

This along with $q(0; x^k, \mathcal{B}^k, H^k) = 0$ shows the former result. The latter one holds from Lemma 2(ii) and the positive definiteness of H^k. □

Now we present the details of our feasible SQP-GS algorithm as follows. For simplicity, we abbreviate $\psi_{x^k}(\cdot)$ as $\psi_k(\cdot)$.

Algorithm 1

```
Step 0. Initialization. Choose $\epsilon^0 > 0$, $\epsilon_{\mathrm{opt}} \geq 0$, $p \geq n+1$, $\alpha, \beta, \gamma \in (0,1)$, $\nu > 0$, $x^0 \in \mathcal{F}$, a positive definite matrix $H^0$ and set $k := 0$.
Step 1. Gradient sampling. Generate $\mathcal{B}^k$ as defined by (8) and (11).
Step 2. Direction finding. Solve problem (9) to obtain the optimal solution $(d^k, z^k)$.
Step 3. Stopping criterion. If $z^k = 0$ and $\epsilon^k \leq \epsilon_{\mathrm{opt}}$, terminate.
Step 4. Parameter updating. If $-z^k \leq \nu(\epsilon^k)^2$, then set $\epsilon^{k+1} := \gamma\epsilon^k$, $x^{k+1} := x^k$, $\lambda^k := 0$, and go to step 6; otherwise set $\epsilon^{k+1} := \epsilon^k$.
Step 5. Line search. Compute the stepsize $\lambda^k$, which is the largest value $\lambda$ in the sequence $\{1, \beta, \beta^2, \cdots\}$ satisfying
            $\psi_k(x^k + \lambda d^k) < \psi_k(x^k) + \alpha\lambda z^k.$        (13)
        Set $x^{k+1} := x^k + \lambda^k d^k$. If $x^{k+1} \notin \mathcal{D} \overset{\triangle}{=} \mathcal{D}^f \cap \mathcal{D}^{c_1} \cap \cdots \cap \mathcal{D}^{c_m}$, then replace $x^{k+1}$ with any point in $\mathcal{D}$ that satisfies
            $\psi_k(x^{k+1}) \leq \psi_k(x^k) + \alpha\lambda^k z^k$ and
            $\|x^k + \lambda^k d^k - x^{k+1}\| \leq \min\{\lambda^k, \epsilon^k\}\|d^k\|.$        (14)
Step 6. Iterative updating. Generate a new positive definite matrix $H^{k+1}$, set $k := k+1$, and go to step 1.
```

Remark 3

(i) In fact, since x^k is feasible, we always have $\psi_k(x^k) = 0$ in (13), but we retain this term for the purpose of intuition — a sufficient reduction of $\psi_k(\cdot)$ is obtained at the new iterate when compared to the current point x^k.

(ii) If $x^k + \lambda^k d^k \notin \mathcal{D}$, then $x^{k+1} \in \mathcal{D}$ can be sampled to satisfy (14) by a certain procedure that terminates finitely with probability 1 (see Lemma 5(ii) below). Since $x^{k+1} \notin \mathcal{D}$ seems unlikely to occur, as in [1, 3, 7], we do not check such possibility in the numerical implementation, i.e., we always set $x^{k+1} = x^k + \lambda^k d^k$. So (14) is not needed in practice. But it must be considered in the theoretical analysis.

Lemma 4 *For $x^k \in \mathcal{F}$, the directional derivative of $\psi_k(\cdot)$ at x^k along d^k denoted by $\psi'_k(d^k; x^k)$ satisfies*

$$\psi'_k(d^k; x^k) \le z^k.$$

Proof For $x^k \in \mathcal{F}$, the directional derivative of $\psi_k(\cdot)$ at x^k along d^k can be written by

$$\psi'_k(d^k; x^k) = \max\{\nabla f(x^k)^T d^k, \nabla c_i(x^k)^T d^k, i \in I(x^k)\},$$

where $I(x) = \{i \in I : c_i(x) = 0\}$. Further from (10), since $x^k \in \mathcal{B}^k_f \cap \mathcal{B}^k_{c_1} \cap \cdots \cap \mathcal{B}^k_{c_m}$, we have

$$\nabla f(x^k)^T d^k \le z^k \text{ and } \nabla c_i(x^k)^T d^k \le z^k,\ i \in I(x^k).$$

So the claim holds. □

The following lemma further shows that Algorithm 3 is well defined.

Lemma 5 *If Step 5 is executed during the k-th iteration, then*

(i) *the line search (13) terminates in a finite number of computations;*
(ii) *if $x^k + \lambda^k d^k \notin \mathcal{D}$, then $x^{k+1} \in \mathcal{D}$ can be sampled to satisfy (14) by a certain procedure that terminates finitely with probability 1.*

Proof

(i) From Lemma 4, we have

$$\begin{aligned} \psi_k(x^k + \lambda d^k) - \psi_k(x^k) - \alpha\lambda z^k &= \lambda\psi'_k(d^k; x^k) - \alpha\lambda z^k + o(\lambda) \\ &\le (1-\alpha)\lambda z^k + o(\lambda). \end{aligned}$$

This together with $z^k < 0$ shows that (13) holds for all $\lambda > 0$ sufficiently small, so (i) holds.

(ii) From the dense property of $\mathcal{D}$ and the continuity of $\psi_k(\cdot)$, the procedure given in [1, 7] can be applied here without any modification. More precisely, if $x^k + \lambda^k d^k \notin \mathcal{D}$, then x^{k+1} can be sampled from a uniform distribution defined on

$$\{x : \|x - (x^k + \lambda^k d^k)\| \le \min\{\lambda^k, \epsilon^k\}\|d^k\|/i\},$$

incrementing i by 1 each time until $x^{k+1} \in \mathcal{D}$ and the first inequality in (14) holds. From the dense property of $\mathcal{D}$ and the continuity of $\psi_k(\cdot)$, this procedure terminates finitely with probability 1. □

4 Global convergence

In this section, we establish the global convergence of Algorithm 3. We will show that(i) if Algorithm 3 stops finitely at x^k, then x^k is ϵ^k-stationary for $\psi_k(\cdot)$ (see Lemma 7); (ii) if Algorithm 3 generates an infinite sequence $\{x^k\}$ of iterates, then with probability one, every cluster point of $\{x^k\}$ is stationary for the improvement function (3) (see Theorem 1).

The following assumption is necessary, see also [1].

Assumption 2

(i) *The sequences of iterates and sample points generated by Algorithm 3 are contained in a convex set X over which the functions* $f, c_i (i \in I)$ *are bounded, and the first derivatives of* $f, c_i (i \in I)$ *are bounded on* $X \cap \mathcal{D}$;

(ii) *there exist constants* $b \geq a > 0$ *such that* $a\|d\|^2 \leq d^T H^k d \leq b\|d\|^2$ *for all k and* $d \in \mathbb{R}^n$.

In what follows, we suppose that x' is a given cluster point of $\{x^k\}$, i.e., there exists an infinite index set K, such that

$$x^k \to x', \quad k \in K. \tag{15}$$

It is obvious that x' is feasible since $x^k \in \mathcal{F}$ for all k. From Assumption 2(ii), we know that the sequence of matrices $\{H^k\}$ is bounded, so by passing to a subsequence if necessary, we may assume without loss of generality that

$$H^k \to H', \quad k \in K.$$

Again from Assumption 2(ii), it follows that the matrix H' is positive definite.

In order to define the stationarity, following [1], we define a local model of the improvement function $\psi_{x'}(\cdot)$ about x' by

$$\begin{aligned} &q(d; x', \mathcal{B}'^{\epsilon}, H') \\ &:= \max\left\{ \sup_{x \in \mathcal{B}'^{\epsilon}_f} \{\nabla f(x)^T d\};\ \max_{i \in I} \left\{ \sup_{x \in \mathcal{B}'^{\epsilon}_{c_i}} \left\{c_i(x') + \nabla c_i(x)^T d\right\} \right\} \right\} + \tfrac{1}{2} d^T H' d, \end{aligned}$$

where

$$\mathcal{B}'^{\epsilon} := \left(\mathcal{B}'^{\epsilon}_f, \mathcal{B}'^{\epsilon}_{c_1}, \cdots, \mathcal{B}'^{\epsilon}_{c_m}\right) := \left(\mathbb{B}_\epsilon(x') \cap \mathcal{D}^f, \mathbb{B}_\epsilon(x') \cap \mathcal{D}^{c_1}, \cdots, \mathbb{B}_\epsilon(x') \cap \mathcal{D}^{c_m}\right).$$

Let d' be the solution to the problem

$$\min_{d \in \mathbb{R}^n} q(d; x', \mathcal{B}'^{\epsilon}, H'). \tag{16}$$

Then (d', z') with

$$z' = \max\left\{ \sup_{x \in \mathcal{B}'^{\epsilon}_f} \{\nabla f(x)^T d'\};\ \max_{i \in I} \left\{ \sup_{x \in \mathcal{B}'^{\epsilon}_{c_i}} \left\{c_i(x') + \nabla c_i(x)^T d'\right\} \right\} \right\}$$

is the solution to the problem

$$\begin{aligned} \min_{d \in \mathbb{R}^n, z \in \mathbb{R}} \quad & z + \tfrac{1}{2} d^T H' d \\ & \nabla f(x)^T d \leq z,\ \forall\, x \in \mathcal{B}'^{\epsilon}_f, \\ & c_i(x') + \nabla c_i(x)^T d \leq z,\ \forall\, x \in \mathcal{B}'^{\epsilon}_{c_i},\ i \in I. \end{aligned} \tag{17}$$

Since $(0, 0)$ is a feasible solution to (17), we have

$$z' + \frac{1}{2}(d')^T H' d' \leq 0. \tag{18}$$

This together with the positive definiteness of H' shows that

$$z' \leq 0. \tag{19}$$

Denote

$$\Delta q' = q(0; x', \mathcal{B}'^{\epsilon}, H') - q(d'; x', \mathcal{B}'^{\epsilon}, H').$$

One of the important features of the penalty SQP-GS algorithm [1] is that stationarity of penalty function (2) is defined in terms of the solution of a subproblem like (16). As noted in [1], such kind of definition resembles the notion of stationarity common in smooth constrained optimization [25, 26]. It is intuitively reasonable since we can treat a point as stationary for a given function whose local quadratic model cannot gain a reduction at this point. Here, following [1] we can naturally give our definition of stationarity. It is worth pointing out that the concept to be defined (Definition 1) is independent of H's, since if the condition $d' = 0$ holds for some H', then it holds for all H's.

Definition 1

(i) *Given $\epsilon > 0$, the point $x' \in \mathcal{F}$ is ϵ-stationary for $\psi_{x'}(\cdot)$ if and only if the solution to (16) is $d' = 0$.*

(ii) *The point $x' \in \mathcal{F}$ is stationary for $\psi_{x'}(\cdot)$ if it is ϵ-stationary for every $\epsilon > 0$.*

Next, we will establish a connection between stationarity defined by Definition 1 and the standard Clarke stationarity. For any point $x \in \mathcal{F}$, we first define the point-to-set mappings

$$\hat{M}(x) = \begin{cases} \bar{\partial} f(x), & \text{if } C(x) < 0, \\ \text{co}\{\bar{\partial} f(x) \cup \bar{\partial} C(x)\}, & \text{if } C(x) = 0; \end{cases}$$

$$\tilde{M}(x) = \begin{cases} \bar{\partial} f(x), & \text{if } C(x) < 0, \\ \text{co}\{\bar{\partial} f(x) \cup \tilde{\partial} C(x)\}, & \text{if } C(x) = 0, \end{cases} \tag{20}$$

where

$$\tilde{\partial} C(x) = \text{co}\{\bar{\partial} c_i(x) : \ i \in I(x)\}, \ I(x) = \{i \in I : \ c_i(x) = 0\}.$$

From Assumption 1, we know that the functions f, c_i, $i \in I$ are all locally Lipschitz, then from Lemma 1.2.5 and relations (1.2.60)-(1.2.62) of [5], one has

$$\bar{\partial} \psi_x(x) \subseteq \hat{M}(x) \subseteq \tilde{M}(x). \tag{21}$$

Lemma 6 *Suppose that $x' \in \mathcal{F}$ is Clarke-stationary for $\psi_{x'}(\cdot)$, i.e., $0 \in \bar{\partial} \psi_{x'}(x')$. Then x' is stationary for $\psi_{x'}(\cdot)$ according to Definition 1(ii).*

Proof Let $x' \in \mathcal{F}$ be Clarke-stationary for $\psi_{x'}(\cdot)$. Then it follows from (21) that

$$0 \in \bar{\partial} \psi_{x'}(x') \subseteq \tilde{M}(x').$$

Let (d', z') be the solution to (17). We continue by considering two cases.

Case I $C(x') < 0$. In this case, from (20) and (5) we have

$$0 \in \tilde{M}(x') = \bar{\partial} f(x') = \bigcap_{\epsilon > 0} \text{cl co} \left\{ \nabla f(x) : \ x \in \mathcal{B}'^{\epsilon}_f \right\}. \tag{22}$$

In addition, from the first group of constraints of subproblem (17), we have

$$\nabla f(x)^T d' \le z', \ \forall\, x \in \mathcal{B}'^{\epsilon}_{f},$$

which further implies

$$g^T d' \le z', \ \forall\, g \in \mathrm{cl\ co}\left\{\nabla f(x) : \ x \in \mathcal{B}'^{\epsilon}_{f}\right\}. \tag{23}$$

Combining (22) with (23), for every $\epsilon > 0$, we have $z' \ge 0$. This together with (19) shows that $z' = 0$, which in turn implies from (18) that $d' = 0$ for every $\epsilon > 0$.

Case II $C(x') = 0$. In this case, it follows that

$$\begin{aligned} 0 \in \tilde{M}(x') &= \mathrm{co}\{\bar{\partial} f(x') \cup \tilde{\partial} C(x')\} \\ &= \bigcap_{\epsilon>0} \mathrm{cl\ co}\left\{\nabla f(x)\colon x \in \mathcal{B}'^{\epsilon}_{f};\ \nabla c_i(x)\colon x \in \mathcal{B}'^{\epsilon}_{c_i},\ i \in I(x')\right\}. \end{aligned} \tag{24}$$

Moreover, since $c_i(x') = 0$ for all $i \in I(x')$, from the constraints of subproblem (17), we have

$$\nabla f(x)^T d' \le z', \ \forall\, x \in \mathcal{B}'^{\epsilon}_{f};$$
$$\nabla c_i(x)^T d' \le z', \ \forall\, x \in \mathcal{B}'^{\epsilon}_{c_i}, \ i \in I(x'),$$

which implies

$$g^T d' \le z', \ \forall\, g \in \mathrm{cl\ co}\{\nabla f(x) : \ x \in \mathcal{B}'^{\epsilon}_{f};\ \nabla c_i(x) : \ x \in \mathcal{B}'^{\epsilon}_{c_i}, \ i \in I(x')\}.$$

This along with (24) shows that $z' \ge 0$ for every $\epsilon > 0$. By applying (19) again, we have $z' = 0$, and therefore $d' = 0$ for every $\epsilon > 0$.

Summarising the two cases above, we can conclude that, for every $\epsilon > 0$, the solution to (16) is $d' = 0$. Hence x' is stationary for $\psi_{x'}(\cdot)$ according to Definition 1(ii). □

Lemma 7 *If $z^k = 0$, then x^k is ϵ^k-stationary for $\psi_k(\cdot)$.*

Proof From Lemma 2, $z^k = 0$ implies $d^k = 0$, which further implies the solution to (9) with $\mathcal{B}^k$ replaced by $(\mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}^f, \mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}^{c_1}, \cdots, \mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}^{c_m})$ is also $(0, 0)$.So the lemma holds from Definition 1. □

Before giving the analysis of global convergence, we need the following lemma that describes some important properties about the sequences $\{d^k\}$ and $\{z^k\}$.

Lemma 8

(i) *The sequence $\{d^k\}_{k=0}^{\infty}$ is bounded;*

(ii) *if there is a subsequence $\mathcal{K}$ such that $d^k \xrightarrow{\mathcal{K}} 0$, then $z^k \xrightarrow{\mathcal{K}} 0$.*

Proof Since $(0, 0)$ is a feasible point for subproblem (9), we have

$$\begin{aligned} 0 \ge z^k + \tfrac{1}{2}(d^k)^T H^k d^k &\ge \nabla f(x^k)^T d^k + \tfrac{1}{2}(d^k)^T H^k d^k \\ &\ge -\|\nabla f(x^k)\|\|d^k\| + \tfrac{1}{2}a\|d^k\|^2, \end{aligned}$$

which together with Assumption 2 proves part (i).

Part (ii) follows from Assumption 2 and the fact that

$$-\|\nabla f(x^k)\|\|d^k\| \leq \nabla f(x^k)^T d^k \leq z^k \leq 0.$$

□

Next, we will show that Algorithm 3 can approximate the solution to (16) when x^k is sufficiently close to x'. Denote

$$\mathcal{S}_{\epsilon^k}(x^k) := \left(\prod_0^p \left(\mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}_f\right), \prod_0^p \left(\mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}_{c_1}\right), \cdots, \prod_0^p \left(\mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}_{c_m}\right)\right),$$

where $\prod_0^p$ refers to the Cartesian product of $p+1$ instances of sets.

Since the results of Lemma 3 also hold for $\Delta q'$ and z', we know that $-z^k$ and $-z'$ play essentially the same role as Δq^k and $\Delta q'$, respectively. In addition, from Definition 1, $-z'$ can be viewed as a measure of the proximity of point x' to ϵ-stationarity. So by slightly modifying the proof of [1, Lemma 3.8], we obtain the following lemma, which shows that if x^k lies in a sufficiently small neighborhood of x', there exists a set of sample sets that Algorithm 3 may generate that will produce a search direction d^k approximating d' with any desired accuracy.

Lemma 9 *Given $\epsilon^k > 0$, for any $\omega > 0$, there exist $\zeta > 0$ and a nonempty open subset $\mathcal{T}$ of $\mathbb{R}^{n(p+1)(m+1)}$ such that for all $x^k \in \mathbb{B}_\zeta(x')$, $k \in K$, we have*

$$\mathcal{T} \subset \mathcal{T}(x^k, x', \omega) \triangleq \left\{\mathcal{B}^k \in \mathcal{S}_{\epsilon^k}(x^k) : -z^k \leq -z' + \omega\right\}.$$

The following lemma shows that "$\epsilon^k \nrightarrow 0$ $(k \to \infty)$" is a probability zero event. Its proof is a modification of [1, Theorem 3.3] which originates from [7, Theorem 3.3].

Lemma 10 *Suppose that the sequence $\{\epsilon^k\}$ is generated by Algorithm 3. Then "$\epsilon^k \nrightarrow 0$ $(k \to \infty)$" is a probability zero event.*

Proof From Step 5, we have

$$\|x^{k+1} - x^k\| \leq \min\{\lambda^k, \epsilon^k\}\|d^k\| + \lambda^k\|d^k\| \leq 2\lambda^k\|d^k\|.$$

This together with (13), (14), Lemma 2(i) and Assumption 2(ii) shows that

$$\begin{aligned}\psi_k(x^k) - \psi_k(x^{k+1}) &\geq -\alpha\lambda^k z^k \geq \alpha\lambda^k (d^k)^T H^k d^k \\ &\geq \alpha\lambda^k a\|d^k\|^2 \geq \tfrac{1}{2}\alpha a\|x^{k+1} - x^k\|\|d^k\|,\end{aligned}$$

which, in view of Assumption 2, further implies

$$\sum_{k=0}^{\infty} \lambda^k(-z^k) < \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \|x^{k+1} - x^k\|\|d^k\| < \infty. \tag{25}$$

Suppose that $\epsilon^k \nrightarrow 0$ $(k \to \infty)$. Since $\{\epsilon^k\}$ is nonincreasing, there must exist $N > 0$ and $\epsilon' > 0$ such that $\epsilon^k = \epsilon'$ for all $k \geq N$. From Step 4 of Algorithm 3, this requires that

$$-z^k > \nu(\epsilon')^2, \quad \forall\, k \geq N, \tag{26}$$

which together with (25) implies $\lambda^k \to 0$. In addition, by combining (26) and Lemma 8(ii), we obtain that $\|d^k\|$ is bounded away from zero, and therefore from (25) that $x^k \to x'$, where x' is given by (15).

If x' is ϵ'-stationary for $\psi_{x'}(\cdot)$, then from Definition 1, the solution to problem (16) is $d' = 0$, which in turn implies $z' = 0$. Set $\omega = \nu(\epsilon')^2$ and choose $\zeta, \mathcal{T}$ satisfying Lemma 9, so there exists $N' \geq N$ such that $x^k \in \mathbb{B}_\zeta(x')$ for all $k \geq N'$ and as long as $\mathcal{B}^k \in \mathcal{T}$, $k \in K$, we have

$$-z^k \leq \nu(\epsilon')^2, \quad \forall\, k \in K, \ k \geq N'.$$

This in conjunction with (26) implies that for every $k \in K$ and $k \geq N'$, it must be the case that $\mathcal{B}^k \notin \mathcal{T}$. But this is a probability zero event, since $\mathcal{T}$ is an open set and the points in $\mathcal{B}^k$ are sampled uniformly from $\mathcal{S}_{\epsilon^k}(x^k)$.

If x' is not ϵ'-stationary, then for any λ not satisfying the line search (13) yields

$$\psi_k(x^k + \lambda d^k) - \psi_k(x^k) \geq \alpha\lambda z^k. \tag{27}$$

On the other hand, by Assumption 2 and Lemma 4, we have

$$\psi_k(x^k + \lambda d^k) - \psi_k(x^k) \leq \lambda z^k + \lambda^2 M^k \|d^k\|^2, \tag{28}$$

where

$$M^k = \sup\left\{ \frac{\left|\psi_k'(d^k; y) - \psi_k'(d^k; x^k)\right|}{\lambda\|d^k\|} \ : \ y \in [x^k, x^k + \lambda d^k] \right\} + 1,$$

which exists and is finite by Assumption 1 (see [1, Theorem 3.3]). In addition, from Assumptions 1 and 2, $x^k \to x'$ and Lemma 8(i), it follows that the sequence $\{M^k\}$ is bounded and therefore since it was proved that $\|d^k\|$ is bounded away from zero, there exists a constant $\bar{M} > 0$ such that

$$M^k \|d^k\|^2 \leq \bar{M}. \tag{29}$$

Combining (27) with (28), we have

$$\lambda \geq (\alpha - 1) z^k / M^k \|d^k\|^2. \tag{30}$$

Since β is the backtracking factor of the line search in Algorithm 3, we know that $\beta^{-1}\lambda^k$ does not satisfy (13). So substituting λ by $\beta^{-1}\lambda^k$ in (30), we have

$$\lambda^k \geq \beta(\alpha - 1) z^k / M^k \|d^k\|^2. \tag{31}$$

On the other hand, letting $\omega = -z' > 0$ and $\zeta, \mathcal{T}$ chosen as Lemma 9, there exists $N'' \geq N$ such that $x^k \in B_\zeta(x')$ for all $k \geq N''$ and as long as $\mathcal{B}^k \in \mathcal{T}$, $k \in K$, we have

$$-z^k \leq -2z', \quad \forall\, k \in K, \ k \geq N''.$$

This together with (29) and (31) shows that λ^k is bounded away from zero for all $k \in K$, $k \geq N''$ such that $\mathcal{B}^k \in \mathcal{T}$. This is also a probability zero event, since $\lambda^k \to 0$ implies that $\mathcal{B}^k \notin \mathcal{T}$ for all $k \in K$, $k \geq N''$. □

Now we present the global convergence result whose proof is a tiny modification of [1, Theorem 3.3] which originates from [7, Theorem 3.3]. From Lemma 7, we know that if Algorithm 3 terminates finitely at Step 3, then x^k is ϵ^k-stationary for $\psi_k(\cdot)$. In what follows, we assume that Algorithm 3 generates an infinite sequence of iterates.

Theorem 1 *Let $\{x^k\}$ be a sequence of iterates generated by Algorithm 3. With probability one, every cluster point of $\{x^k\}$ is stationary for the improvement function* (3).

Proof From Lemma 10, we know that "$\epsilon^k \nrightarrow 0$ $(k \to \infty)$" is a probability zero event. Now suppose that

$$\epsilon^k \downarrow 0, \ k \to \infty \tag{32}$$

and x' is a cluster point of the sequence $\{x^k\}$ given by (15). We will first show that

$$\liminf_{k\to\infty} \ \max\left\{\|x^k - x'\|, \|d^k\|\right\} = 0. \tag{33}$$

If $x^k \to x'$, then from Step 4 of Algorithm 3 and Lemma 2(i), we know that $\epsilon^k \downarrow 0$ if and only if there exists an infinite subsequence $\mathcal{K}$ satisfying

$$a\|d^k\|^2 \le (d^k)^T H^k d^k \le -z^k \le \nu(\epsilon^k)^2, \ \ \forall\, k \in \mathcal{K}.$$

This implies $d^k \xrightarrow{\mathcal{K}} 0$, and thus (33) holds.

On the other hand, if $x^k \nrightarrow x'$, then suppose by contradiction that (33) does not hold. Since x' is a cluster point of $\{x^k\}$, there exist $\epsilon' > 0$ and an index $k' \ge 0$ such that the set

$$K' := \{k : \ k \ge k', \ \|x^k - x'\| \le \epsilon', \ \|d^k\| > \epsilon'\}$$

is infinite. This together with the second inequality of (25) shows that

$$\sum_{k\in K'} \|x^{k+1} - x^k\| < \infty. \tag{34}$$

Since $x^k \nrightarrow x'$, there exists an $\epsilon > 0$ such that for all $k_1 \in K'$ with $\|x^{k_1} - x'\| \le \epsilon'/2$ there is $k_2 > k_1$ satisfying

$$\|x^{k_1} - x^{k_2}\| > \epsilon \ \text{ and } \ \|x^k - x'\| \le \epsilon', \ \ \forall\, k_1 \le k < k_2.$$

Therefore, by the triangle inequality, we have

$$\epsilon < \|x^{k_1} - x^{k_2}\| \le \sum_{k=k_1}^{k_2-1} \|x^{k+1} - x^k\|. \tag{35}$$

However, for $k_1 \in K'$ sufficiently large, (34) implies that the right-hand side of (35) is less than ϵ, a contradiction. Thus (33) holds. This together with Assumption 2(ii) and Lemma 8(ii) shows that there exists an infinite index set $\bar{K}$ such that

$$x^k \to x', \ d^k \to 0, \ H^k d^k \to 0, \ z^k \to 0, \ k \in \bar{K}. \tag{36}$$

Denote

$$\hat{G}_\epsilon(x) = \text{cl co}\left\{\nabla f(y) : \ y \in \mathbb{B}_\epsilon(x) \cap \mathcal{D}^f; \ \nabla c_i(y) : \ y \in \mathbb{B}_\epsilon(x) \cap \mathcal{D}^{c_i}, \ i \in I(x')\right\},$$

where $I(x') = \{i \in I : c_i(x') = 0\}$, and denote

$$\widetilde{G}_\epsilon(x) = \bigcap_{j=1}^{\infty} \hat{G}_{\epsilon+\theta_j}(x),$$

where $\{\theta_j\}_{j=1}^{\infty}$ is any sequence of positive numbers converging downward to *0*. From Proposition 2.7 of [4] (with tiny extension), it follows that

$$the\ mapping\ \ x \mapsto \widetilde{G}_\epsilon(x)\ \ is\ upper\text{-}semi\text{-}continuous, \tag{37}$$

i.e., for any sequences $\{x^k\}$ and $\{v^k\}$ satisfying $x^k \to \bar{x}$, $v^k \to \bar{v}$ and $v^k \in \widetilde{G}_\epsilon(x^k)$, one has $\bar{v} \in \widetilde{G}_\epsilon(\bar{x})$.

Next, we will show that $0 \in \hat{G}_\epsilon(x')$ for every $\epsilon > 0$. For any given $\epsilon > 0$, from (32), (36) and the last relation of KKT conditions (10), there exists an index $\bar{k}$ such that

$$\epsilon^k \le \epsilon/3,\ \forall\, k \ge \bar{k},$$

and (since $c_i(x') < 0,\ i \in I \setminus I(x')$)

$$\mu_i^k(x) = 0,\ x \in \mathcal{B}_{c_i}^k,\ i \in I \setminus I(x'),\ \forall\, k \ge \bar{k},\ k \in \bar{K}.$$

This together with the KKT conditions (10) shows that

$$-H^k d^k \in \hat{G}_k,\ \forall\, k \ge \bar{k},\ k \in \bar{K},$$

where

$$\hat{G}_k = \mathrm{cl\ co}\{\nabla f(y) : \ y \in \mathcal{B}_f^k;\ \nabla c_i(y) : \ y \in \mathcal{B}_{c_i}^k,\ i \in I(x')\}.$$

Therefore, from

$$\mathcal{B}^k \subset \left(\mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}^f, \mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}^{c_1}, \cdots, \mathbb{B}_{\epsilon^k}(x^k) \cap \mathcal{D}^{c_m}\right),$$

we have

$$-H^k d^k \in \hat{G}_k \subseteq \hat{G}_{\epsilon^k}(x^k) \subseteq \widetilde{G}_{\epsilon/2}(x^k),\ \forall\, k \ge \bar{k},\ k \in \bar{K}.$$

This along with (36) and (37) shows that

$$0 \in \widetilde{G}_{\epsilon/2}(x')$$

which, from the relation $\widetilde{G}_{\epsilon/2}(x') \subseteq \hat{G}_\epsilon(x')$, in turn implies that $0 \in \hat{G}_\epsilon(x')$. Therefore, since the above $\epsilon(> 0)$ is arbitrary, we can conclude that

$$0 \in \hat{G}_\epsilon(x') \tag{38}$$

for every $\epsilon > 0$.

Finally, let (d', z') be the solution to (17), then from the constraints of (17) and the notation of $\hat{G}_\epsilon(x)$ we have

$$g^T d' \le z',\ \forall\, g \in \hat{G}_\epsilon(x').$$

This together with (38) shows that $z' \ge 0$, and therefore from (19) and (18) we have that $z' = 0$ and $d' = 0$ for every $\epsilon > 0$. So, from Definition 1(ii), we can conclude that x' is stationary for the improvement function $\psi_{x'}(\cdot)$. □

5 Numerical Results

In this section, we aim to test the practical effectiveness of Algorithm 3 (denoted by FSQP-GS below[1]). We tested the following four problems (P1-P4), where P1 is taken from [1], P2 is taken from [27, 28], and P3 and P4 are taken from [29]. Among them, P2 is a convex program, while the other three are not. We compared the numerical results reported by FSQP-GS with the software SLQP-GS [1], which (as far as we are aware) is the only constrained GS algorithm proposed to date, and therefore is the only valid alternative. At the time of this writing, the SLQP-GS code can be downloaded freely from the link: http://coral.ie.lehigh.edu/~frankecurtis/software. We should point out that this is only a "rough" comparison, since the same starting point (on different runs) may produce different results because of the stochastic nature of the GS-type algorithms.

All numerical experiments were implemented by using MATLAB, and on a PC with 1.46GHz CPU. The quadratic programming solver is MOSEK [30]. In our experiments, we chose the same values for all the parameters and the same updating strategy for H^k as in SLQP-GS [1] (with default setting). The stopping criterion at Step 3 is: $|z^k| \le 10^{-6}$ and $\epsilon^k \le 10^{-4}$.

The numerical results are listed in Tables 1, 2, 3, 4 whose columns have the following meanings:

NI —— number of iterations;
NF —— number of objective function evaluations;
NC —— number of constraints evaluations;
Time —— CPU time (sec.);
ObjValue —— final objective value.

P1 Find the minimizer of a nonsmooth Rosenbrock function subject to an inequality constraint on a weighted maximum value of the variables (see [1]):

$$\begin{aligned} &\min\ 8\left|x_1^2 - x_2\right| + (1 - x_1)^2 \\ &\text{s.t.}\ \ \max\left\{\sqrt{2}x_1, 2x_2\right\} \le 1. \end{aligned}$$

The solution of this problem is $x^* = (\frac{\sqrt{2}}{2}, \frac{1}{2})$ at which both the objective and the constraint function are nondifferentiable.

P2 Rosen-Suzuki problem [27, 28].

$$\begin{aligned} \min\ \ & f(x) = \max\{f_j(x) : j = 1, \cdots, 4\} \\ \text{s.t.}\ \ & \max\{c_i(x) : i = 1, \cdots, 3\} \le 0, \end{aligned}$$

where $f_1(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$,
$f_2(x) = f_1(x) + 10c_1(x),\ \ f_3(x) = f_1(x) + 10c_2(x)$,
$f_4(x) = f_1(x) + 10c_3(x)$,
$c_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8$,
$c_2(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10$,
$c_3(x) = x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5$.

[1] The Matlab code of FSQP-GS is available from the authors.

Table 1 Comparison between SLQP-GS and FSQP-GS for P1

No.	x^0	Algorithm	NI	NF	NC	Time	ObjValue
1	$(0.066661, -0.350366)^T$	SLQP-GS	69	287	70	3.0	0.0857869
		FSQP-GS	65	234	66	2.4	0.0857870
2	$(0.433746, -1.447530)^T$	SLQP-GS	58	259	59	2.1	0.0857864
		FSQP-GS	67	166	68	2.2	0.0857910
3	$(-0.889838, -1.354211)^T$	SLQP-GS	74	287	75	2.9	0.0857923
		FSQP-GS	68	204	69	2.3	0.0857870
4	$(0.314871, 0.317637)^T$	SLQP-GS	48	165	49	2.1	0.0857952
		FSQP-GS	54	166	55	2.6	0.0857870
5	$(0.049795, -0.344264)^T$	SLQP-GS	55	192	56	3.0	0.0858014
		FSQP-GS	68	217	69	3.0	0.0857870
6	$(0.202167, -0.174977)^T$	SLQP-GS	57	249	58	2.1	0.0857865
		FSQP-GS	72	199	73	2.5	0.0857880
7	$(-1.921442, 0.401340)^T$	SLQP-GS	74	318	75	3.5	0.0857872
		FSQP-GS	85	217	86	3.4	0.0857990
8	$(-0.722157, -0.519986)^T$	SLQP-GS	41	156	42	3.2	0.0858019
		FSQP-GS	46	122	47	2.5	0.0858900
9	$(0.552076, -1.549885)^T$	SLQP-GS	59	221	60	2.4	0.0857872
		FSQP-GS	59	147	60	2.6	0.0857970
10	$(-1.215498, -0.714855)^T$	SLQP-GS	58	222	59	5.0	0.0857888
		FSQP-GS	72	190	73	3.9	0.0857950
	Average	SLQP-GS	59	236	60	3.0	0.0857914
		FSQP-GS	66	186	67	2.8	0.0858008

P3 Taken from [29] (objective 2.8 + constraint 4.2(ii)).

$$\begin{array}{ll}\min & \sum_{i=1}^{n-1}(-x_i + 2(x_i^2 + x_{i+1}^2 - 1) + 1.75|x_i^2 + x_{i+1}^2 - 1|) \\ \text{s.t.} & \sum_{i=1}^{n-2}((3 - 2x_{i+1})x_{i+1} - x_i - 2x_{i+2} + 2.5) \le 0,\end{array}$$

where $n = 10$.

P4 Taken from [29] (objective 2.6 + constraint 4.2(i)).

$$\begin{array}{ll}\min & \max_{1\le i\le n}\{g(-\sum_{i=1}^{n} x_i), g(x_i)\} \\ \text{s.t.} & \sum_{i=1}^{n-2}((3 - 0.5x_{i+1})x_{i+1} - x_i - 2x_{i+2} + 1.0) \le 0,\end{array}$$

where $g(y) = \ln(|y| + 1),\ \forall y \in \mathbb{R}, n = 20$.

From the numerical results listed in Tables 1–4, we see that FSQP-GS is comparable with SLQP-GS. We analyze the results in more detail as follows:

(i) For problem P1, we tested ten randomly generated starting points. The last row of Table 1 gives the average performance of these ten runs. Table 1 shows that FSQP-GS needs slightly larger number of iterations than SLQP-GS. This is expected, since FSQP-GS is a feasible algorithm, so in order to force feasibility

Table 2 Comparison between SLQP-GS and FSQP-GS for P2

No.	x^0	Algorithm	NI	NF	NC	Time	ObjValue
1	$(1, 1, 1, 1)^T$	SLQP-GS	43	137	44	4.619	44.00000
		FSQP-GS	51	151	52	4.011	43.99800
2	$(0, 0, 0, 0)^T$	SLQP-GS	50	169	51	4.093	44.00000
		FSQP-GS	56	132	57	3.887	43.99905
3	$(0.4031, 0.5233, 0.3925, 0.0670)^T$	SLQP-GS	44	163	45	4.719	44.00000
		FSQP-GS	56	138	57	4.920	43.99921
4	$(0.1838, 0.8868, 0.2135, 0.5428)^T$	SLQP-GS	50	173	51	5.037	44.00000
		FSQP-GS	55	163	56	4.809	43.99843
5	$(0.9473, 0.0914, 0.3827, 0.7305)^T$	SLQP-GS	43	162	44	3.045	44.00000
		FSQP-GS	58	146	59	3.296	43.99876
6	$(0.4668, 0.8179, 0.0832, 0.0673)^T$	SLQP-GS	40	158	41	3.420	44.00000
		FSQP-GS	56	164	57	3.705	43.99862
7	$(0.1673, 0.8474, 0.4847, 0.7001)^T$	SLQP-GS	45	155	46	4.216	44.00000
		FSQP-GS	52	178	53	4.386	43.99879
8	$(0.0603, 0.3285, 0.0288, 0.5631)^T$	SLQP-GS	44	140	45	4.405	44.00000
		FSQP-GS	52	162	53	5.124	43.99893
9	$(0.4796, 0.2130, 0.6354, 0.6415)^T$	SLQP-GS	42	174	43	3.118	44.00000
		FSQP-GS	55	167	56	3.473	43.99899
10	$(0.6448, 0.1792, 0.1448, 0.4797)^T$	SLQP-GS	47	146	48	4.826	44.00000
		FSQP-GS	58	192	59	5.182	43.99930
	Average	SLQP-GS	45	158	46	4.150	44.00000
		FSQP-GS	55	159	56	4.280	43.99881

it may produce shorter steps and therefore needs more iterations. However, Table 1 also shows that FSQP-GS generally needs a fewer number of objective function evaluations than SLQP-GS. So roughly speaking, these two algorithms are comparable for problem P1.

(ii) For problem P2, we also tested ten randomly generated starting points. From Table 2, we see that FSQP-GS needs slightly larger number of iterations and number of constraints evaluations than SLQP-GS, while the number of objective function evaluations seems comparable. In general, FSQP-GS performs slightly worse than SLQP-GS for problem P2.

(iii) For problem P3, we tested five different starting points (multiple of ones). For simplicity, we use a MATLAB command `ones(n,1)` to denote an $n \times 1$ vector of ones. From Table 3, it seems that FSQP-GS performs slightly better than SLQP-GS for problem P3.

(iv) For problem P4, we tested a fixed starting point $x^0 = 2.0 \times \text{ones}(20, 1)$. In our implementation we set the maximum iteration as 2500 and ran the two algorithms six times under the given starting point. Both the two algorithms either stop normally or reach the maximum iteration. For comparison, we also

Table 3 Comparison between SLQP-GS and FSQP-GS for P3

No.	x^0	Algorithm	NI	NF	NC	Time	ObjValue
1	2*ones(10,1)	SLQP-GS	140	962	141	10.288	18.238952
		FSQP-GS	158	355	159	7.306	18.239047
2	3*ones(10,1)	SLQP-GS	141	1005	142	11.123	18.238950
		FSQP-GS	174	441	175	8.277	18.239000
3	4*ones(10,1)	SLQP-GS	120	767	121	9.811	18.238950
		FSQP-GS	175	431	176	7.382	18.239085
4	5*ones(10,1)	SLQP-GS	125	720	126	8.550	18.238952
		FSQP-GS	163	392	164	6.866	18.239055
5	6*ones(10,1)	SLQP-GS	138	1025	139	8.968	18.238950
		FSQP-GS	158	389	159	6.731	18.239035
	Average	SLQP-GS	133	896	134	9.748	18.238951
		FSQP-GS	166	402	167	7.312	18.239044

used the public software SolvOpt [31] to solve this problem and obtained an (approximately) optimal objective value 0.751267. From Table 4, we see that FSQP-GS always produce better (lower) objective values than SLQP-GS for problem P4.

From the timing comparison, we also see that FSQP-GS is comparable with SLQP-GS for problems P1 and P2, and that it outperforms SLQP-GS on solving P3.

In addition, we observed that both algorithms reach almost the same "optimal" value for problems P1, P2 and P3, while FSQP-GS reaches obviously lower objective value for problem P4. This might be the reason that the former three problems are relatively simple, while the fourth problem is somewhat complex. By comparing with

Table 4 Comparison between SLQP-GS and FSQP-GS for P4

No.	Algorithm	NI	ObjValue
1	SLQP-GS	1546	1.157452
	FSQP-GS	1955	0.762889
2	SLQP-GS	1582	0.940005
	FSQP-GS	2500	0.768407
3	SLQP-GS	2500	0.853981
	FSQP-GS	2269	0.751243
4	SLQP-GS	782	1.157453
	FSQP-GS	2500	0.767875
5	SLQP-GS	782	1.157453
	FSQP-GS	1460	0.751243
6	SLQP-GS	761	1.157453
	FSQP-GS	2500	0.762935

the analytic global solution (which is known) of P1, we see that both algorithms reach (approximately) global solutions for P1. Also, the "optimal" value for P2 is hopefully the (approximately) global minimum, since P2 is a convex problem. But it seems hard to judge whether the "optimal" value for P3 is the global minimum. For problem P4, it is obvious that `SLQP-GS` does not reach the global minimum, since `FSQP-GS` produces obviously lower objective value which is very close to that calculated by the public software `SolvOpt`. Nevertheless, we cannot guarantee that `FSQP-GS` reaches the global minimum for P4.

During the test, we also observed that `SLQP-GS` may generate infeasible iteration points and infeasible approximately optimal solutions, though the starting points are all feasible. This is expected, since `SLQP-GS` is a penalty function type method. In contrast, `FSQP-GS` always generates feasible iteration points and approximately optimal solutions, since we use an improvement function to ensure the feasibility of the constraints and the decrease of the objective function. Therefore, if the problems needed to be solved require feasibility of the iteration points and/or approximately optimal solutions, and the feasible starting points are (relatively) easily available, `FSQP-GS` will be a suitable choice.

6 Conclusions

In this paper, we have presented a feasible SQP-GS algorithm for solving nonconvex, nonsmooth constrained optimization. It generates a sequence of feasible iterates, and guarantees that the objective function is monotonically decreasing. By making use of an improvement function, global convergence of the algorithm is proved. Limited numerical results show that the proposed algorithm is promising. As future work, we would extend the feasible SQP-GS algorithm to accept infeasible starting points by combining the idea of the strongly sub-feasible direction methods [20–22]. Also, we may further improve the proposed algorithm by making use of the adaptive strategy given in Curtis and Que [9].

Acknowledgments The authors wish to thank the Editor-in-Chief, Prof. Claude Brezinski, and two anonymous referees for their helpful comments and suggestions, which led to great improvement of the original version of this paper. They also thank Prof. Frank E. Curtis for his generous help on understanding some proofs of the penalty SQP-GS method.

References

1. Curtis, F.E., Overton, M.L.: A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization. SIAM J. Optim. **22**, 474–500 (2012)
2. Burke, J.V., Lewis, A.S., Overton, M.L.: Two numerical methods for optimizing matrix stability. Linear Algebra Appl. **351/352**, 147–184 (2002)
3. Burke, J.V., Lewis, A.S., Overton, M.L.: A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM J. Optim. **15**, 751–779 (2005)
4. Goldstein, A.A.: Optimization of Lipschitz continuous functions. Math. Program **13**, 14–22 (1977)
5. Kiwiel, K.C.: Methods of Descent for Nondifferentiable Optimization. Lecture Notes in Mathematics. Springer-Verlag, Berlin (1985)

6. Sagastizábal, C.A., Solodov, M.V.: An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. SIAM J. Optim. **16**, 146–169 (2005)
7. Kiwiel, K.C.: Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization. SIAM J. Optim. **18**, 379–388 (2007)
8. Kiwiel, K.C.: A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization. SIAM J. Optim. **20**, 1983–1994 (2010)
9. Curtis, F.E., Que, X.: An adaptive gradient sampling algorithm for nonsmooth optimization. Optim. Method Softw. (2012). doi:10.1080/10556788.2012.714781
10. Burke, J.V., Lewis, A.S., Overton, M.L.: Pseudospectral components and the distance to uncontrollability. SIAM J. Matrix Anal. Appl. **26**, 350–361 (2004)
11. Lewis, A.S.: Local structure and algorithms in nonsmooth optimization. Optimization and Applications, Mathematisches Forschungsinstitut Oberwolfach, Oberwolfach, Germany (2005)
12. Boggs, P.T., Tolle, J.W.: Sequential quadratic programming. Acta Numerica, pp. 1–51. Cambridge University Press, Cambridge (1995)
13. Schittkowski, K., Yuan, Y.X.: Sequential quadratic programming methods. Wiley encyclopedia of operations research and management science. (2011). doi:10.1002/9780470400531.eorms0984
14. Karmitsa, N., Tanaka Filho, M., Herskovits, J.: Globally convergent cutting plane method for nonconvex nonsmooth minimization. J. Optim. Theory Appl. **148**, 528–549 (2011)
15. Herskovits, J., Freire, W.P., Tanaka Filho, M., Canelas, A.: A feasible directions method for nonsmooth convex optimization. Struct. Multidisc. Optim. **44**, 363–377 (2011)
16. Panier, E.R., Tits, A.L.: A superlinearly convergent feasible method for the solution of inequality constrained optimization problems. SIAM J. Control Optim. **25**, 934–950 (1987)
17. Panier, E.R., Tits, A.L.: On combining feasibility, descent and superlinear convergence in inequality constrained optimization. Math. Program. **59**, 261–276 (1993)
18. Lawarence, C.T., Tits, A.L.: A computationally efficient feasible sequential quadratic programming algorithm. SIAM J. Optim. **11**, 1092–1118 (2001)
19. Jian, J.B.: New sequential quadratically constrained quadratic programming norm-relaxed method of feasible directions. J. Optim. Theory Appl. **129**, 109–130 (2006)
20. Jian, J.B., Tang, C.M., Hu, Q.J., Zheng, H.Y.: A new superlinearly convergent strongly sub-feasible sequential quadratic programming algorithm for inequality-constrained optimization. Numer. Funct. Anal. Optim. **29**, 376–409 (2008)
21. Jian, J.B., Tang, C.M., Zheng, H.Y.: Sequential quadratically constrained quadratic programming algorithm of strongly sub-feasible directions. Eur. J. Oper. Res. **200**, 645–657 (2010)
22. Tang, C.M., Jian, J.B.: Strongly sub-feasible direction method for constrained optimization problems with nonsmooth objective functions. Eur. J. Oper. Res. **218**, 28–37 (2012)
23. Polak, E., Trahan, R., Mayne, D.Q.: Combined phase I-phase II methods of feasible directions. Math. Program. **17**, 61–73 (1979)
24. Clarke, F.H.: Optimization and Nonsmooth Analysis. Willey, New York (1983)
25. Byrd, R.H., Lopez-Calva, G., Nocedal, J.: A line search exact penalty method using steering rules. Math. Program. **133**, 39–73 (2012)
26. Han, S.P., Mangasarian, O.L.: Exact penalty functions in nonlinear programming. Math. Program. **17**, 251–269 (1979)
27. Asaadi, J.: A computational comparison of nonlinear programs. Math. Program. **4**, 144–154 (1973)
28. Rustem, B., Nguyen, Q.: An algorithm for the inequality-constrained discrete minimax problem. SIAM J. Optim. **8**, 265–283 (1998)
29. Karmitsa, N.: Test problems for large-scale nonsmooth minimization. Reports of the Department of Mathematical Information Technology, Series B, Scientific computing, No. B 4/2007. University of Jyväskylä. Jyväskylä (2007)
30. Mosek ApS: The MOSEK optimization toolbox for MATLAB manual. Version 6.0, http://www.mosek.com/
31. Kuntsevich, A., Kappel, F.: SolvOpt—The solver for local nonlinear optimization problems: Matlab, C and Fortran source codes. Institute for Mathematics, Karl-Franzens University of Graz. http://www.uni-graz.at/imawww/kuntsevich/solvopt/ (1997)