

## Delaunay-based derivative-free optimization via global surrogates, part II: convex constraints

Pooriya Beyhaghi<sup>1</sup> · Thomas R. Bewley<sup>1</sup>

Received: 9 May 2015 / Accepted: 8 April 2016 / Published online: 19 April 2016  
© Springer Science+Business Media New York 2016

**Abstract** The derivative-free global optimization algorithms developed in Part I of this study, for linearly constrained problems, are extended to nonconvex  $n$ -dimensional problems with convex constraints. The initial  $n + 1$  feasible datapoints are chosen in such a way that the volume of the simplex generated by these datapoints is maximized; as the algorithm proceeds, additional feasible datapoints are added in such a way that the convex hull of the available datapoints efficiently increases towards the boundaries of the feasible domain. Similar to the algorithms developed in Part I of this study, at each step of the algorithm, a search function is defined based on an interpolating function which passes through all available datapoints and a synthetic uncertainty function which characterizes the distance to the nearest datapoints. This uncertainty function, in turn, is built on the framework of a Delaunay triangulation, which is itself based on all available datapoints together with the (infeasible) vertices of an exterior simplex which completely contains the feasible domain. The search function is minimized within those simplices of this Delaunay triangulation that do not include the vertices of the exterior simplex. If the outcome of this minimization is contained within the circumsphere of a simplex which includes a vertex of the exterior simplex, this new point is projected out to the boundary of the feasible domain. For problems in which the feasible domain includes edges (due to the intersection of multiple twice-differentiable constraints), a modified search function is considered in the vicinity of these edges to assure convergence.

**Keywords** Derivative-free · Global surrogate · Convex constraints · Delaunay triangulation

---

✉ Pooriya Beyhaghi  
pbeyhagh@ucsd.edu

Thomas R. Bewley  
bewley@ucsd.edu

<sup>1</sup> Flow Control Lab, University of California San Diego, La Jolla, CA, USA

## 1 Introduction

In this paper, a new derivative-free optimization algorithm is presented to minimize a (possibly nonconvex) function subject to convex constraints<sup>1</sup> on a bounded feasible region in parameter space:

$$\text{minimize } f(x) \text{ with } x \in L = \{x \mid c_i(x) \leq 0, \forall i = \{1, 2, \dots, m\}\}, \quad (1)$$

where  $c_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  are assumed to be convex, and both  $f(x)$  and the  $c_i(x)$  assumed to be twice differentiable. Moreover, the feasible domain  $L$  is assumed to be bounded with a nonempty interior (note that this assumption is only technical: if a given feasible domain has an empty interior, relaxing these constraints by  $\epsilon$  generates a feasible domain with a nonempty interior; further related discussion on this matter is deferred to the last paragraph of Sect. 5.2).

The algorithms developed in Part I (see [7]) of this study were restricted to problems with linear constraints, as the domain searched was limited to the convex hull of the initial datapoints, which in Part I was taken as all vertices of the (there, polyhedral) feasible domain. Another potential drawback of the approach taken in Part I was the expense of the initialization of the algorithm:  $2^n$  initial function evaluations were needed in the case of box constraints, and many more initial function evaluations were needed when there were many linear constraints. This paper addresses both of these issues.

Constrained optimization problems have been widely considered with local optimization algorithms in both the derivative-based and the derivative-free settings. For global optimization algorithms, the precise nature of the constraints on the feasible region of parameter space is a topic that has received significantly less attention, as many global optimization methods (see for e.g., [8, 27–29, 40, 42, 44, 48]) have very similar implementations in problems with linear and nonlinear constraints.

There are three classes of approaches for Nonlinear Inequality Problems (NIPs) using local derivative-based methods. Those in the first class, called sequential quadratic programming methods (see [21, 24]), impose (and, successively, update) a local quadratic model of the objective function  $f(x)$  and a local linear model of the constraints  $c_i(x)$  in order to estimate the local optimal solution at each step. These models are defined based on the local gradient and Hessian of the objective function  $f(x)$ , and the local Jacobian of the constraints  $c_i(x)$ , at the datapoint considered at each step. Those in the second class, called quadratic penalty methods (see [10, 30, 31]), perform some function evaluations outside of the feasible domain, with a quadratic term added to the cost function which penalizes violations of the feasible domain boundary, and solves a sequence of subproblems with successively stronger penalization terms in order to ultimately solve the problem of interest. Those in the third class, called interior point methods (see [22]), perform all function evaluations inside the feasible domain, with a log barrier term added to the cost function which penalizes proximity to the feasible domain boundary (the added term goes to infinity at the domain boundary), and solves a sequence of subproblems with successively weaker penalization terms in order to ultimately solve the problem of interest.

NIPs are a subject of significant interest in the derivative-free setting as well. One class of derivative-free optimization methods for NIPs is called direct methods [34], which includes the well-known General Pattern Search (GPS) [47] methods which restrict all function evalu-

<sup>1</sup> The representation of a convex feasible domain as stated in (1) is the standard form used, e.g., in [15], but is not completely general. Certain convex constraints of interest, such as those implied by linear matrix inequalities (LMIs), can not be represented in this form. The extension of the present algorithm to feasible domains bounded by LMIs will be considered in future work.

ations to lie on an underlying grid which is successively refined. GPS methods were initially designed for unconstrained problems, but have been modified to address box-constrained problems [33], linearly-constrained problems [34], and smooth nonlinearly-constrained problems [35]. Mesh Adaptive Direct Search (MADS) algorithms [1–4] are modified GPS algorithms that handle non-smooth constraints. GPS and MADS algorithms have been extended in [9] to handle coordination with lattices (that is, non-Cartesian grids) given by  $n$ -dimensional sphere packings, which significantly improves efficiency in high dimensional problems.

The leading class of derivative-free optimization algorithms today is known as Response Surface Methods. Methods of this class leverage an underlying inexpensive model, or “surrogate”, of the cost function. Kriging interpolation is often used to develop this surrogate [8]; this convenient choice provides both an interpolant and a model of the uncertainty of this interpolant, and can easily handle extrapolation from the convex hull of the data points out to the (curved) boundaries of a feasible domain bounded by nonlinear constraints. Part I of this study summarized some of the numerical issues associated with the Kriging interpolation method, and developed a new Response Surface Method based on *any* well-behaved interpolation method, such as polyharmonic spline interpolation, together with a synthetic model of the uncertainty of the interpolant built upon a framework provided by a Delaunay triangulation.

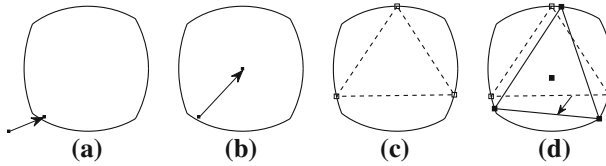
Unfortunately, the uncertainty function used in Part I of this study is only defined within the convex hull of the available datapoints, so the algorithms described in Part I do not extend immediately to more general problems with convex constraints. The present paper develops the additional machinery necessary to make this extension effectively, by appropriately increasing the domain which is covered by convex hull of the datapoints as the algorithm proceeds. As in Part I, we consider optimization problems with expensive cost function evaluations but computationally inexpensive constraint function evaluations; we further assume that the computation of the surrogate function has a low computational cost. The algorithm developed in Part II of this study has two significant advantages over those developed in Part I: (a) it solves a wider range of optimization problems (with more general constraints), and (b) the number of initial function evaluations is reduced (this is significant in relatively high-dimensional problems, and in problems with many constraints).

The paper is structured as follows. Section 2 discusses briefly how the present algorithm is initialized. Section 3 presents the algorithm. Section 4 analyzes the convergence properties of the algorithm. In Sect. 5, the optimization algorithm proposed is applied to a number of test functions with various different constraints in order to quantify its behavior. Conclusions are presented in Sect. 6.

## 2 Initialization

In contrast to the algorithms developed in Part I, the algorithm developed here is initialized by performing initial function evaluations at only  $n + 1$  feasible points. There is no need to cover the entire feasible domain by the convex hull of the initial datapoints; however, it is more efficient to initialize with a set of  $n + 1$  points whose convex hull has the maximum possible volume.

Before calculating the initial datapoints to be used, a feasible point  $x_f$  which satisfies all constraints must be identified. The feasible domain considered [see (1)] is  $L = \{x | c_i(x) \leq 0, 1 \leq i \leq m\}$ , where the  $c_i(x)$  are convex. The convex feasibility problem (that is, finding



**Fig. 1** Representation of Algorithm 1 on an illustrative example. **a** Step 1, **b** Step 2, **c** Step 3, **d** Steps 4–6

a feasible point in a convex domain) is a well-known problem in convex optimization; a number of effective methods are presented in [10, 14]. In this paper, the convex feasibility problem is solved by minimizing the following quadratic penalty function

$$P_1(x) = \sum_{i=1}^m \max(c_i(x), 0)^2, \tag{2}$$

where  $P_1(x)$  is simply the quadratic penalty function used by quadratic penalty methods for solving NIPs (see Sect. 1). Since the  $c_i(x)$  are convex,  $P_1(x)$  is also convex; thus, if the feasible domain is nonempty, any local minimizer of  $P_1(x)$  is feasible. Note that the feasible domain is empty if the minimum of  $P_1(x)$  is greater than 0.

Note that  $P_1(x)$  is minimized via an iterative process which uses the Newton direction at each step, together with a line search, in order to guaranty the Armijo condition. Hessian modification [23] may be required to find the Newton’s direction, since  $P_1(x)$  is not strictly convex. An example of this procedure to find a feasible point from an initial infeasible point is illustrated in Fig. 1a.

The point  $x_f$  generated above is not necessarily an interior point of  $L$ . Note that the interior of  $L$  is nonempty if and only if the MFCQ (Mangasarian-Fromovitz constraint qualification) holds at  $x_f$  (see Proposition 3.2.7 in [19]). Checking the MFCQ at point  $x_f$  is equivalent to solving a linear programming problem [36], which either (a) generates a direction towards the interior of  $L$  from  $x_f$  (from which a point  $x_T$  on the interior of  $L$  is easily generated), or (b) establishes that the interior is empty. The optimization algorithm developed in this paper is valid only in case (a).

Starting from this interior feasible point  $x_T$ , the next step in the initialization identifies another feasible point that is, in a sense, far from all the boundaries of feasibility. This is achieved by minimizing the following logarithmic barrier function:

$$P_2(x) = - \sum_{i=1}^m \log(-c_i(x)). \tag{3}$$

It is easy to verify that  $P_2(x)$  is convex, and has a unique global minimum. Note that since initial point is an interior point,  $P_2(x)$  can be defined at it. This function is also minimized via Newton’s method; the line search at each step of this minimization is confined to the interior of the feasible domain. The outcome of this procedure, denoted  $X_0$ , is illustrated in Fig. 1b.

After finding the interior point  $X_0$  from the above procedure, a regular simplex<sup>2</sup>  $\Delta$  whose body center is  $X_0$  is constructed. Finding the coordinates of a regular simplex is well-known problem in computational geometry (see, e.g., §8.7 of [13]). The computational cost of finding a regular simplex is  $O(n^2)$ , which is insignificant compared with the rest of the algorithm.

Before continuing the initialization process, a new concept is introduced which is used a few times in this paper.

<sup>2</sup> A simplex is said to be regular if all of its edges are equal.

**Definition 1** The *prolongation* of point  $M$  from a feasible point  $O$  onto the feasibility boundary  $\partial L$ , is the unique point on the ray from  $O$  that passes through  $M$  which is on the boundary of feasibility. In order to find the prolongation of  $M$  from  $O$  on  $L$ , first the following 1D convex programming problem has to be solved.

$$\begin{aligned} & \max_{\alpha \in \mathbb{R}} \alpha, \\ & \text{subject to } c_i(O + \alpha(M - O)) \leq 0, \quad \forall 1 \leq i \leq m. \end{aligned} \tag{4}$$

Then,  $N = O + \alpha(M - O)$ , the prolongation point. It is easy to observe that  $N$  is located on the boundary of  $L$ . Since  $\alpha = 0$  is a feasible point for (4), it can be solved using the interior point method [21]. The computational cost of this subproblem is not significant if the computational cost of the constrained functions  $c_i(x)$  are negligible.

Based on the above definition, the initialization process is continued by prolongation of the vertices of the regular simplex from  $X_0$  onto  $L$ . As illustrated in Fig. 1c, the simplex so generated by this prolongation has a relatively large volume, and all of its vertices are feasible. The algorithm developed in the remainder of this paper incrementally increases the convex hull of the available datapoints towards the edges of the feasible domain itself. This process is generally accelerated if the volume of the initial feasible simple is maximized.

The feasible simplex generated above is not the feasible simplex of maximal volume. As illustrated in Fig. 1d, we next perform a simple iterative adjustment to the feasible vertices of this simplex to locally maximize its volume.<sup>3</sup> That is, denoting  $\Psi\{\Delta\{V_1, V_2, \dots, V_{m+1}\}\}$  as the volume of an  $m$ -dimensional simplex with corners  $\{V_1, V_2, \dots, V_{m+1}\}$ , we consider the following problem:

$$\begin{aligned} & \text{maximize } \Psi\{\Delta\{V_1, V_2, \dots, V_{n+1}\}\}, \\ & \text{where } \{V_1, V_2, \dots, V_{n+1}\} \in L. \end{aligned} \tag{5}$$

The problem of finding  $p > 2$  points in a convex subset of  $\mathbb{R}^2$  which maximizes its enclosing area is a well-known problem in the fields of interpolation, data compression, and robotic sensor networks. An efficient algorithm to solve this problem is presented in [46]. Note that (5) differs from the problem considered in [46] in three primary ways:

- (5) is defined in higher dimensions ( $n \geq 2$ );
- the boundary of the domain  $L$  is possibly nondifferentiable, whereas the problem in [46] assumed the boundary is twice differentiable;
- (5) is easier in the sense that a simplex is considered, not a general convex polyhedron.

As a result of these differences, a different strategy is applied to the present problem, as described below.

Consider  $V_1, V_2, \dots, V_{n+1}$  as the vertices of a simplex  $\Delta_x$ . The volume of  $\Delta_x$  may be written:

$$\Psi(\Delta_x) = \frac{\Psi(\Delta'_x)L_{V_k}}{n}, \tag{6}$$

where  $\Delta'_x$  is the  $n - 1$  dimensional simplex generated by all vertices except  $V_k$ ,  $H_k$  is the  $n - 1$  dimensional hyperplane containing  $\Delta'_x$ , and  $L_{V_k}$  is the perpendicular distance from the vertex  $V_k$  to the hyperplane  $H_k$ .

<sup>3</sup> The problem of globally maximizing the volume of a feasible simplex inside a convex domain is, in general, a nonconvex problem. We do not attempt to solve this global maximization, which is unnecessary in our algorithm.

By (6),  $L_{V_k}$  must be maximized to maximize the volume of the simplex if the other vertices are fixed. Furthermore, it is easily verified that the perpendicular distance of a point  $p$  to the hyperplane  $H_k$ , characterized by  $a_k^T x = b_k$ , is equal to  $|(a_k^T p - b_k)/(a_k^T a_k)|$ ; thus,

$$V_k = \operatorname{argmax}_{p \in L} |a_k^T p - b_k|. \tag{7}$$

Solving the optimization problem (7) is equivalent to finding the maximum of two convex optimization problems with linear objective functions. The method used to solve these two convex problems is the primal-dual-barrier method explained in detail in [25] and [22].

Based on the tools developed above, (5) is solved via the following procedure:

- Algorithm 1**
1. Find a point  $X$  in  $L = \{x | c_i(x) \leq 0, 1 \leq i \leq m\}$  by minimizing  $P_1(x)$  defined in (2); then, goes to the interior of  $L$  by checking the MFCQ condition.
  2. Starting from  $X$ ; then, goes to another feasible point  $X_0$  which is far from the constraint boundaries by minimizing  $P_2(x)$  defined in (3).
  3. Generate a uniform simplex with body center  $X_0$ ; denote the vertices of this simplex  $X_1, X_2, \dots, X_{n+1}$ .
  4. Determine  $V_1, V_2, \dots, V_{n+1}$  as the prolongations of  $X_1, X_2, \dots, X_{n+1}$  from  $X_0$  to the boundary of  $L$ .
  5. For  $k = 1$  to  $n + 1$ , modify  $V_k$  to maximize the distance from the hyperplane which passes through the other vertices by solving (7).
  6. If all modification at step 5 are small, stop the algorithm; otherwise repeat from 5.

**Definition 2** The simplex  $\Delta_i$  obtained via Algorithm 1, is referred to the *initial simplex*.

**Definition 3** For each vertex  $V_k$  of the initial simplex, the hyperplane  $H_k$ , characterized by  $a_k^T x = b_k$ , passes through all vertices of the initial simplex except  $V_k$ . Without loss of generality, the sign of the vector  $a_k$  may be chosen such that, for any point  $x \in L$ ,  $a_k^T x \leq a_k^T V_k$ . Now define the *enclosing simplex*  $\Delta_e$  as follows:

$$\Delta_e = \left\{ x \mid a_i^T x \leq a_i^T V_i, 1 \leq i \leq n + 1 \right\};$$

note that the feasible domain  $L$  is a subset of the enclosing simplex  $\Delta_e$ .

**Lemma 1** Consider  $V_1, V_2, \dots, V_{n+1}$  as the vertices of the initial simplex  $\Delta_i$ , and  $P_1, P_2, \dots, P_{n+1}$ , as those of the enclosing simplex  $\Delta_e$  given in Definition 3; then

$$P_k = \sum_{j=1}^{n+1} V_j - nV_k. \tag{8}$$

*Proof* Define  $P_k$  according to (8). Then, for all  $i \neq k$ ,

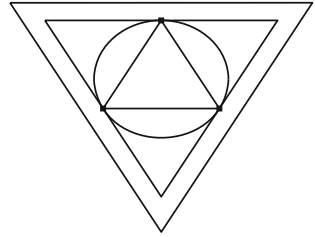
$$a_i^T P_k = \sum_{j=1}^{n+1} a_i^T V_j - na_i^T V_k.$$

According to Definition 3,  $a_i^T V_j = b_i, \forall j \neq i$ ; thus, above equation is simplified to:

$$a_i^T P_k = a_i^T V_i.$$

Thus,  $n$  of the independent constraints on the enclosing simplex given in Definition 3 are binding at  $P_k$ , and thus  $P_k$  is a vertex of  $\Delta_e$ . □

**Fig. 2** Illustration of the initial, enclosing and exterior simplices for an elliptical feasible domain



**Definition 4** Consider  $P_1, P_2, \dots, P_{n+1}$  as the vertices of the enclosing simplex  $\Delta_e$ , and  $O$  as its body center. The vertices of the exterior simplex  $\Delta_E$  are defined as follows:

$$E_i = O + \kappa(P_i - O) \tag{9}$$

where  $\kappa > 1$  is called the extending parameter.

The relative positions of the initial, enclosing, and exterior simplices are illustrated in Fig. 2. It follows from (8) and (9) that  $\sum V_i = \sum P_i = \sum E_i$ ; that is, the initial, enclosing, and exterior simplices all have the same body center, denoted  $O$ .

*Remark 1* In this paper, the following condition is imposed on the extending parameter  $\kappa$ :

$$\kappa \geq \frac{2 \max_{y \in L} \|y - O\|}{n \min_{1 \leq i \leq n+1} \|V_i - O\|}, \tag{10}$$

where  $O$  is the body center of the initial simplex, and  $\{V_1, V_2, \dots, V_{n+1}\}$  are the vertices of the initial simplex. In Sect. 4, it is seen that this condition is necessary to ensure convergence of the optimization algorithm presented in Sect. 3. In general, the value of  $\max_{y \in L} \{\|y - O\|\}$  is not known and is difficult to obtain<sup>4</sup>; however, the following upper bound is known,

$$\max_{y \in L} \|y - O\| \leq n \max_{1 \leq i \leq n+1} \|V_i - O\|.$$

Thus, by choosing  $\kappa$  as follows, (10) is satisfied:

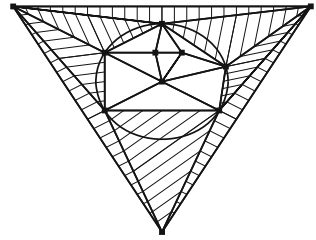
$$\kappa = \frac{2 \max_{1 \leq i \leq n+1} \|V_i - O\|}{\min_{1 \leq i \leq n+1} \|V_i - O\|}. \tag{11}$$

**Definition 5** The vertices of the initial simplex  $\Delta_i$ , together with its body center  $O$ , form the initial evaluation set  $S_E^0$ . The union of  $S_E^0$  and the vertices of the exterior simplex form the initial triangulation set  $S_T^0$ .

**Algorithm 2** After constructing  $S_E^0$  and  $S_T^0$ , a Delaunay triangulation  $\Delta^0$  over  $S_T^0$  is calculated. If the body center  $O$  and a vertex  $E$  of the exterior simplex  $\Delta_E$  are both located in any simplex of the triangulation  $\Delta^0$ ; then  $E'$  is defined as the intersection of segment  $OE$  with the boundary of  $L$ , and  $E'$  is added to both  $S_E^0$  and  $S_T^0$ , and the triangulation is updated. After at most  $n + 1$  modifications of this sort, the body center  $O$  is not located in the same simplex of the triangulation as any of the vertices of the exterior simplex. As a result, the number of points in  $S_E^0$  and  $S_T^0$  are at most  $2n + 3$ , and  $3n + 4$  respectively. The sets  $S_E^0$  and  $S_T^0$  are used to initialize the optimization algorithm presented in the following section.

<sup>4</sup> This is a maximization problem for a convex function in a convex compact domain. This problem is studied in [49].

**Fig. 3** Representation of the boundary (*hashed*) and interior (*non-hashed*) simplices in the Delaunay triangulation of a set of feasible evaluation points together with the three vertices of the exterior simplex



### 3 Description of the optimization algorithm

In this section, we present an algorithm to solve the optimization problem defined in (1). We assume that calculation of the constraint functions  $c_i(x)$  are computationally inexpensive compared to the function evaluations  $f(x)$ , and that the gradient and Hessian of  $c_i(x)$  are available.

Before presenting the optimization algorithm itself, some preliminary concepts are first defined.

**Definition 6** Consider  $\Delta^k$  as a Delaunay triangulation of a set of points  $S_T^k$ , which includes the vertices of the exterior simplex  $\Delta_E$ . As illustrated in Fig. 3, there are two type of simplices:

- (a) *boundary* simplices, which include at least one of the vertices of the exterior simplex  $\Delta_E$ , and
- (b) *interior* simplices, which do not include any vertices of the exterior simplex  $\Delta_E$ .

**Definition 7** For any boundary simplex  $\Delta_x \in \Delta^k$  which includes only one vertex of the exterior simplex  $\Delta_E$ , the  $n - 1$  dimensional simplex formed by those vertices of  $\Delta_x$  of which are not in common with the exterior simplex is called a *face* of  $\Delta_x$ .

**Definition 8** For each point  $x \in L$ , the *constraint function*  $g(x)$  is defined as follows:

$$g(x) = \max_{1 \leq j \leq m} \{c_j(x)\}. \tag{12}$$

For each point  $x$  in a face  $F \in \Delta^k$ , the *linearized constraint function* with respect to the face  $F$ ,  $g_L^F(x)$ , is defined as follows:

$$c_{j,L}^F(x) = \sum_{i=1}^n w_i c_j(V_i), \tag{13a}$$

$$g_L^F(x) = \max_{1 \leq j \leq m} \{c_{j,L}^F(x)\}, \tag{13b}$$

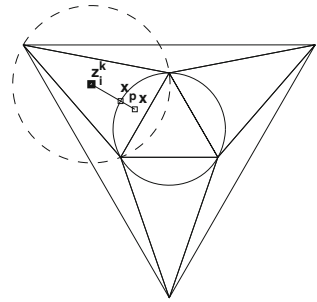
where the weights  $w_i \geq 0$  are defined such that

$$x = \sum_{i=1}^r w_i V_i \quad \text{with} \quad \sum_{i=1}^r w_i = 1.$$

**Definition 9** Consider  $x$  as a point which is located in the circumsphere of a boundary simplex in  $\Delta^k$ . Identify  $\Delta_i^k$  as that boundary simplex of  $\Delta^k$  which includes  $x$  in its circumsphere and which maximizes  $g(z_i^k)$ . The prolongation (see Footnote 1) of  $z_i^k$  from  $x$  onto the boundary of feasibility is called a *feasible boundary projection* of the point  $x$ , as illustrated in Fig. 4.



**Fig. 4** Illustration of a feasible boundary projection: the *solid circle* indicates the feasible domain,  $x$  is the initial point,  $z_i^k$  is the circumcenter of a boundary simplex  $\Delta_i^k$  containing  $x$ , the *dashed circle* indicates the corresponding circumsphere, and  $x_p$  is the feasible boundary projection



**Algorithm 3** As described in Sect. 2, prepare the problem for optimization by (a) executing Algorithm 1 to find the initial simplex  $\Delta_i$ , (b) identifying the exterior simplex  $\Delta_E$  as described in Definition 4, and (c) applying the modifications described in Algorithm 2 to remove the body center  $O$  from the boundary simplices. Then, proceed as follows:

0. Take  $S_E^0$  and  $S_T^0$  as the initial evaluation set and the initial triangulation set, respectively. Evaluate  $f(x)$  at all points in  $S_E^0$ . Set  $k = 0$ .
1. Calculate (or, for  $k > 0$ , update) an interpolating function  $p^k(x)$  that passes through all points in  $S_E^k$ .
2. Perform (or, for  $k > 0$ , incrementally update) a Delaunay triangulation  $\Delta^k$  over all points in  $S_T^k$ .
3. For each simplex  $\Delta_i^k$  of the triangulation  $\Delta^k$  which includes at most one vertex of the exterior simplex, define  $F_i^k$  as the corresponding face of  $\Delta^k$  if  $\Delta_i^k$  is a boundary simplex, or take  $F_i^k = \Delta_i^k$  itself otherwise. Then:
  - a. Calculate the circumcenter  $z_i^k$  and the circumradius  $r_i^k$  of the simplex  $F_i^k$ .
  - b. Define the **local uncertainty function**  $e_i^k(x)$  as

$$e_i^k(x) = \left(r_i^k\right)^2 - \left(x - z_i^k\right)^T \left(x - z_i^k\right). \tag{14}$$

- c. Define the **local search function**  $s_i^k(x)$  as follows: if  $\Delta_i^k$  is an interior simplex, take

$$s_i^k(x) = \frac{p^k(x) - y_0}{e_i^k(x)}; \tag{15}$$

otherwise, take

$$s_i^k(x) = \frac{p^k(x) - y_0}{-g_L^{F_i^k}(x)}, \tag{16}$$

where  $y_0$  is an estimate (provided by the user) for the value of the global minimum.

- d. Minimize the local search function  $s_i^k(x)$  in  $F_i^k$ .
4. If a point  $x$  at step 3 is found in which  $s^k(x) \leq y_0$ , then redefine the search  $s_k^k(x) = p^k(x)$  in all simplices, and take  $x_k$  as the minimizer of this search function in  $L$ ; otherwise, take  $x_k$  as the minimizer of the local minima identified in step 3d.
5. If  $x_k$  is not inside the circumsphere of any of the boundary simplices, define  $x'_k = x_k$ ; otherwise, define  $x'_k$  as the feasible boundary projection (see Definition 9) of  $x_k$ . Perform a function evaluation at  $x'_k$ , and take  $S_E^{k+1} = S_E^k \cup \{x'_k\}$  and  $S_T^{k+1} = S_T^k \cup \{x'_k\}$ .
6. Repeat from step 1 until  $\min_{x \in S^k} \{\|x'_k - x\|\} \leq \delta_{des}$ .

*Remark 2* Algorithm 3 generalizes the adaptive  $K$  algorithm developed in Part I of this work to convex domains. We could similarly extend the constant  $K$  algorithm developed in Part I by modifying (15) and (16) as follows: if  $\Delta_k^i$  is an interior simplex, take

$$s_i^k(x) = p^k(x) - K e_i^k(x);$$

otherwise, take

$$s_i^k(x) = p^k(x) + K g_L^{F_i^k}.$$

Such a modification might be appropriate if an accurate estimate of the Hessian of  $f(x)$  is available, but an accurate estimate of the lower bound  $y_0$  of  $f(x)$  is not.

*Remark 3* At each step of Algorithm 3, a feasible point  $x'_k$  is added to  $L$ . If Algorithm 3 is not terminated at finite  $k$ , since the feasible domain  $L$  is compact, the sequence of  $x'_k$  will have at least one convergent subsequence, by the Bolzano Weierstrass theorem. Since this subsequence is convergent, it is also Cauchy; therefore, for any  $\delta_{des} > 0$ , there are two integers  $k$  and  $m < k$  such that  $\|x'_m - x'_k\| \leq \delta_{des}$ . Thus,  $\min_{x \in S^k} \{\|x'_k - x\|\} \leq \delta_{des}$ ; thus, the termination condition will be satisfied at step  $k$ . As a result, Algorithm 3 will terminate in a finite number of iterations.

**Definition 10** At each step of Algorithm 3, and for each point  $x \in L$ , there is a simplex  $\Delta_i^k \in \Delta^k$  which includes  $x$ . The *global uncertainty function*  $e^k(x)$  is defined as  $e^k(x) = e_i^k(x)$ . It is shown in Part I (Lemmas 3 and 4) that  $e^k(x)$  is continuous and Lipschitz with Lipschitz constant of  $r_{max}^k$ , the maximum circumradius of  $\Delta^k$ , and that the Hessian of  $e^k(x)$  inside each simplex, and over each face  $F$  of each simplex, is  $-2I$ .

There are three principle difference between Algorithm 3 above and the corresponding algorithm proposed in Part I for the linearly-constrained problem:

- In the initialization, instead of calculating the objective function at all of the vertices of feasible domain (which is not possible if the constraints are nonlinear), the present algorithm is instead initialized with between  $n + 2$  and  $2n + 3$  function evaluations.
- The local search function is modified in the boundary simplices.
- The feasible *boundary* projections used in the present work are analogous to (but slightly different from) Algorithm 3 of Part I, which applies one or more feasible *constraint* projections.

### 3.1 Minimizing the search function

As with Algorithm 2 of Part 1 of this work, the most expensive part of Algorithm 3, separate from the function evaluations themselves, is step 3 of the algorithm. The cost of this step is proportional to the total number of simplices  $S$  in the Delaunay triangulation. As derived in [38], a worst-case upper bound for the number of simplices in a Delaunay triangulation is  $S \sim O(N^{\frac{n}{2}})$ , where  $N$  is the number of vertices and  $n$  is the dimension of the problem. As shown in [17, 18], for vertices with a uniform random distribution, the number of simplices is  $S \sim O(N)$ . This fact limits the present algorithm to be applicable only for relatively low DOF problems (say,  $n < 10$ ). In practice, the most limiting part of step 3 is the memory requirement imposed by the computation of the Delaunay triangulations. Thus, the present algorithm itself is applicable only to those problems for which Delaunay triangulations can be performed amongst the datapoints.

In this section, we describe some details and facts that simplifies certain other aspects of step 3 (besides the Delaunay triangulations). In general, there are two type of simplices in  $\Delta^k$  whose definition of the search function is different.

In the interior simplices  $\Delta_i^k$ , the function  $s_i^k(x) = \frac{p^k(x)-y_0}{e_i^k(x)}$  has to be minimized in  $\Delta_i^k$ . One important property of  $e^k(x)$  which makes this problem easier is  $e^k(x) = \max_{j \in S} e_j^k(x)$  (Lemma 2 in Part I), and as it shown in Lemma 5 of Part I, if  $s^k(x)$  is minimized in  $L$  rather than  $\Delta_i^k$ , the position of point  $x_k$  would not be changed. Another important issue is the method of initialization for the minimization of  $s_i^k(x)$ . To accomplish this, we choose a point  $\hat{x}_i^k$  which maximizes  $e_i^k(x)$  as the initial point for the minimization algorithm. Note that, if the circumcenter of this simplex is included in this simplex, the circumcenter is the point that maximizes  $e_i^k(x)$ ; otherwise, this maximization is a simple quadratic programming problem. The computational cost of calculating  $\hat{x}_i^k$  is similar to that of calculating  $x_c^k$ . After the initializations described above, the subsequent minimizations are performed using Newton’s method with Hessian modification via modified Cholesky factorization (see [23]) to minimize  $s_i^k(x)$  within  $L$ . It will be shown (see Theorem 1) that performing Newton’s method is actually not necessary to guaranty the convergence of Algorithm 3, and having a point whose value  $s^k(x)$  is less than the initial points is sufficient to guaranty the convergence; however, performing Newton minimizations at this step can improve the speed of convergence. In practice, we will perform Newton minimizations only in a few simplices whose initial points have small values for  $s^k(x)$ .

For the boundary simplices  $\Delta_i^k$ , the function  $s_i^k(x)$  is defined only over the face  $F_i^k$  of  $\Delta_i^k$ . The minimization in each boundary simplex is initialized at the point  $\tilde{x}_i^k$  which maximizes the piecewise linear function  $-g_L^{F_i^k}(x)$ ; this maximization may be written as a linear programming problem with the method described in §1.3, p. 7, in [11]. Similarly, it can be seen that initialization with these points is enough to guaranty the convergence; thus, it is not necessary to perform the Newton method after this initialization. In our numerical implementation, the best of these initial points are considered for the simulation; however, the implementation of Newton minimizations on these faces could further improve the speed of convergence.

*Remark 4* As in Algorithm 4 of Part I, since Newton’s method doesn’t always converge to a global minimum,  $x_k$  is not necessarily a global minimizer of  $s^k(x)$ . However, the following properties are guaranteed:

$$\begin{aligned} \text{if } s^k(x) = p^k(x), \text{ then } p^k(x_k) &\leq y_0; & (17a) \\ \text{otherwise } s^k(x_k) &\leq s_j^k(\hat{x}_j^k) \quad \forall \Delta_j^k \in \Delta^k, & (17b) \\ \text{and } s^k(x_k) &\leq s_j^k(\tilde{x}_j^k) \quad \forall F_j^k \in \Delta^k. & (17c) \end{aligned}$$

Recall that  $\hat{x}_j^k$  is the maximizer of  $e_j^k(x)$  in the interior simplex  $\Delta_j^k \in \Delta^k$ , and  $\tilde{x}_j^k$  is the maximizer of  $-g_L^{F_j^k}(x)$  over the face  $F_j^k$  of the boundary simplex  $\Delta_j^k \in \Delta^k$ . These properties are all that are required by Theorem 1 in order to establish convergence.

### 4 Convergence analysis

In this section, the convergence properties of Algorithm 3 are analyzed. The structure of the convergence analysis is similar to that in Part I.

In this section, the following conditions for the function of interest,  $f(x)$ , the constraints  $c_i(x)$ , and the interpolating functions  $p^k(x)$  are imposed.

**Assumption 1** The interpolating functions  $p^k(x)$ , for all  $k$ , are Lipschitz with the same Lipschitz constant  $L_p$ .

**Assumption 2** The function  $f(x)$  is Lipschitz with Lipschitz constant  $L_f$ .

**Assumption 3** The individual constraint functions  $c_i(x)$  for  $1 \leq i \leq m$  are Lipschitz with Lipschitz constant  $L_c$ .

**Assumption 4** A constant  $K_{pf}$  exists<sup>5</sup> in which

$$K_{pf} > \lambda_{\max} \left( \nabla^2 (f(x) - p^k(x)) / 2 \right), \quad \forall x \in L \text{ and } k > 0.$$

**Assumption 5** A constant  $K_f$  exists in which

$$K_f > \lambda_{\max} \left( \nabla^2 (f(x)) / 2 \right), \quad \forall x \in L.$$

**Assumption 6** A constant  $K_g$  exists in which

$$K_g > \lambda_{\max} \left( \nabla^2 (c_i(x)) / 2 \right), \quad \forall x \in L.$$

Before analyzing the convergence of Algorithm 3, some preliminary definitions and lemmas are needed.

**Definition 11** According to the construction of the initial simplex, its body center is an interior point in  $L$ ; thus, noting (12),  $g(O) < 0$ . Thus, the quantity  $L_O = \max_{y \in L} \|y - O\| / (-g(O))$  is a bounded positive real number.

**Lemma 2** Each vertex of the boundary simplices of  $\Delta^k$ , for all steps of Algorithm 3, is either a vertex of the exterior simplex or is located on a boundary of  $L$ .

*Proof* We will prove this lemma by induction on  $k$ . According to Algorithm 2,  $O$  is not in any boundary simplex of  $\Delta^0$ , and all other points of  $S_E^k$  are on the boundary of  $L$ ; therefore, the lemma is true for the base case  $k = 0$ . Assuming the lemma is true for the case  $k - 1$ , we now show that it follows that the lemma is also true for the case  $k$ .

At step  $k$  of Algorithm 3, we add a point  $x'_k$  to the triangulation set  $S_T^k$ . As described step 5 of Algorithm 3, this point arises from one of two possible cases.

In the first case,  $x_k$  is not located in the circumsphere of any boundary simplex, and a feasible boundary projection is not performed; as a result,  $x'_k = x_k$  is an interior point of  $L$ . In this case, the incremental update of the Delaunay triangulation at step  $k$  does not change the boundary simplices of  $\Delta^{k-1}$  (see, e.g., section 2.1 in [12]), and thus the lemma is true in this case.

In the other case,  $x_k$  is located in the circumsphere of a boundary simplex, and a feasible boundary projection is performed; thus,  $x'_k$  is on the boundary of  $L$ . Consider  $\Delta_x$  as one of the new boundary simplices which is generated at step  $k$ . By construction,  $x_k$  is a vertex of  $\Delta_x$ . Define  $F$  as the  $n - 1$  dimensional face of  $\Delta_x$  which does not include  $x_k$ . Based on the incremental construction of the Delaunay triangulation,  $F$  is a face of another simplex  $\Delta'_x$  at step  $k - 1$ . Since  $\Delta_x$  is a boundary simplex,  $F$  includes a vertex of the exterior simplex; thus,  $\Delta'_x$  is a boundary simplex in  $\Delta^{k-1}$ ; thus, each vertex of  $F$  is either a boundary point or a vertex of the exterior simplex; moreover,  $x'_k$  is a boundary point. As a result,  $\Delta_x$  satisfies the lemma. For those boundary simplices which are not new, the lemma is also true, by the induction hypothesis. Thus, the lemma is also true in this case. □

<sup>5</sup> The maximum eigenvalue is denoted  $\lambda_{\max}(\cdot)$ .

**Definition 12** For any point  $x \in L$ , the *interior projection* of  $x$ , denoted by  $x_I$ , is defined as follows: If  $x$  is located inside or on the boundary of an interior simplex, put  $x_I = x$ ; otherwise,  $x_I$  is taken as the point of intersection, closest to  $x$ , of the line segment  $Ox$  with the boundary of the union of the interior simplices. Thus,  $x_I$  is located on a face of the triangulation.

**Lemma 3** Consider  $x$  as a point in  $L$  which is not located in the union of the interior simplices at step  $k$  of Algorithm 3. Define  $x_I$  as the interior projection of  $x$  (see Definition 12). Then,

$$p^k(x_I) - f(x) \leq e^k(x_I)\{K_{pf} + L_f K_g L_O\} - L_f L_O g_L^F(x_I), \tag{18}$$

where  $e^k(x)$  is the uncertainty function,  $g_L^F(x)$  is the linearized constraint function (see Definition 8) with respect to a face  $F$  which includes  $x_I$ , and  $L_f, K_{pf}, K_g,$  and  $L_O$  are defined in Assumptions 2, 4 and 6 and Definition 11, respectively.

*Proof* By Definition 12,  $x_I$  is a point on the line segment  $Ox$ ; thus,

$$x_I = x \frac{\|O - x_I\|}{\|O - x\|} + O \frac{\|x - x_I\|}{\|O - x\|}.$$

Define  $\{V_1, V_2, \dots, V_n\}$  as the vertices of  $F$ , and  $G_j^k(x) = c_{j,L}^F(x) - c_j(x) - K_g e^k(x)$ , where  $c_{j,L}^F$  is the linear function defined in (13a). First we will show that  $G_j^k(x)$  is strictly convex in  $F$ . Since  $c_{j,L}^F(x)$  is a linear function of  $x$  in  $F$ , and  $\nabla^2 e^k(x) = -2I$  (see (14)), it follows that

$$\nabla^2 G_j^k(x) = -\nabla^2 \{c_j(x)\} + 2K_g I. \tag{19}$$

According to Assumption 6  $\nabla^2 G_j^k(x) > 0$  in  $F$ ; thus, it is strictly convex; therefore, its maximum in  $F$  is located at one of the vertices of  $F$ . Furthermore, by construction,  $G_j^k(V_i) = 0$ ; thus,  $G_j^k(x_I) \leq 0$ , and

$$\begin{aligned} c_j(x_I) &\geq c_{j,L}^F(x_I) - K_g e^k(x_I) \quad \forall 1 \leq j \leq m, \\ g(x_I) &\geq g_L^F(x_I) - K_g e^k(x_I), \end{aligned} \tag{20}$$

where  $g(x)$  is defined in Definition 8. Since the  $c_i(x)$  are convex,  $g(x)$  is also convex; thus,

$$g(x_I) \leq g(x) \frac{\|O - x_I\|}{\|O - x\|} + g(O) \frac{\|x - x_I\|}{\|O - x\|}. \tag{21}$$

Since  $x \in L, g(x) \leq 0$ . Since  $O$  is in the interior of  $L, g(O) < 0$ ; from (20) and (21), it thus follows that

$$\begin{aligned} g_L^F(x_I) - K_g e^k(x_I) &\leq g(O) \frac{\|x - x_I\|}{\|O - x\|}, \\ \|x - x_I\| &\leq \frac{\|x - O\|}{-g(O)} \{K_g e^k(x_I) - g_L^F(x_I)\}. \end{aligned}$$

By Definition 11, this leads to

$$\|x - x_I\| \leq L_O \left\{ K_g e^k(x_I) - g_L^F(x_I) \right\}. \tag{22}$$

Now define  $T(x) = p^k(x) - f(x) - K_{pf} e^k(x)$ . By Assumption 4 and Definition 10, similar to  $G_j^k(x), T(x)$  is also strictly convex in  $F$ , and  $T(V_i) = 0$ ; thus,

$$p^k(x_I) - K_{pf} e^k(x_I) \leq f(x_I). \tag{23}$$

Furthermore,  $L_f$  is a Lipchitz constant for  $f(x)$ ; thus,

$$f(x_I) - f(x) \leq L_f \|x - x_I\|. \tag{24}$$

Using (22),(23), and (24), (18) is satisfied. □

*Remark 5* If  $x$  is a point in an interior simplex, it is easy to show [as in the derivation of (23)] that (18) is modified to

$$p^k(x) - f(x) \leq K_{pf} e^k(x). \tag{25}$$

**Lemma 4** Consider  $F$  as a face of the union of the interior simplices at step  $k$  of Algorithm 3,  $x$  as a point on  $F$ , and  $g_F^L(x)$  as the linearized constraint function on  $F$  as defined in Definition (8). Then,

$$-g_F^L(x) \leq L_c \sqrt{e^k(x)}. \tag{26}$$

*Proof* By Assumption 3, the  $c_i(x)$  are Lipchitz with constant  $L_c$ . Thus, by (13a), the  $c_{i,F}^L(x)$  are Lipchitz with the same constant. Furthermore, the maximum of a finite set of Lipchitz functions with constant  $L_c$  is Lipchitz with the same constant (see Lemma 2.1 in [26]). Define  $\{V_1, V_2, \dots, V_n\}$  as the vertices of  $F$ ; then, by Lemma 2,  $V_i$  is on the boundary of  $L$ ; thus,  $g^L(V_i) = 0$  for all  $1 \leq i \leq n$ , and

$$-g_F^L(x) \leq L_c \min_{1 \leq i \leq n} \{\|x - V_i\|\}. \tag{27}$$

Now define  $\{w_0, w_1, \dots, w_n\}$  as the weights of  $x$  in  $F$  (that is,  $x = \sum_{j=0}^n w_j V_j$  where  $w_i \geq 0$  and  $\sum_{j=0}^n w_j = 1$ ),  $z$  as the circumcenter of  $F$ , and  $r$  as the circumradius of  $F$ . Then, for each  $j$ ,

$$r^2 = \|V_j - z\|^2 = \|V_j - x\|^2 + \|x - z\|^2 + 2(V_j - x)^T(x - z).$$

Multiplying the above equations by  $w_j$  and taking the sum over all  $j$ , noting that  $\sum_{j=0}^n w_j = 1$ , it follows that

$$r^2 = \sum_{j=0}^n w_j \|V_j - x\|^2 + \|x - z\|^2 + 2 \sum_{j=1}^n w_j (V_j - x)^T(x - z).$$

Since  $\sum_{j=0}^n w_j (V_j - x) = 0$ , this simplifies to

$$r^2 = \sum_{j=0}^n w_j \|V_j - x\|^2 + \|x - z\|^2,$$

$$\sum_{j=0}^n w_j \|V_j - x\|^2 = r^2 - \|x - z\|^2,$$

$$\sum_{j=0}^n w_j \|V_j - x\|^2 = e^k(x), \tag{28}$$

$$e^k(x) \geq \min_{0 \leq i \leq n} \{\|x - V_i\|\}^2. \tag{29}$$

Since both side of (29) are positive numbers, we can take square root of the both sides; moreover,  $L_c$  is a positive number.

$$L_c \sqrt{e^k(x)} \geq L_c \min_{1 \leq i \leq n} \{\|x - V_i\|\}. \tag{30}$$

Using (27) and (30), (26) is satisfied. □

Note that, by (28), the global uncertainty function  $e^k(x)$  defined in (14) and Definition 10 is simply the weighted average of the squared distance of  $x$  from the vertices of the simplex that contains  $x$ .

The other lemma which is essential for analysis the convergence of Algorithm 3, is that the maximum circumradius of  $\Delta^k$  is bounded.

**Lemma 5** Consider  $\Delta^k$  as a Delaunay triangulation of a set of triangulation points  $S_T^k$  at step  $k$ . The maximum circumradius of  $\Delta^k$ ,  $r_{max}^k$  is bounded as follows:

$$r_{max}^k \leq \kappa L_1 \sqrt{1 + \left(\kappa \frac{L_1}{2(\kappa - 1)\delta_O}\right)^2}, \tag{31}$$

where  $L_1$  is the maximum edge length of the enclosing simplex  $\Delta_e$ ,  $\kappa$  is the extending parameter (Definition 4), and  $\delta_O$  is the minimum distance from  $O$  (the body center of the exterior simplex) to the boundary of  $\Delta_e$ .

*Proof* Consider  $\Delta_x$  as a simplex in  $\Delta^k$  which has the maximum circumradius. Define  $z$  as the circumcenter of  $\Delta_x$ , and  $x$  as a vertex of  $\Delta_x$  which is not a vertex of the exterior simplex. If  $z$  is located inside of the exterior simplex  $\Delta_E$ , then,

$$r_{max}^k = \|z - x\| \leq L_1 \kappa, \tag{32}$$

which shows lemma in this case. If  $z$  is located outside of  $\Delta_E$ , then define  $z_p$  as the point closest to  $z$  in the exterior simplex. That is,

$$\begin{aligned} z_p &= \operatorname{argmin}_{y \in \Delta_E} \|z - y\| \\ \text{subject to } a_{i,E}^T y &\leq b_{i,E}, \forall 1 \leq i \leq n + 1, \end{aligned} \tag{33}$$

where  $a_{i,E}^T y \leq b_{i,E}$  defines the  $i$ 'th face of the exterior simplex. Define  $A_a(Z_p)$  as the set of active constraints at  $Z_p$  in the constraints of (33), and  $v$  as a vertex of the exterior simplex in which all constraints in  $A_a(z_p)$  are active at  $z_p$ ; then, using the optimality condition at  $Z_p$ , it follows that

$$\begin{aligned} (z - z_p)^T (z_p - v) &= 0, \\ \|z - v\|^2 &= \|z - z_p\|^2 + \|v - z_p\|^2. \end{aligned} \tag{34}$$

Now consider  $p$  as the intersection of the line segment  $zx$  with the boundary of the exterior simplex; then,

$$\|z - x\| = \|z - p\| + \|p - x\|. \tag{35}$$

Furthermore, by construction,

$$\|z - z_p\| \leq \|z - p\|. \tag{36}$$

Since  $p$  is on the exterior simplex  $\Delta_E$ , and  $x$  is inside of the enclosing simplex  $\Delta_e$ , it follows that

$$\|p - x\| \geq (\kappa - 1)\delta_O. \tag{37}$$

Note that  $x$  is a vertex of the simplex  $\Delta_x$ , and  $v$  is a point in  $S^k$ . Since the triangulation  $\Delta^k$  is Delaunay,  $v$  is located outside of the interior of the circumsphere of  $\Delta_x$ . Recall,  $z$  is the circumcenter of  $\Delta_x$ ; thus,

$$R_x = \|x - z\| \leq \|z - v\|, \tag{38}$$

where  $R_x$  is the circumcenter of  $L$ . Using (34), (35) and (38) it follows that

$$\|z - p\|^2 + \|p - x\|^2 + 2\|z - p\|\|p - x\| \leq \|z - z_p\|^2 + \|v - z_p\|^2. \tag{39}$$

By (36), (37) and (39), it follows that

$$2(\kappa - 1)\delta_O \|z - z_p\| \leq \|z - z_p\|^2.$$

Note that  $\|v - z_p\| \leq \kappa L_1$ ; thus,

$$\|z - z_p\| \leq \kappa^2 \frac{L_1^2}{2(\kappa - 1)\delta_O}. \tag{40}$$

Combining (34) and (40), noting that  $r_{max}^k \leq \|z - v\|$ , (31) is satisfied. □

The final lemma which is required to establish the convergence of Algorithm 3, given below, establishes an important property of the feasible boundary projection.

**Lemma 6** Consider  $\Delta^k$  as a Delaunay triangulation of Algorithm 3 at step  $k$ , and  $x'_k$  as the feasible boundary projection of  $x_k$ . If the extending parameter  $\kappa$  satisfies (10), then, for any point  $V \in S^k$ ,

$$\frac{\|V - x'_k\|}{\|V - x_k\|} \geq \frac{1}{2}. \tag{41}$$

*Proof* If a feasible boundary projection is not performed at step  $k$ ,  $x'_k = x_k$ , and (41) is satisfied trivially. Otherwise, consider  $\Delta_i^k$  in  $\Delta^k$  as the boundary simplex, with circumcenter  $Z$  and circumradius  $R$ , which is used in the feasible boundary projection (see Definition 9). First, we will show that, for a value of  $\kappa$  that satisfies (10),  $Z$  is not in  $L$ . This fact is shown by contradiction; thus, first assume that  $Z$  is in  $L$ . Define  $A$  as a shared vertex of  $\Delta_i^k$  and the exterior simplex, and  $V_1, V_2, \dots, V_{n+1}$  as the vertices of the initial simplex  $\Delta_i$  (see Definition 2). Since the body center of the exterior simplex  $O \in S_T^k$ , and  $\Delta^k$  is Delaunay,

$$\begin{aligned} \|O - Z\| &\geq \|A - Z\|, \\ \|O - Z\| &\geq \min_{y \in L} \{\|A - y\|\}, \\ \|O - Z\| + \|O - y\| &\geq \min_{y \in L} \{\|A - y\|\} + \|O - y\|, \\ 2 \max_{y \in L} \{\|O - y\|\} &\geq \|O - A\|, \\ \|O - A\| &\geq n\kappa \min_{1 \leq j \leq n+1} \{\|O - V_j\|\}, \\ 2 \max_{y \in L} \{\|y - O\|\} &\geq n\kappa \min_{1 \leq j \leq n+1} \{\|V_j - O\|\}. \end{aligned} \tag{42}$$

However, (42) is in contradiction with (10); thus,  $Z$  is not in  $L$ ; therefore, by Definition 9,  $x'_k$  is on the line segment  $Zx_k$ .

Now we will show (41). Consider  $V_p$  as the orthogonal projection of  $V$  onto the line through  $x_k, x'_k$  and  $Z$ ; then, we may write  $V_p = \beta x_k$  and, by construction:

$$\begin{aligned} \|V - x_k\|^2 &= \|V - V_p\|^2 + \|V_p - x_k\|^2, \\ \|V - x'_k\|^2 &= \|V - V_p\|^2 + \|V_p - x'_k\|^2, \\ \|V - Z\|^2 &= \|V - V_p\|^2 + \|V_p - Z\|^2. \end{aligned} \tag{43}$$



Since  $Z$  is outside of  $L$ ,  $x_k$  is in  $L$ ,  $x'_k$  is on the boundary of  $L$ , and they are collinear ; then, there is a real number  $0 \leq \alpha \leq 1$ , such that  $x'_k = Z + \alpha (x_k - Z)$ , which leads to

$$\begin{aligned} \|V_p - x_k\|^2 &= \|V_p - Z\|^2 + 2(Z - V_p)^T(x_k - Z) + 2\|x_k - Z\|^2, \\ \|V_p - x'_k\|^2 &= \|V_p - Z\|^2 + 2\alpha(Z - V_p)^T(x_k - Z) + 2\alpha^2\|x_k - Z\|^2. \end{aligned}$$

Therefore, above equations are simplified to.

$$\begin{aligned} \|V - x_k\|^2 &= \|V - Z\|^2 + 2(Z - V_p)^T(x_k - Z) + 2\|x_k - Z\|^2, \\ \|V - x'_k\|^2 &= \|V - Z\|^2 + 2\alpha(Z - V_p)^T(x_k - Z) + 2\alpha^2\|x_k - Z\|^2. \end{aligned} \tag{44}$$

By defining  $\beta = \frac{(Z - V_p)^T(x_k - Z)}{\|x_k - Z\|\|V - Z\|}$  and  $\gamma = \frac{\|x_k - Z\|}{\|V - Z\|}$ ; the above equation is simplified to:

$$\frac{\|V - x'_k\|^2}{\|V - x_k\|^2} = \frac{1 + 2\alpha\beta\gamma + \alpha^2\gamma^2}{1 + 2\beta\gamma + \gamma^2}. \tag{45}$$

By construction,  $x_k$  is inside the circumsphere of  $\Delta_i^k$ . Moreover,  $\Delta^k$  is a Delaunay triangulation for  $S^k$ , and  $V \in S^k$ ; thus,  $V$  is outside the interior of circumsphere of  $\Delta_i^k$ . As a result,  $\|Z - x_k\| \leq \|Z - V\|$ . Moreover,  $\gamma$  is trivially positive; thus,  $0 \leq \gamma \leq 1$ . Now we identify lower and upper bounds for  $\beta$ :

$$\begin{aligned} -\|Z - V_p\|\|x_k - Z\| &\leq (Z - V_p)^T(x_k - Z) \leq \|Z - V_p\|\|x_k - Z\|, \\ \|Z - V_p\| &\leq \|Z - V\|, \\ -\|Z - V\|\|x_k - Z\| &\leq (Z - V_p)^T(x_k - Z) \leq \|Z - V\|\|x_k - Z\|, \\ -1 &\leq \beta \leq 1. \end{aligned}$$

The right hand side of (45) is a function of  $\alpha, \beta, \gamma$ . Moreover, it is shown that  $0 \leq \alpha \leq 1, -1 \leq \beta \leq 1$ , and  $0 \leq \gamma \leq 1$ . In order to show (41), It is sufficient to prove that the minimum of this three dimensional function, denoted by  $\Gamma(\alpha, \beta, \gamma)$ , in the box that characterized its variable is  $\frac{1}{4}$ .

The function  $\Gamma$  is a fractional linear function of  $\beta$ , thus,

$$\begin{aligned} \Gamma(\alpha, \beta, \gamma) &\geq \min\{\Gamma(\alpha, -1, \gamma), \Gamma(\alpha, 1, \gamma)\}, \\ \Gamma(\alpha, 1, \gamma) &= \left(\frac{1 + \alpha\gamma}{1 + \gamma}\right)^2 \geq \frac{1}{(1 + \gamma)^2} \geq \frac{1}{4}, \\ \Gamma(\alpha, -1, \gamma) &= \left(\frac{1 - \alpha\gamma}{1 - \gamma}\right)^2 \geq \left(\frac{1 - \gamma}{(1 - \gamma)^2}\right)^2 \geq 1, \\ \Gamma(\alpha, \beta, \gamma) &\geq \frac{1}{4}. \end{aligned} \tag{46}$$

Using (45) and (46), (41) is verified. □

Finally, using the above lemmas, we can prove the convergence of Algorithm 3 to the global minimum in the feasible domain  $L$ .

**Theorem 1** *If Algorithm 3 is terminated at step  $k$ , and  $y_0 \leq f(x^*)$ , then there are real bounded numbers  $A_1, A_2, A_3$  such that*

$$\min_{z \in S^k} f(z) - f(x^*) \leq A_1 \delta_{des} + A_2 \sqrt{\delta_{des}} + A_3 \sqrt[4]{\delta_{des}}. \tag{47}$$

*Proof* Define  $S_E^k, S_T^k, r_{\max}^k$ , and  $L_2^k$  as the evaluation set, the triangulation set, the maximum circumradius of  $\Delta^k$ , and the maximum edge length of  $\Delta^k$ , respectively, where  $\Delta^k$  is a Delaunay triangulation of  $S_T^k$ . Define  $x_k$  as the outcome of Algorithm 3 though step 4, which at step  $k$  satisfies (17), and  $x'_k$  as the feasible boundary projection of  $x_k$  on  $L$ .

Define  $y_1 \in S_E^k$  as the point which minimizes  $\delta = \min_{x \in S_E^k} \|x - x_k\|$ , and the parameters  $\{A, B, C, D, E\}$  as follows:

$$A = \max\{K_f + L_f K_g L_O, L_f L_O\}, \tag{48a}$$

$$B = \max\{K_{pf} + L_f K_g L_O, L_f L_O\}, \tag{48b}$$

$$C = \max\left\{4 A r_{\max}^k, L_p + 4 B r_{\max}^k\right\}, \tag{48c}$$

$$D = 2 \max\left\{A L_c \sqrt{2r_{\max}^k}, B L_c \sqrt{2r_{\max}^k}, \right. \\ \left. \times \sqrt{L_c L_2^k L_p A r_{\max}^k}\right\}, \tag{48d}$$

$$E = \sqrt{2 L_c L_2^k L_p A} \sqrt[4]{2r_{\max}^k}. \tag{48e}$$

We will now show that

$$\min_{z \in S^k} f(z) - f(x^*) \leq C\delta + D\sqrt{\delta} + E\sqrt[4]{\delta}, \tag{49}$$

where  $x^*$  is a global minimizer of  $f(x^*)$ .

During the iterations of Algorithm 3, there two possible cases for  $s^k(x)$ . The first case is when  $s^k(x) = p^k(x)$ . In this case, via (17a),  $p^k(x_k) \leq y_0$ , and therefore  $p^k(x_k) \leq f(x^*)$ . Since  $y_1 \in S_E^k$ , it follows that  $p^k(y_1) = f(y_1)$ . Moreover,  $L_p$  is a Lipchitz constant for  $p^k(x)$ ; therefore,

$$p^k(y_1) - p^k(x_k) \leq L_p \delta, \\ f(y_1) - p^k(x_k) \leq L_p \delta, \\ f(y_1) - f(x^*) \leq L_p \delta, \\ \min_{z \in S^k} f(z) - f(x^*) \leq L_p \delta.$$

which shows that (49) is true in this case.

The other case is when  $s^k(x) = (p^k(x) - y_0)/e^k(x)$  in the interior simplices, and  $s^k(x) = (p^k(x) - y_0)/(-g_L^F(x))$  on the faces of  $\Delta^k$ . For this case, we will show that (49) is true when  $x^*$  is not in an interior simplex. When  $x^*$  is in an interior simplex, (49) can be shown in an analogous manner.

Define  $F_i^k$  as a face in  $\Delta^k$  which includes  $x_I^*$ , the interior projection (see Definition 9) of  $x^*$ , and  $\Delta_i^k$  as an interior simplex which includes  $x_I^*$ . Define  $\hat{x}_i^k$  and  $\tilde{x}_i^k$  as the points which maximize  $e_i^k(x)$  and  $-g_L^{F_i^k}(x)$  in  $\Delta_i^k$  and  $F_i^k$ , respectively, where  $e_i^k(x)$  and  $-g_L^{F_i^k}(x)$  are the local uncertainty function in  $\Delta_i^k$  and the linearized constraint function (see Definition 8) in  $F_i^k$ .

According to Lemma 3 and (48b),

$$p^k(x_I^*) - f(x^*) \leq B e^k(x_I^*) - B g_L^{F_i^k}(x_I^*), \\ p^k(x_I^*) - f(x^*) \leq B e_i^k(\hat{x}_i^k) - B g_L^{F_i^k}(\tilde{x}_i^k). \tag{50}$$

Further, by replacing the linear interpolation  $p^k(x)$  with the linear function  $L_i^k(x)$ , in which  $L_i^k(x) = f(x)$  at the vertices of  $F$ , and using  $\min_{z \in S^k} f(z) \leq L^k(x_j^*)$  and the fact that the Hessian of  $L_i^k(x)$  is zero, and noting (48a), (50) may be modified to:

$$\min_{z \in S_E^k} \{f(z)\} - f(x^*) \leq A e_i^k(\hat{x}_i^k) - A g_L^{F_i^k}(\hat{x}_i^k). \tag{51}$$

If the search function at  $x_k$  is defined by (15), take  $\hat{e}_k = e^k(x_k)$ ; on the other hand, if it is defined by (16), take  $\hat{e}_k = -g_L^{F_j^k}(x_k)$  where  $F_j^k$  is the face of  $\Delta^k$  which includes  $x_k$ .

Since  $2r_{\max}^k$  is a Lipchitz constant for  $e^k(x)$  (see Lemma 4 in Part I), by using Lemma 4 above, it follows

$$\begin{aligned} \hat{e}_k &\leq \max \left\{ 2r_{\max}^k \delta, L_c \sqrt{2r_{\max}^k} \delta \right\}, \\ \hat{e}_k &\leq 2r_{\max}^k \delta + L_c \sqrt{2r_{\max}^k} \delta \end{aligned} \tag{52}$$

If  $e_i^k(\hat{x}) - g_L^{F_i^k}(\hat{x}_i^k) \leq 2\hat{e}_k$ , then by (51) and (52), (49) is satisfied. Otherwise, dividing  $f(x^*) - y_0 \geq 0$  by this expression with the opposite inequality,

$$\frac{2f(x^*) - 2y_0}{e_i^k(\hat{x}) - g_L^{F_i^k}(\hat{x}_i^k)} < \frac{f(x^*) - y_0}{\hat{e}_k}. \tag{53}$$

Using (17b), (17c) and (53)<sup>6</sup>

$$\begin{aligned} \frac{p^k(x_k) - y_0}{\hat{e}_k} &\leq \frac{p^k(\hat{x}_i^k) - y_0}{e_i^k(\hat{x}_j^k)}, \\ \frac{p^k(x_k) - y_0}{\hat{e}_k} &\leq \frac{p^k(\hat{x}_i^k) - y_0}{-g_L^{F_i^k}(\hat{x}_j^k)}, \\ \frac{p^k(x_k) - f(x^*)}{\hat{e}_k} &\leq \frac{p^k(\hat{x}_i^k) + p^k(\hat{x}_i^k) - 2f(x^*)}{e_i^k(\hat{x}_j^k) - g_L^{F_i^k}(\hat{x}_j^k)}. \end{aligned}$$

According to Assumption 1,  $L_p$  is a Lipchitz constant for  $p^k(x)$ ; noting  $\max\{\|\hat{x}_i^k - x_I^*\|, \|\hat{x}_i^k - x_j^*\|\} \leq L_2^k$  and (50), the above equation thus simplifies to

$$\begin{aligned} \frac{p^k(x_k) - f(x^*)}{\hat{e}_k} &\leq 2B + \frac{2L_2^k L_p}{e_i^k(\hat{x}_j^k) - g_L^{F_i^k}(\hat{x}_j^k)}, \\ f(y_1) - p^k(x_k) &\leq L_p \delta, \\ f(y_1) - f(x^*) &\leq L_p \delta + \hat{e}_k \left\{ 2B + \frac{2L_2^k L_p}{e_i^k(\hat{x}_j^k) - g_L^{F_i^k}(\hat{x}_j^k)} \right\}. \end{aligned}$$

<sup>6</sup> If  $x \leq a/b$ ,  $x \leq c/d$ , and  $b, d > 0$  then  $x \leq (a + c)/(b + d)$ .

Using (52), (51) and  $f(x^*) \leq \min_{z \in S^k} \{f(z)\} \leq f(y_1)$ , it follows:

$$\min_{z \in S^k} \{f(z)\} - f(x^*) \leq L_p \delta + \hat{e}_k \left\{ 2B + \frac{2L_2^k L_p A}{\min_{z \in S^k} \{f(z)\} - f(x^*)} \right\},$$

$$\left[ \min_{z \in S^k} \{f(z)\} - f(x^*) \right]^2 \leq 2L_2^k L_p A \hat{e}_k + \{L_p \delta + 2B e_k\} \left[ \min_{z \in S^k} \{f(z)\} - f(x^*) \right].$$

Perform the quadratic inequality<sup>7</sup> on the above, and the triangular inequality<sup>8</sup> on the square root of (52), it follows that:

$$\min_{z \in S^k} \{f(z)\} - f(x^*) \leq \sqrt{2L_2^k L_p A \hat{e}_k} + [L_p \delta + 2B \hat{e}_k],$$

$$\sqrt{\hat{e}_k} \leq \sqrt{2r_{\max}^k} \sqrt{\delta} + \sqrt{L_c} \sqrt[4]{2r_{\max}^k \delta}.$$

By using above equations and (52), (49) is satisfied.

According to Lemma 5,  $r_{\max}^k$  is bounded above; since  $L_2^k$  is also bounded, since the feasible domain is bounded, it follows that  $C$ ,  $D$  and  $E$  are bounded real numbers. Furthermore, according to Lemma 6,

$$\min_{z \in S^k} \|x'_k - y\| \geq \frac{\delta}{2}. \tag{54}$$

Since  $\|x'_k - y\| = \delta_{\text{des}}$ , we have

$$\min_{z \in S^k} \{f(z)\} - f(x^*) \leq \varepsilon_k,$$

where  $\varepsilon_k = 2C\delta_{\text{des}} + D\sqrt{2\delta_{\text{des}}} + E\sqrt[4]{2\delta_{\text{des}}}$

(55)

Thus, (47) is satisfied for  $A_1 = 2C$ ,  $A_2 = \sqrt{2}D$ , and  $A_3 = \sqrt[4]{2}D$ . □

In the above theorem, it is shown that a point can be obtained whose function value is arbitrarily close to the global minimum, as long as Algorithm 3 is terminated with a sufficiently small value for  $\delta_{\text{des}}$ . This result establishes convergence with respect to the function values for a finite number of steps  $k$ . In the next theorem, we present a property of Algorithm 3 establishing its convergence to a global minimizer in the limit that  $k \rightarrow \infty$ .

**Theorem 2** *If Algorithm 3 is not terminated at step 6, then the sequence  $\{x'_k\}$  has an  $\omega$ -limit point<sup>9</sup> which is a global minimizer of  $L$ .*

*Proof* Define  $z_k$  as a point in  $S^k$  that has the minimum objective value. By construction,  $f(z_r) \leq f(z_l)$ , if  $r > l$ ; thus,  $f(z_k)$  is monotonically non-increasing. Moreover, according to Theorem 1,

$$f(z_k) - f(x^*) \leq A_1 \delta_k + A_2 \sqrt{\delta_k} + A_3 \sqrt[4]{\delta_k},$$

where  $\delta_k = \min_{x \in S^k} \{\|x - x_k\|\}$ ,

(56)

<sup>7</sup> If  $A, B, C > 0$ , and  $A^2 \leq C + BA$  then  $A \leq \sqrt{C} + B$ .

<sup>8</sup> If  $x, y > 0$ , then  $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$ .

<sup>9</sup> The point  $\hat{x}$  is an  $\omega$ -limit point (see [45]) of the sequence  $x_k$ , if a subsequence of  $x_k$  converges to  $\hat{x}$ .

where  $f(x^*)$  is the global minimum. According to Remark 3, for any arbitrarily small  $\delta > 0$ , there is a  $k$  such that  $\min_{x \in S^k} \{\|x - x_k\|\} \leq \delta$ ; thus,

$$f(z_k) - f(x^*) \leq A_1\delta + A_2\sqrt{\delta} + A_3\sqrt[4]{\delta}. \tag{57}$$

Thus, for any  $\varepsilon > 0$ ; there is a  $k$  such that  $0 \leq f(z_k) - f(x^*) \leq \varepsilon$ . Now consider  $x_1$  as an *omega*-limit point of the sequence  $\{z_k\}$ ; thus, there is a subsequence  $\{a_i\}$  of  $\{z_k\}$  that converges to  $x_1$ . Since  $f(x)$  is continuous,

$$f(x_1) = \lim_{i \rightarrow \infty} f(a_i) = f(x^*), \tag{58}$$

which establishes that the  $x_1$  is a global minimizer of  $f(x)$ . □

*Remark 6* Theorems 1 and 2 ensure that Algorithm 3 converges to the global minimum if  $y_0 \leq f(x^*)$ ; however, if  $y_0 > f(x^*)$ , in a manner analogous to Theorem 6 in Part I, it can be shown that Algorithm 3 converges to a point where the function value is less than or equal to  $y_0$ .

### 5 Results

In this section, we apply Algorithm 3 to some representative examples in two and three dimensions with different types of convex constraints to analyze its performance. Three different test functions for  $f(x)$ , where  $x = [x_1, x_2, \dots, x_n]^T$ , are considered:

*Parabolic function:*

$$f(x) = \sum_{i=1}^n x_i^2. \tag{59}$$

*Rastrigin function:* Defining  $A = 2$ ,

$$f(x) = An + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)]. \tag{60}$$

*Shifted Rosenbrock function:*<sup>10</sup> Defining  $p = 10$ ,

$$f(x) = \sum_{i=1}^{n-1} [(x_i)^2 + p(x_{i+1} - x_i^2 - 2x_i)^2]. \tag{61}$$

In the unconstrained setting, the global minimizer of all three test functions considered is the origin.

Another essential part of the class of problems considered in (1) is the constraints that are imposed; this is, in fact, a central concern of the present paper.

Note that all simulations in this section are stopped when  $\min_{y \in S^k} \{\|y - x_k\|\} \leq 0.01$ .

#### 5.1 2D with circular constraints

We first consider 2D optimization problems in which the feasible domain is a circle. In problems such as this, in which there is only one constraint ( $m = 1$ ),  $g_F^L(x) = 0$  for all faces  $F$  at all steps. Thus, no searches are performed on the faces; rather, all searches

<sup>10</sup> This is simply the classical Rosenbrock function with its unconstrained global minimizer shifted to the origin.

are performed in the interior simplices, with some points  $x_k$  (that is, any point  $x_k$  that lands within the circumsphere of a boundary simplex) projected out to the feasible domain boundary.

In this subsection, two distinct circular constraints are considered:

$$(x_1 + 1.5)^2 + (x_2 + 1.5)^2 \leq 5.5, \quad (62a)$$

$$(x_1 + 1.5)^2 + (x_2 + 1.5)^2 \leq 3.8. \quad (62b)$$

The constraint (62a) includes the origin (and, thus, the global minimizer of the test functions considered), whereas the constraint (62b) does not. For the problems considered in this subsection, we take  $\kappa = 1.4$  for all simulations performed, which satisfies (10).

It is observed that, for test functions (59) and (60), the global minimizer in the domain that satisfies (62b) is  $x^* = [-0.1216; -0.1216]$ , and the global minima are  $f(x^*) = 0.0301$  and  $f(x^*) = 0.8634$ , respectively. For the Rosenbrock function (61), the global minimizer is at  $x^* = [-0.0870; -0.1570]$ , and the global minimum is  $f(x^*) = 0.0085$  [it is easy to check that this point is KKT (see [15]), and that the problem is convex].

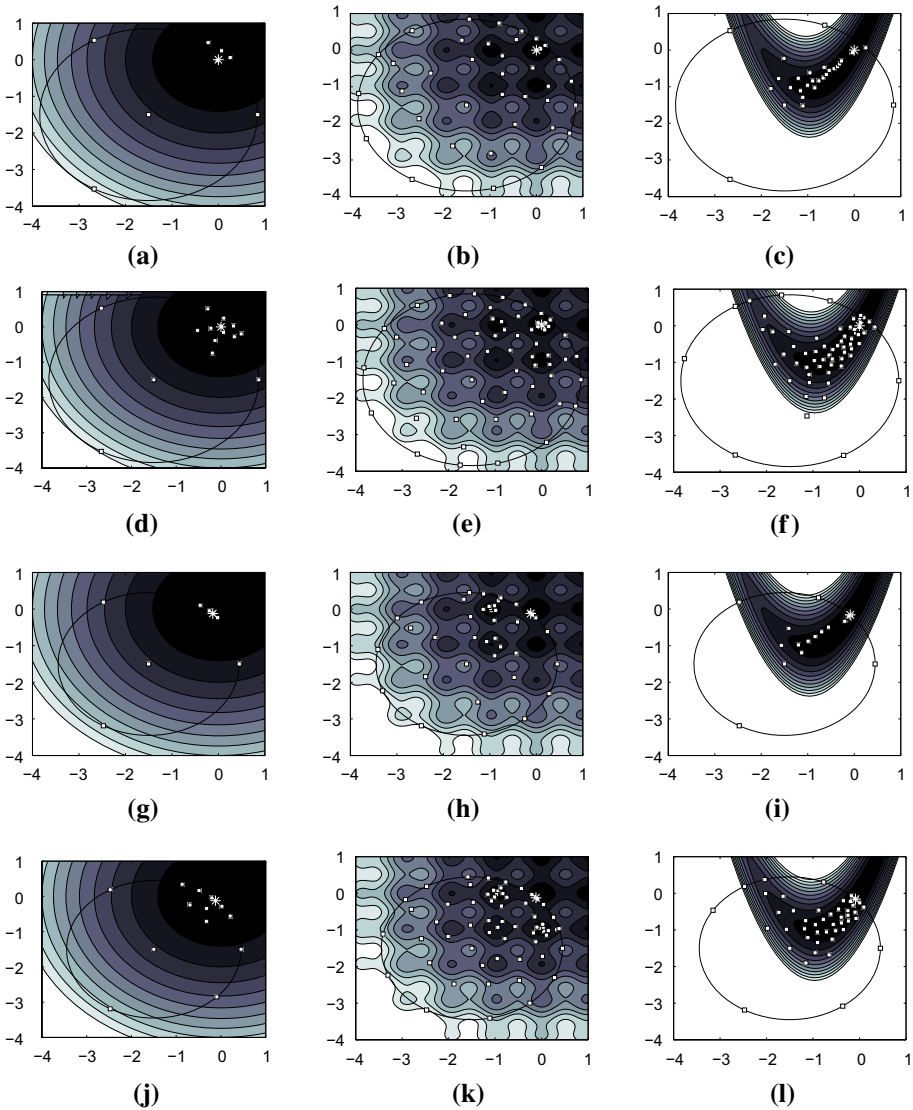
Algorithm 3 is implemented for six optimization problems constructed with the above test functions and constraints. In each problem, the parameter  $y_0$  in Algorithm 3 is chosen to be a lower bound for  $f(x^*)$ . Two different cases are considered: one with  $y_0 = f(x^*)$ , and one with  $y_0 < f(x^*)$ .

The first two problems consider the parabolic test function (59), first with the constraint (62a), then with the constraint (62b). For these problems, in the case with  $y_0 = f(x^*)$  (that is,  $f(x^*) = 0$  for constraint (62a), and  $f(x^*) = 0.0301$  for constraint (62b)), a total of 8 and 7 function evaluations are performed by the optimization algorithm for the constraints given by (62a) and (62b), respectively (see Fig. 5a, g). This remarkable convergence rate is, of course, due to the facts that the global minimum is known and both the function and the curvature of the boundary are smooth.

For the two problems related to the parabolic test function in the case with  $y_0 < f(x^*)$  (we take  $y_0 = -0.1$  in both problems), slightly more exploration is performed before termination; a total of 15 and 13 function evaluations are performed by the optimization algorithm for the constraints given by (62a) and (62b), respectively (see Fig. 5d, j). Another observation is that, for the constraint given by (62b), more function evaluations on the boundary are performed; this is related to the fact that the function value on the boundary is reduced along a portion of the boundary.

The next two problems consider the Rastrigin test function (60), first with the constraint (62a), then with the constraint (62b). With the constraint (62a), this problem has 16 local minima in the interior of the feasible domain, and 10 constrained local minima on the boundary of  $L$ ; with the constraint (62b), it has 12 local minima in the interior of the feasible domain, and 4 constrained local minima on the boundary of  $L$ . For these problems, in the case with  $y_0 = f(x^*)$  (that is,  $f(x^*) = 0$  for constraint (62a), and  $f(x^*) = 0.8634$  for constraint (62b)), a total of 41 and 38 function evaluations are performed by the optimization algorithm for the constraints given by (62a) and (62b), respectively (see Fig. 5b, h). As expected, more exploration is needed for this test function than for a parabola. Another observation for the problem constrained by (62b) is that two different regions are characterized by rather dense function evaluations, one in the vicinity of the global minimizer, and the other in a neighborhood of a local minima (at  $x_1 = [-1; 0]$ ) where the cost function value ( $f(x_1) = 1$ ) is relatively close to the global minimum  $f(x^*) = 0.8634$ .

For the two problems related to the Rastrigin test function in the case with  $y_0 < f(x^*)$  (we take  $y_0 = -0.2$  for the problem constrained by (62a), and  $y_0 = 0.5$  for the problem



**Fig. 5** Location of function evaluations when applying Algorithm 3 to the 2D parabola (59), Rastrigin function (60), and Rosenbrock function (61), when constrained by a circle (62a) containing the unconstrained global minimum, and when constrained by a circle (62a) not containing the unconstrained global minimum. The constrained global minimizer is marked by *star* in all figures. **a** Parabola, in (62a),  $y_0 = f(x^*)$ . **b** Rastrigin, in (62a),  $y_0 = f(x^*)$ . **c** Rosenbrock, in (62a),  $y_0 = f(x^*)$ . **d** Parabola, in (62a),  $y_0 < f(x^*)$ . **e** Rastrigin, in (62a),  $y_0 < f(x^*)$ . **f** Rosenbrock, in (62a),  $y_0 < f(x^*)$ . **g** Parabola, in (62b),  $y_0 = f(x^*)$ . **h** Rastrigin, in (62b),  $y_0 = f(x^*)$ . **i** Rosenbrock, in (62b),  $y_0 = f(x^*)$ . **j** Parabola, in (62b),  $y_0 < f(x^*)$ . **k** Rastrigin, in (62b),  $y_0 < f(x^*)$ . **l** Rosenbrock, in (62b),  $y_0 < f(x^*)$

constrained by (62b)), more exploration is, again, performed before termination; a total of 70 and 64 function evaluations are performed by the optimization algorithm for the constraints given by (62a) and (62b), respectively (see Fig. 5e, k). For the problem constrained by (62b), three different regions are characterized by rather dense function evaluations: in the

vicinity of  $[-1; 0]$ , in the vicinity of  $[0; -1]$ , and in the vicinity of the global minimizer at  $[-0.1216; -0.1216]$ .

The last two problems of this subsection consider the Rosenbrock test function (61), first with the constraint (62a), then with the constraint (62b). This challenging problem is characterized by a narrow valley, with a relatively flat floor, in the vicinity of the curve  $x_2 = x_1^2$ . For these problems, in the case with  $y_0 = f(x^*)$  (that is,  $f(x^*) = 0$  for constraint (62a), and  $f(x^*) = 0.0085$  for constraint (62b)), a total of 29 and 31 function evaluations are performed by the optimization algorithm for the constraints given by (62a) and (62b), respectively (see Fig. 5c, i).

For the two problems related to the Rosenbrock test function in the case with  $y_0 < f(x^*)$  (we take  $y_0 = -0.5$  in both problems), more exploration is performed before termination; a total of 47 and 54 function evaluations are performed by the optimization algorithm for the constraints given by (62a) and (62b), respectively (see Fig. 5f, l). As with the previous problems considered, exact knowledge of  $f(x^*)$  significantly improves the convergence rate. Additionally, such knowledge confines the search to a smaller region around the curve  $y = x^2$ , and fewer boundary points are considered for function evaluations.

## 5.2 2D with elliptical constraints

The next two constraints considered are diagonally-oriented ellipses with aspect ratios 4 and 10:

$$(x + y + 3)^2 + 16(x - y)^2 \leq 10, \quad (63a)$$

$$(x + y + 3)^2 + 100(x - y)^2 \leq 10. \quad (63b)$$

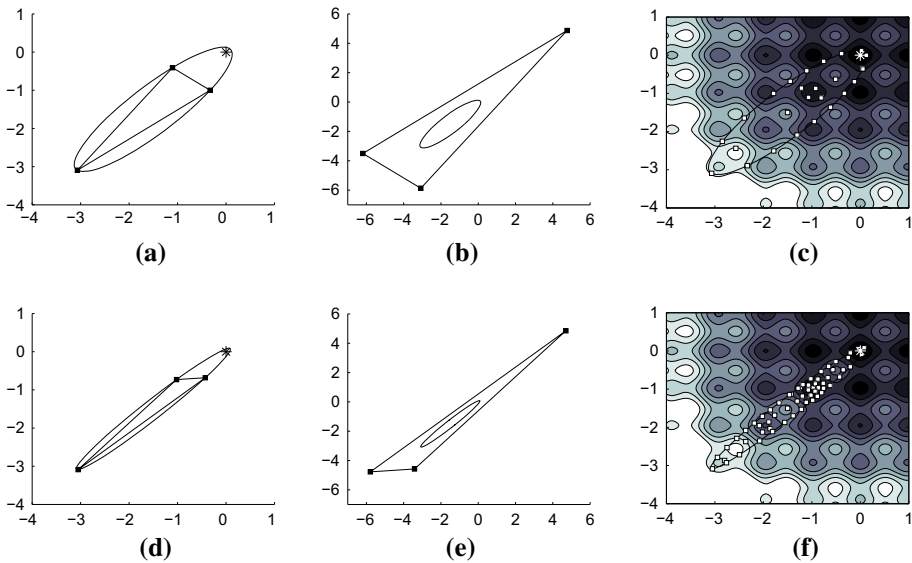
These constraints are somewhat more challenging to deal with than those considered in Sect. 5.1. As in Sect. 5.1,  $g_F^L(x) = 0$  for all faces  $F$  at all steps; thus, no searches are performed on the faces.

The Rastrigin function has 4 unconstrained local minimizers inside each of these ellipses. Additionally, there are 6 constrained local minimizers on the boundary (63a); note that there are no constrained local minimizers on the boundary (63b).

In this subsection, Algorithm 3 is applied to the 2D Rastrigin function (60) for the two different elliptical constraints considered. The unconstrained global minimizer of this function is at  $x^* = [0, 0]$ , where the global minimum is  $f(x^*) = 0$ ; this unconstrained global minimizer is contained within the feasible domain for both (63a) and (63b). We take  $y_0 = f(x^*) = 0$  for all simulations reported in this subsection.

For the problems considered in this subsection, we take  $\kappa = 2$  for all simulations performed, which satisfies (10). Note that, though the aspect ratio of the feasible domain is relatively large (especially in (63b)), the minimum distance of the body center of the initial simplex from its vertices is relatively large; thus, a relatively small value for extending parameter  $\kappa$  may be used. As seen in Fig. 6, for the cases constrained by (63a) and (63b), the simulations are terminated with 24 and 60 function evaluations, respectively. This indicates that the performance of the algorithm depends strongly on the aspect ratio of the feasible domain, as predicted by the analysis presented in Sect. 4, as the parameter  $K_g$  in Assumption 6 increases with the square of the aspect ratio of the elliptical feasible domain [see the explicit dependence on  $K_g$  in (48) and (49)]. In the case constrained by (63b), Fig. 6d shows that the global minimizer lies outside of the smallest face of the initial simplex, and is situated relatively far from this face. In this case, the reduced uncertainty  $e^k(x)$  across this small face (a result of the fact that its vertices are close together) belies the substantially larger uncertainty





**Fig. 6** Implementation of Algorithm 3 with  $y_0 = 0$  on the 2D Rastrigin problem (60) within the ellipse (63). **a** Initial simplex for (63a). **b** Exterior simplex for (63a). **c** Evaluation points for constraint (63a). **d** Initial simplex for (63b). **e** Exterior simplex for (63b). **f** Evaluation points for constraint (63b)

present in the feasible domain far outside the initial simplex, in the vicinity of the global minimizer; this, in effect, slows convergence. Note that this situation only occurs when the curvature of the boundary is large. An alternative approach to address problems with large aspect ratio is to eliminate the highly-constrained coordinate direction(s) altogether, solve a lower-dimensional optimization problem, then locally optimize the original problem in the vicinity of the lower-dimensional optimal point. Note that, through the construction of the enclosing simplex (see Definition 3), the highly-constrained coordinate directions can easily be identified and eliminated.

### 5.3 2D with multiple constraints

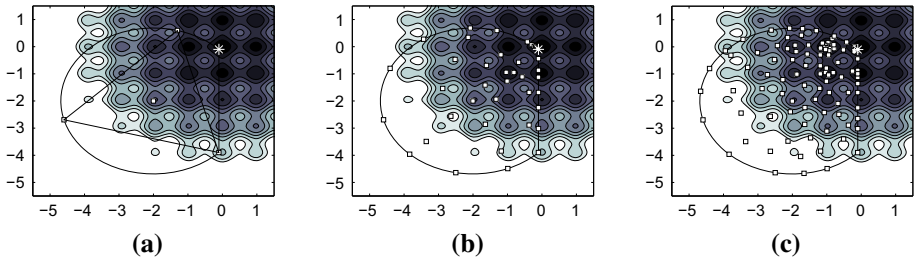
In this subsection, Algorithm 3 is implemented for problems characterized by the union of multiple linear or nonlinear constraints on the feasible domain, which specifically causes corners (that is, points at which two or more constraints are active) in the feasible domain. In such problems, the value of  $g_F^L(x)$  is nonzero at some faces of the Delaunay triangulations; Algorithm 3 thus requires searches over the faces of the triangulations. The example considered in this subsection quantifies the importance of this process of searching over the faces of the triangulations in such problems.

The test function considered in this subsection is the 2D Rastrigin function (60), and the feasible domain considered is as follows:

$$c_1(x) = (x_1 + 2)^2 + (x_2 + 2)^2 \leq 2 \times 1.9^2, \tag{64a}$$

$$c_2(x) = x_1 \leq -0.1. \tag{64b}$$

The Rastrigin function in the feasible domain characterized by (64) has 18 local minima inside the feasible domain, and 7 constrained local minima on the boundary of the feasible domain.



**Fig. 7** Implementation of Algorithm 3 with  $y_0 = f(x^*)$  on the 2D Rastrigin function (60) with constraints (64), with and without searching over the faces. **a** Location of the initial simplex. **b** Implementation of Algorithm 3. **c** Skipping the searches over the faces

Note that the unconstrained global minimizer of (60) is not in the domain defined by the constraints (64).

The global minimizer of (60) within the feasible domain defined by (64) is  $x^* = [-0.1; -0.1]$ , and the global minimum is  $f(x^*) = 0.7839$ .

In order to benchmark the importance of the search over the faces of the triangulations in Algorithm 3, two different optimizations on the problem described above (both taking  $y_0 = f(x^*)$  and  $\kappa = 1.4$ ) are performed. In the first, we apply Algorithm 3. In the second, the same procedure is applied; however, the search over the faces of the triangulation is skipped.

As shown in Fig. 7b, Algorithm 3 requires only 35 function evaluations before the algorithm terminates, and a point in the vicinity of the global minimizer is found. In contrast, as shown in Fig. 7c, when the search over the faces of the triangulation is skipped in Algorithm 3, 104 function evaluations are required before the algorithm terminates. In problems of this sort, in which the constrained global minimizer is at a corner, Algorithm 3 requires searches over the faces in order to move effectively along the faces into the corner.

Note also in Fig. 7b that, when a point in the corner is the global minimizer, Algorithm 3 converges slowly along the constraint boundary towards the corner. A potential work-around to this problem might be to switch to a derivative-free local optimization method (e.g. [2, 9, 35]) when a point in the vicinity of the solution near a corner is identified.

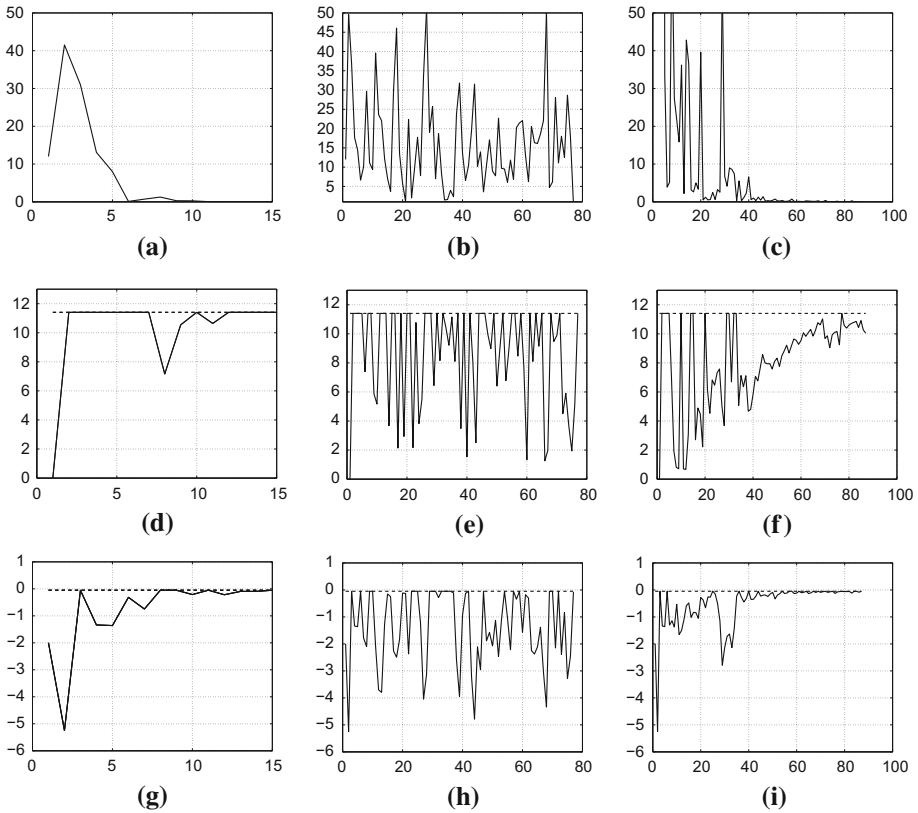
### 5.4 3D problems

To benchmark the performance of Algorithm 3 on higher-dimensional problems, it was applied to the 3D parabolic (59), 3D Rastrigin (60) and 3D Rosenbrock (61) test functions with a feasible domain that looks approximately like the  $D$ -shaped feasible domain illustrated in Fig. 7 rotated about the horizontal axis:

$$c_1(x) = \sum_{i=1}^3 (x_i + 2)^2 \leq 3 \times 1.95^2, \tag{65a}$$

$$c_2(x) = x_1 \leq -0.05. \tag{65b}$$

The constrained global minimizer  $x^*$  for the 3D parabolic and 3D Rastrigin test functions in this case is at the corner  $[-0.05; -0.05; -0.05]$ , where both constraints are active. The constrained global minimizer  $x^*$  for the 3D Rosenbrock test functions is at  $[-0.05; -0.0975; -0.1875]$ , where one of the constraints  $[c_2(x)]$  is active.



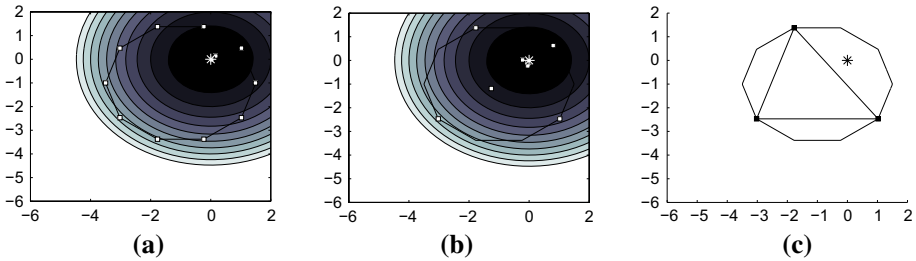
**Fig. 8** Algorithm 3 applied to the 3D parabolic (59), Rastrigin (60), and Rosenbrock (61) test functions, with a feasible domain defined by (65). **a** Function values for Parabola (59). **b** Function values for Rastrigin (60). **c** Function values for Rosenbrock (61). **d**  $c_1(x)$  at evaluated points for parabola. **e**  $c_1(x)$  for Rastrigin. **f**  $c_1(x)$  for Rosenbrock. **g**  $c_2(x)$  at evaluated points for parabola. **h**  $c_2(x)$  for Rastrigin. **i**  $c_2(x)$  for Rosenbrock

Algorithm 3 with  $y_0 = 0$  and  $\kappa = 1.4$  was applied to the problems described above. As shown in Fig. 8, the parabolic, Rastrigin, and Rosenbrock test cases terminated with 15, 76 and 87 function evaluations, respectively. As mentioned in the last paragraph of Sect. 5.3, convergence in the first two of these test cases is slowed somewhat from what would be achieved otherwise due to the fact that the constrained global minimum lies in a corner of the feasible domain. Note that the Rastrigin case is fairly difficult, as this function is characterized by approximately 140 local minima inside the feasible domain, and 50 local minima on the boundary of the feasible domain; nonetheless, Algorithm 3 identifies the nonlinearly constrained global minimum in only 76 function evaluations.

### 5.5 Linearly constrained problems

In this section, Algorithm 3 is compared with Algorithm 4 in Part I of this work (the so-called “Adaptive  $K$ ” algorithm) for a linearly constrained problem with a smooth, simple objective function, but a large number of corners of the feasible domain.

The test function considered is the 2D parabola (59), and the set of constraints applied is



**Fig. 9** Implementation of Algorithm 3 of the present paper, and Algorithm 4 of Part I, to a parabolic test function constrained by a decagon. **a** Algorithm 4 of Part I. **b** Algorithm 3 of the present paper. **c** Initial simplex used in **(b)**

$$\begin{aligned}
 a_i &= [\cos(2\pi i/m) \quad \sin(2\pi i/m)]^T, \\
 b_i &= 2.5 \cos(\pi/m) - [1 \quad 1] a_i, \\
 a_i^T x &\leq b_i, \quad 1 \leq i \leq m.
 \end{aligned}
 \tag{66}$$

The feasible domain characterized by (66) is a uniform  $m$ -sided polygon with  $m$  vertices. The feasible domain contains the unconstrained global minimum at  $[0, 0]$ , and  $y_0 = f(x^*) = 0$  is used for both simulations reported.

The initialization of Algorithm 4 in Part I requires  $m$  initial function evaluations, at the vertices of the feasible domain; the case with  $m = 10$  (taking  $r = 5$ , as discussed in §4 of Part I) is illustrated in Fig. 9a, and is seen to stop after 15 function evaluations (that is, 5 iterations after initialization at all corners of the feasible domain).

In contrast, as illustrated Fig. 9c, Algorithm 3 of the present paper (taking  $\kappa = 1.4$ ), applied to the same problem, stops after only 10 function evaluations (that is, 6 iterations after initialization at the vertices of the initial simplex and its body center). Note that one feasible boundary projection is performed in this simulation, as the global minimizer is not inside the initial simplex.

These simulations show that, for linearly-constrained problems with smooth objective functions, it is not necessary to calculate the objective function at all vertices of the feasible domain. As the dimension of the problem and/or the number of constraints increases, the cost of such an initialization might become large. Using the algorithm developed in the present paper, it is enough to initialize with between  $n + 2$  and  $2n + 3$  function evaluations.

### 5.6 Role of the extending parameter $\kappa$

The importance of the extending parameter  $\kappa$  is now examined by applying Algorithm 3 to the Rastrigin function (60) within a linearly-constrained domain:

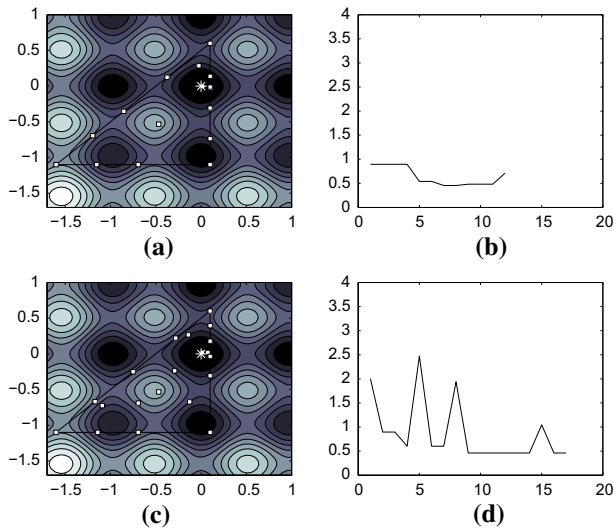
$$x \leq 0.1, \tag{67a}$$

$$-1.1 \leq y, \tag{67b}$$

$$y - x \leq 0.5. \tag{67c}$$

The Rastrigin function (60) has 3 local minimizers inside this triangle, and there are not any constrained local minimizers on the boundary of this triangle.

The feasible domain includes the unconstrained global minimizer in this case; thus, the constrained global minimizer is  $[0, 0]$ . Two different values of the extending parameter are considered,  $\kappa = 1.1$  and  $\kappa = 100$ . It is observed (see Fig. 10) that, for  $\kappa = 1.1$ , the algorithm



**Fig. 10** Implementation of Algorithm 3 with  $y_0 = 0$  on 2D Rastrigin problem (60) in a triangle characterized by (67) with  $\kappa = 1.1$  and  $\kappa = 100$ . **a** Evaluation points for  $\kappa = 1.1$ . **b** The maximum circumradius for  $\kappa = 1.1$ . **c** Evaluation points for  $\kappa = 100$ . **d** The maximum circumradius for  $\kappa = 100$

converges to the vicinity of the solution in only 12 function evaluations; for  $\kappa = 100$ , 16 function evaluations are required, and the feasible domain is explored more.

The main reason for this phenomenon is related to the value of the maximum circumradius (see Fig. 10b, d). In the  $\kappa = 100$  case, Algorithm 3 performs a few unnecessary function evaluations, close to existing evaluation points, at those iterations during which the maximum circumradius of the triangulation is large. The  $\kappa = 1.1$  case better regulates the maximum circumradius of the triangulation, thus inhibiting such redundant function evaluations from occurring.

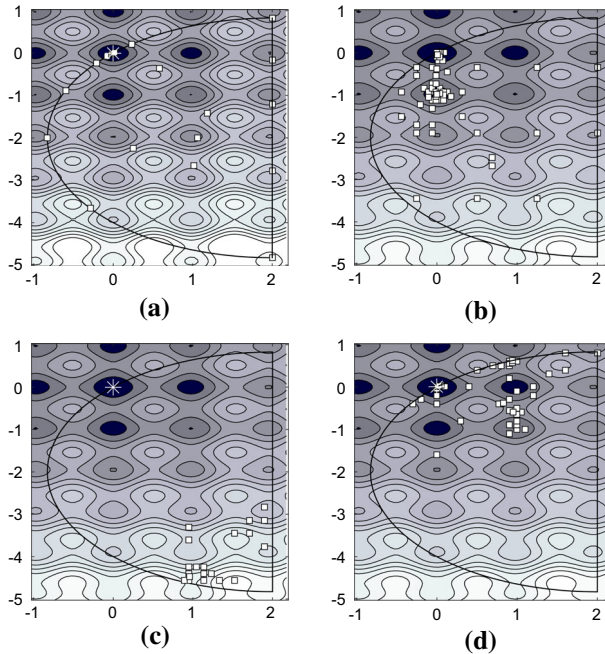
### 5.7 Comparison with other derivative-free methods

In this section, we briefly compare our new algorithm with two modern benchmark methods among derivative free optimization algorithms. The first method considered is MADS (mesh adaptive direction search; see [3]), as implemented in the NOMAD software package [32]. This method (NOMAD with  $2n$  neighbors) is a local derivative-free optimization algorithm which can handle constraints on the feasible domain. The second method considered is SMF (surrogate management framework; see [8]). This algorithm is implemented in [37] for airfoil design.<sup>11</sup> This method is a hybrid method that combines a generalized pattern search with a kriging based optimization algorithm.

The test problem considered here for the purpose of this comparison is the minimization of the 2D Rastrigin function (60), with  $A = 3$ , inside a feasible domain  $L$  characterized by the following constraints:

$$c_1(x) = x \leq 2, \quad c_2(x) = (x - 2)^2 + (y + 2)^2 \leq 8. \tag{68}$$

<sup>11</sup> Though this code is not available online, we have obtained a copy of it by personal communication from Prof. Marsden.



**Fig. 11** Implementation of **a** Algorithm 3, **b** SMF, and **c, d** MADS on minimization of the 2D Rastrigin function (60) inside the feasible domain characterized by (68). The feasible domain is indicated by the *black semicircle*, the function evaluations performed are indicated as *white squares*, and the feasible global minimizer is indicated by an *asterisk*

In this test, for Algorithm 3, the value of  $y_0 = f(x^*) = 0$  was used, and the termination condition, similar to the previous sections, is taken by  $\min_{x \in S^k} \|x - x_k\| \leq 0.01$ . Results are shown in Fig. 11.

Algorithm 3 performed 19 function evaluations (see Fig. 11a), which includes some initial exploration of the feasible domain, and more dense function evaluations in the vicinity the global minimizer. Note that, since Algorithm 3 uses interpolation during its minimization process, once it discovers a point in the neighborhood of the global solution, convergence to the global minimum is achieved rapidly.

The SMF algorithm similarly performs both global exploration and local refinement. However, the number of function evaluations required for the same test problem increases to 66 (see Fig. 11b). The reasons for this appear to be essentially twofold. First, the numerical performance of the polyharmonic spline interpolation together with our synthetic uncertainty function appear to behave favorably as compared with Kriging, as also observed in Part I of this work. Second, the search step of the SMF [37] was not designed for nonlinearly constrained problems. In other words, the polling step is the only step of SMF that deals with the complexity of the constraints.

Unlike Algorithm 3, MADS is not designed for global optimization; thus, there is no guaranty of convergence to the global minimum. In this test, two different initial points were considered to illustrate the behavior of MADS. Starting from  $x_0 = [1.9; -\pi]$  (see Fig. 11c), MADS converged to a local minimizer after 19 function evaluations; however, this local minimum is not the global solution. Starting from  $x_0 = [1; -1]$  (see Fig. 11d), MADS converged to the global minimizer at  $[0, 0]$  after 59 function evaluations.

In the above tests, we have counted those function evaluations that are performed within  $L$ , as the constraint functions  $c_i(x)$  are assumed to be computationally inexpensive. Note that Algorithm 3 does not include any function evaluations outside of  $L$ .

Although this particular test shows a significant advantage for Algorithm 3 compared to the SMF and MADS tests, more study is required to analyze properly the performance of this algorithm. An important fact about this algorithm is that its performance is dependent on the interpolation strategy used; this dependence will be studied in future work. A key advantage of Algorithm 3 is that it lets the user choose the interpolation strategy to be implemented; the best choice might well be problem dependent.

## 6 Conclusions

In this paper, the derivative-free global optimization algorithms developed in Part I of this study were extended to the optimization of nonconvex functions within a feasible domain  $L$  bounded by a set of convex constraints. The paper focused on extending Algorithm 4 of Part I (the so-called Adaptive  $K$  algorithm) to convex domains; we noted in Remark 2 that the extension of Algorithm 2 of Part I (the so-called Constant  $K$  algorithm) may be extended in an analogous fashion.

We developed three algorithms:

- Algorithm 1 showed how to initialize the optimization by identifying  $n + 1$  points in  $L$  which maximize the volume of the initial simplex generated by these points. An enclosing simplex was also identified which includes  $L$ .
- The initial simplex identified by Algorithm 1, together with its body center, were augmented by Algorithm 2 with up to  $n + 1$  additional points to better separate the interior simplices from the boundary simplices in the Delaunay triangulations to be used.
- Algorithm 3 then presented the optimization algorithm itself, which modified the Algorithms of Part I in order to extend the convex hull of the evaluation points to cover the entire feasible domain as the iterations proceed. An important property of this algorithm is that it ensures that the triangulation remains well behaved (that is, the maximum circumradius remains bounded) as new datapoints are added.

In the algorithm developed, there are two adjustable parameters. The first parameter is  $y_0$ ; this is an estimate of the global minimum, and has a similar effect on the optimization algorithm as seen in Algorithm 4 of Part I. The second parameter is  $\kappa > 1$ , which quantifies the size of the exterior simplex. It is shown in our analysis (Sect. 4) that, for  $1 < \kappa < \infty$ , the maximum circumradius is bounded. In practice, it is most efficient to choose this parameter as small as possible while respecting the technical condition (10) required in order to assure convergence.

The performance of the algorithm developed is shown in our results (Sect. 5) to be good if the aspect ratio of  $L$  of the feasible domain is not too large. Specifically, the curvature of each constraint  $c_i(x)$  is seen to be as significant to the overall rate of convergence as the smoothness of the objective function  $f(x)$  itself. It was suggested in Sect. 5 that, in feasible domains with large aspect ratios, those directions in parameter space that are highly constrained can be identified and eliminated from the global optimization problem; after the global optimization is complete, local refinement can be performed to optimize in these less important directions.

Another challenging aspect of the algorithm developed is its reduced rate of convergence when the global minimizer is near a corner of the feasible domain. This issue can be addressed

by switching to a local optimization method when it is detected that the global minimizer is near such a corner (that is, when multiple values of  $c_i(x)$  are near zero).

In future work, we extend the algorithms developed to problems in which function evaluations are inexact, and can be improved with additional computational effort. Moreover, this algorithm will be applied on a wide range of test problems. In particular, the performance of the present algorithm depends on the particular interpolation strategy used (unlike various other response surface methods, this method is not limited to a specific type of interpolation strategy).

**Acknowledgments** The authors gratefully acknowledge Prof. Phillip Gill and Prof. Alison Marsden for their collaborations and funding from AFOSR FA 9550-12-1-0046, Cymer Center for Control Systems & Dynamics, and Leidos corporation in support of this work.

## References

1. Abramson, M.A., Audet, C., Dennis Jr., J.E., Le Digabel, S.: OrthoMADS: a deterministic MADS instance with orthogonal directions. *SIAM J. Optim.* **20**(2), 948–966 (2009)
2. Audet, C., Dennis Jr., J.E.: A pattern search filter method for nonlinear programming without derivatives. *SIAM J. Optim.* **14**(4), 980–1010 (2004)
3. Audet, C., Dennis Jr., J.E.: Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.* **17**(1), 188–217 (2006)
4. Audet, C., Dennis Jr., J.E.: A progressive barrier for derivative-free nonlinear programming. *SIAM J. Optim.* **20**(1), 445–472 (2009)
5. Barvinok, A.: *A Course in Convexity*, vol. 50. American Mathematical Society, Providence (2002)
6. Bauschke, H.H., Borwein, J.M.: On projection algorithms for solving convex feasibility problems. *SIAM Rev.* **38**(3), 367–426 (1996)
7. Beyhaghi, P., Cavaglieri, D., Bewley, T.: Delaunay-based derivative-free optimization via global surrogates, part I: linear constraints. *J. Glob. Optim.* **63**(3), 1–52 (2015)
8. Booker, A.J., Dennis Jr., J.E., Frank, P.D., Serafini, D.B., Torczon, V., Trosset, M.W.: A rigorous framework for optimization of expensive functions by surrogates. *Struct. Optim.* **17**(1), 1–13 (1999)
9. Belitz, P., Bewley, T.: New horizons in sphere-packing theory, part II: lattice-based derivative-free optimization via global surrogates. *J. Glob. Optim.* **56**(1), 61–91 (2013)
10. Bertsekas, D.P.: On penalty and multiplier methods for constrained minimization. *SIAM J. Control Optim.* **14**(2), 216–235 (1976)
11. Bertsimas, D., Tsitsiklis, J.N.: *Introduction to Linear Optimization*, vol. 6. Athena Scientific, Belmont (1997)
12. Boissonnat, J.D., Devillers, O., Hornus, S.: Incremental construction of the Delaunay triangulation and the Delaunay graph in medium dimension. In: *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, pp. 208–216. ACM (2009)
13. Coxeter, H.S.M.: *Regular Polytopes*. Courier Corporation, Chelmsford (1973)
14. Combettes, P.L.: Hilbertian convex feasibility problem: convergence of projection methods. *Appl. Math. Optim.* **35**(3), 311–330 (1997)
15. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
16. Dwyer, R.A.: A faster divide-and-conquer algorithm for constructing Delaunay triangulations. *Algorithmica* **2**(1–4), 137–151 (1987)
17. Dwyer, R.A.: Higher-dimensional Voronoi diagrams in linear expected time. *Discrete Comput. Geom.* **6**(3), 343–367 (1991)
18. Dwyer, R.A.: The expected number of k-faces of a Voronoi diagram. *Comput. Math. Appl.* **26**(5), 13–19 (1993)
19. Facchinei, F., Pang, J.S.: *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer, Berlin (2007)
20. Floudas, C.A., Pardalos, P.M.: *A Collection of Test Problems for Constrained Global Optimization Algorithms*, vol. 455. Springer, Berlin (1990)
21. Forsgren, A., Gill, P.E., Wright, M.H.: Interior methods for nonlinear optimization. *SIAM Rev.* **44**(4), 525–597 (2002)



22. Forsgren, A., Gill, P.E.: Primal-dual interior methods for nonconvex nonlinear programming. *SIAM J. Optim.* **8**(4), 1132–1152 (1998)
23. Gill, P.E., Murray, W.: Newton-type methods for unconstrained and linearly constrained optimization. *Math. Program.* **7**(1), 311–350 (1974)
24. Gill, P.E., Murray, W., Saunders, M.A., Wright, M.H.: A Schur-Complement Method for Sparse Quadratic Programming (No. SOL-87-12). Stanford University CA Systems Optimization Lab (1987)
25. Gill, P.E., Murray, W., Saunders, M.A., Tomlin, J.A., Wright, M.H.: On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Math. Program.* **36**(2), 183–209 (1986)
26. Heinonen, J.: Lectures on Lipschitz analysis. University Report, University of Jyväskylä Department of Mathematics and Statistics, 100, University of Jyväskylä, Jyväskylä, pp. 1–77 (2005)
27. Horst, R., Pardalos, P.M. (eds.): Handbook of Global Optimization, vol. 2. Springer, Berlin (2013)
28. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theory Appl.* **79**(1), 157–181 (1993)
29. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. *J. Glob. Optim.* **21**(4), 345–383 (2001)
30. Kort, B.W., Bertsekas, D.P.: A new penalty function method for constrained minimization. In: Proceedings of the 1972 IEEE Conference on Decision and Control, 1972 and 11th Symposium on Adaptive Processes, pp. 162–166. IEEE (1972)
31. Kort, B.W., Bertsekas, D.P.: Combined primal-dual and penalty methods for convex programming. *SIAM J. Control Optim.* **14**(2), 268–294 (1976)
32. Le Digabel, S.: Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm. *ACM Trans. Math. Softw.* **37**(4), 44 (2011)
33. Lewis, R.M., Torczon, V.: Pattern search algorithms for bound constrained minimization. *SIAM J. Optim.* **9**(4), 1082–1099 (1999)
34. Lewis, R.M., Torczon, V.: Pattern search methods for linearly constrained minimization. *SIAM J. Optim.* **10**(3), 917–941 (2000)
35. Lewis, R.M., Torczon, V.: A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM J. Optim.* **12**(4), 1075–1089 (2002)
36. Mangasarian, O.L., Fromovitz, S.: The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. *J. Math. Anal. Appl.* **17**(1), 37–47 (1967)
37. Marsden, A.L., Wang, M., Dennis Jr., J.E., Moin, P.: Optimal aeroacoustic shape design using the surrogate management framework. *Optim. Eng.* **5**(2), 235–262 (2004)
38. McMullen, P.: The maximum numbers of faces of a convex polytope. *Mathematika* **17**(02), 179–184 (1970)
39. Nocedal, J., Wright, S.: Numerical Optimization. Springer, New York (2006)
40. Paulavicius, R., Zilinskas, J.: Simplicial Global Optimization. Springer, New York (2014)
41. Preparata, F.P., Shamos, M.: Computational Geometry: An Introduction. Springer, Berlin (2012)
42. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
43. Schonlau, M., Welch, W.J., Jones, D.R.: A data-analytic approach to bayesian global optimization. In: Department of Statistics and Actuarial Science and The Institute for Improvement in Quality and Productivity, ASA Conference (1997)
44. Shubert, B.O.: A sequential method seeking the global maximum of a function. *SIAM J. Numer. Anal.* **9**(3), 379–388 (1972)
45. Smith, H.L., Waltman, P.: The Theory of the Chemostat: Dynamics of Microbial Competition, vol. 13. Cambridge University Press, Cambridge (1995)
46. Susca, S., Bullo, F., Martinez, S.: Gradient algorithms for polygonal approximation of convex contours. *Automatica* **45**(2), 510–516 (2009)
47. Torczon, V.: On the convergence of pattern search algorithms. *SIAM J. Optim.* **7**(1), 1–25 (1997)
48. Zhigljavsky, A., Zilinskas, A.: Stochastic Global Optimization, vol. 9. Springer, Berlin (2007)
49. Zwart, P.B.: Global maximization of a convex function with linear inequality constraints. *Oper. Res.* **22**(3), 602–609 (1974)