Geodesic and contour optimization using conformal mapping

Ricky Fok1 · Aijun An1 · Xiaogong Wang2

Received: 15 May 2015 / Accepted: 14 September 2016 / Published online: 19 September 2016
© Springer Science+Business Media New York 2016

Abstract  We propose a novel optimization algorithm for differentiable functions utilizing geodesics and contours under conformal mapping. The algorithm can locate multiple optima by first following a geodesic curve to a local optimum then traveling to the next search area by following a contour curve. Alongside we implement a jumping mechanism which we call shadow casting to help geodesics jump to locations closer to the global optimum. To improve the efficiency, local search methods such as the Newton–Raphson algorithm are also employed. For functions with many optima or when the global optimum is very close to a local one, numerical analyses have shown that the resulting algorithm, SGEO-QN, can outperform recent derivative-free DIRECT variants in number of function/gradient evaluations. The results also indicate that under certain conditions, number of function/gradient evaluations for SGEO-QN scales nearly linearly with increasing dimensionality. Lastly, SGEO-QN appears to be less affected by rotational transforms of the objective functions than the variants of DIRECT compared.

Keywords  Large scale · Black box global optimization · Box constrained · Multistart · Trajectory method

Ricky Fok
ricky.fok3@gmail.com
Aijun An
ann@cse.yorku.ca
Xiaogong Wang
stevenw@mathstat.yorku.ca

1 Department of Computer Science, York University, 4700 Keele Street, Toronto M3J 1P3, Canada
2 Department of Mathematics and Statistics, York University, 4700 Keele Street, Toronto M3J 1P3, Canada
1 Introduction

We propose an algorithm with a jumping mechanism to solve black box optimization problems of differentiable functions with box constraints. The goal is to find $x^*$ such that

$$x^* = \arg \max_x f(x), \quad x \in [u, v]^D, \quad u, v \in \mathbb{R},$$

and $f : \mathbb{R}^D \to \mathbb{R}$ is a differentiable function.

The volume of literature on optimization methods is massive. A review of trajectory methods is given in [1], where the paths of optimization are traced by solving differential equations, for instance, Newton’s equation of motion. In homotopy and continuation methods [2–4], one attempts to trace a parametrized “zero-curve” that connects any point to a local optimum of $f$. An effort to employ homotopy methods for global optimization has been made [5]. Research related to optimization techniques on Riemannian manifolds includes [6–8], where the constraint surfaces are viewed as Riemannian manifolds and local search techniques such as conjugate gradient descent and Newton’s method are generalized from Euclidean to Riemannian manifolds in order to solve constrained optimization problems. Rapcsák [9–12] has conducted numerous investigations on optimization and convexity on Riemannian manifolds. Further, derivative-free methods exist to solve optimization problems with non-differentiable objective functions, as long as they satisfy certain criteria such as Lipschitz continuity. One such example is the Dividing Rectangle (DIRECT) algorithms [13–15]. Variants for differentiable functions also exist, for instance, [16,17]. Space-filling curve methods also exist as an alternative to DIRECT algorithms [18–20].

In this paper we propose a new trajectory method utilizing geodesics on a Riemannian manifold conformally related to $\mathbb{R}^D$ that follow either the gradient or level curves asymptotically. The step size of the geodesic is inversely proportional to the norm of the gradient. Hence it is able to escape from one basin and explore another part of search space efficiently. The role of the geodesic is two-folds. First, it can locate multiple optima and second, dependent upon the function values along the geodesic, it stochastically defines the direction of line search for the jumping mechanism described in the next paragraph, whether it is towards the “spotlight”, which is a specified point in the search space, or in some other direction.

Many of the aforementioned methods focus on locating the global optimum or its basin of attraction. As the dimension increases the number of function evaluations required to locate the optimum may increase exponentially. To such a point that the cost of optimization becomes impractical. In addition to the geodesic search method, we propose a method to alleviate the crippling dependence on dimensionality. The idea is to transform the basin of attraction of the undiscovered global optimum, say, to a much larger region. Then the optimization problem reduces to finding the enlarged region and map the solution to the global optimum. As an analogy, suppose the task at hand is to find a small pebble hung by an invisible string in a big warehouse lit by a spotlight. The procedure is analogous to finding the shadow cast by the pebble instead of the pebble itself, then tracing back (e.g. via a line search) in the direction towards the spotlight in order to locate the pebble. Hence we name the procedure *shadow casting*. Continuing the analogy, the size of the shadow is proportional to the solid angle subtended by the cross-section of the pebble from the spotlight, which increases with dimension. The key point is that no matter how badly the spotlight is placed (e.g. when the basin of the optimum and the spotlight are at the opposite corners of the search space). The size of the enlarged region grows with dimension and it should be much easier to find rather than the basin of attraction itself.

In combining shadow casting and the geodesic we show that the search space can be effectively explored - the geodesic propagates towards optima whereas line searches along
the direction of the spotlight is independent of the basins of attraction. The algorithm performs line searches along directions determined by the trajectories of the geodesics. Local searches can also be implemented along the geodesics or the line searches. In this way, our method is similar to the two-phase search strategy [21,22].

Numerical results show that the method performs well on test problems in high dimensions, problems with many optima and problems where the global optimum is very close to a local optimum. In particular, our method requires fewer function/gradient evaluations on the majority of “hard” problems defined in [15], in some cases up to two to three orders of magnitude.

The remaining parts of the paper is organized as follows. In Sect. 2 we give an introductory review on differential geometry and geodesics. Theoretical properties of the geodesics are established in Sect. 3. In Sect. 4, we describe the shadow casting mechanism and the algorithmic scheme. Results of the numerical analysis is given in Sect. 5. Finally, the conclusion is given in Sect. 6.

2 Review of differential geometry

We review the basics of differential geometry required for constructing the geodesics in order to discover multiple maxima on a manifold conformally related to $\mathbb{R}^D$.

2.1 Geodesics and geodesic equations

We consider a topological manifold that is a second countable and locally compact Hausdorff space. It is also connected and completely regular. Detailed discussions can be found in [23] and [24]. A Riemannian metric on a smooth and differentiable manifold $M$ is a 2-tensor field $T^2(M)$ that is symmetric and positive definite. A Riemannian metric thus determines an inner product on each tangent space $T_p(M)$, which is typically written as $g(U, V)$ for $U, V \in T_p(M)$. For an Euclidean space, the metric matrix (or just metric henceforth) in component form is the Kronecker delta, i.e $g_{ij} = \delta_{ij}$. The inner product $g(U, V)$ in Euclidean reduces to the dot product, $\sum_{ij} \delta_{ij} U^i V^j$, where the sum is over all dimensions. In the Einstein summation convention, it is understood that repeated indices are summed over and the inner product is expressed as $g_{ij} U^i V^j$.

A geodesic is defined to be the path of minimum length for two given distinct points in a connected manifold. It is simply a straight line in Euclidean space. On a curved manifold, however, it is no longer a straight line when projected onto $\mathbb{R}^D$.

Let $X^i(t)$ denote the local coordinate for the $i$-th dimension for a parameter $t$ which is a time step in our case. The geodesic is then characterized by a set of partial differential equations, using the Einstein summation convention:

$$\frac{d^2 X^i(t)}{dt^2} + \Gamma^i_{jk} \frac{dX^j(t)}{dt} \frac{dX^k(t)}{dt} = 0,$$

(1)

where $\Gamma^i_{jk}$ are Christoffel symbols defined to be

$$\Gamma^i_{jk} = \frac{1}{2} g^{im} \left( \frac{\partial g_{mj}}{\partial x^k} + \frac{\partial g_{mk}}{\partial x^j} - \frac{\partial g_{jk}}{\partial x^m} \right).$$

$g_{ij}$ is the metric and $g^{ij}$ is the inverse metric.
There exists a unique vector field on the tangent bundle of manifold, denoted as $TM$, whose trajectories are of the form $(\gamma(t), \gamma'(t))$ where $\gamma$ is the geodesic. Geodesics play an important role in General Relativity, see [25], where the presence of massive bodies distorts the 4-D spacetime and paths of free falling objects are given by the geodesic on the resulting curved 4-D spacetime.

2.2 Conformal mapping

Numerical calculations of the Christoffel symbols involve the inversion of the metric and can be unstable and computationally costly. One strategy to avoid such complications is to calculate the Christoffel symbols in a manifold where the metric is easily inverted and then map the results to the manifold desired. We consider the case where the manifold containing information about the objective function is conformally mapped from $\mathbb{R}^D$, where the metric and the inverse metric is the Kronecker delta. In such a case an analytic expression for the Christoffel symbols is available and the costly matrix inversion is avoided. Conformal mappings of manifolds was used to construct Penrose diagrams in General Relativity [26].

Each local neighborhood in the new manifold is holomorphic to $\mathbb{R}^D$. The resulting metric under the conformal mapping is said to be conformally related to the Euclidean metric

$$g_{ij} = \Psi(x)^2 \delta_{ij},$$

where the scale factor $\Psi(x) = e^{f(x)}$ and $f(x)$ is a real valued objective function. Trivially, the manifolds obtained this way are Riemannian as the metric tensors are both symmetric and positive definite. The existence of such mappings is trivial since we assume the existence of $f(x)$ to begin with.

3 Theoretical properties of the geodesic in a conformally flat metric

This section investigates the path of the geodesic by considering the direction of its tangent vector and the jumping mechanism used in the algorithm. Unless otherwise stated, the Einstein summation convention is used on all quantities in component form.

3.1 The geodesics under conformal mapping

**Definition 3.1** Let $v_t$ be the tangent vector of a geodesic, where $t$ parametrizes the geodesic, and $w$ be the tangent vector of a level curve or the gradient at $x \in \mathbb{R}^D$. The geodesic at $x$ is said to be asymptotically parallel to $w$ at $x$ if there exists $k \in \mathbb{R}$ such that $v_t \to kw$ as $t \to \infty$.

**Theorem 3.1** The geodesics on any manifold conformally related to Euclidean space are asymptotically parallel to either the gradient or the level curves of the objection function $f$.

**Proof** Consider the case where the metric $\tilde{g}_{ij}$ and the Christoffel symbols $\tilde{\Gamma}^i_{jk}$ are mapped conformally to $g_{ij}$ and $\Gamma^i_{jk}$, respectively. Suppose that the new metric is conformally related to the Euclidean metric, i.e. $g_{ij} = \Psi^2 \delta_{ij}$, where $f = \log \Psi$. Using the relation between Christoffel symbols under conformal transformations,\(^1\)

$$\Gamma^i_{jk} = \tilde{\Gamma}^i_{jk} + \tilde{g}^i_j \partial_k f + \tilde{g}^i_k \partial_j f - \tilde{g}_{jk} \tilde{g}^{lm} \partial_m f,$$

\(^1\) We use a shorthand for partial derivatives, $\partial_t \equiv \frac{\partial}{\partial x^t}$.
where $\tilde{g}_{ij} = \delta_{ij}$ and $\tilde{\Gamma}^i_{jk} = 0$ since the metric is conformally flat, we have,

$$\Gamma^i_{jk} = \delta^i_j \partial_k f + \delta^i_k \partial_j f - \delta^i_j \delta^m_k \partial_m f.$$ 

From the $\Gamma^i_{jk} x'(t)^j x'(t)^k$ term in Eq. 1, the following quantities appear in the geodesic equation

$$\delta^i_j (\partial_k f) v^j v^k = (v \cdot \nabla f) v^i,$$

$$\delta^i_j (\partial_j f) v^j v^k = (v \cdot \nabla f) v^i,$$

$$-\delta^i_j \delta^m_k \partial_m f v^j v^k = -||v||^2 \delta^m_k \partial_m f.$$ 

where $v = dx(t)/dt$ is the tangent vector of the geodesic, $||v||^2 = \delta_{ij} v^i v^j$ and $0 < t < t_c/2$. We can think of the problem as a particle moving in Euclidean space under the influence of a force induced by the curvature (through the Christoffel symbols). The geodesic equation becomes

$$\frac{d}{dt} v = ||v||^2 \nabla f - 2(v \cdot \nabla f)v.$$ 

Now, define a local coordinate frame with a basis vector parallel to $\nabla f$ and the rest perpendicular to it. In this local coordinate frame, there exists a plane spanned by $v$ and $\nabla f$. Decomposing the tangent vector on this plane gives $v = (v_{||}, v_{\perp})$, where the component parallel to $\nabla f$ is $v_{||}$ and the one perpendicular to it is $v_{\perp}$. Further, we align the axes so that the gradient and the component of the tangent vector perpendicular to the gradient are always positive, so that $\nabla f = (|\nabla f|, 0)$ and $v = (s|v||, |v||)$, where $s = \text{sign}(v \cdot \nabla f)$ and $\text{sign}() = \pm 1$. The geodesic equation in component form gives

$$\frac{d}{dt} v_{||} = v^2 |\nabla f|(1 - 2|v_{||}|^2),$$

$$\frac{d}{dt} v_{\perp} = -2s|v_{\perp}||v|||\nabla f|,$$

where $v^2 = ||v||^2$. Setting $|v|| = \sqrt{v^2 - |v_{\perp}|^2}$ and keeping $v^2$ constant (we only consider the direction here and this is also to keep the differential equation analytically solvable. Also, we use $v^2 = 1$ in the algorithm), the equation for $|v||$ becomes

$$\frac{d}{dt} v_{\perp} = -2s|v_{\perp}|\sqrt{v^2 - |v||^2}|\nabla f|.$$ 

First we solve for $v_{\perp}$, then use $|v|| = \sqrt{v^2 - v_{\perp}^2}$ to obtain $|v||$. The solutions are

$$v_{||} = v \tanh(2sv|\nabla f| t + vC_1),$$

$$v_{\perp} = v \sech(2sv|\nabla f| t + vC_1),$$

$$C_1 = \frac{1}{v} \sech^{-1}(\frac{v_{\perp}^0}{v}),$$

and $v_{\perp} = |v_{\perp}(t = 0)|$.

There are two qualitatively different behaviors that depend on the sign of $(v \cdot \nabla f)$. Taking $v^2 = 1$ and consider the first derivative of the ratio $|v||/|v_{\perp}|$ at $t = 0$, which is

$$\frac{d}{dt} \left(\frac{|v||}{|v_{\perp}|}\right)_{t=0} = 2s|\nabla f| v_{\perp}^0 \sqrt{(1 - v_{\perp}^2)(1 + v_{\perp}^2)}.$$ 

2 Here, $v$ and $\nabla f$ represent the tangent and gradient vectors in Euclidean space.
Note that $0 < v_{1,0} < 1$ as $v^2 = 1$. We see that the sign is solely determined by $s = \text{sign} (v \cdot \nabla f)$. When the geodesic is in the direction of the gradient ($s = +1$, i.e. approaching a local maximum), its component parallel to the gradient becomes larger while the perpendicular component drops. Therefore after an infinitesimal time step the tangent vector will be more aligned with the gradient. On the other hand, when $s = -1$, the perpendicular component grows while the other drops and the geodesic turns until it becomes perpendicular to the gradient (i.e. on the level surfaces of $f$).

Suppose initially the geodesic travels along the gradient, then it would follow the gradient and pass through the maximum because $s = +1$. Once it passes through, the value of $s$ changes ($s = -1$) and it begins to turn until it becomes parallel to the level surfaces of $f$ and turns towards the gradient (due to finite step sizes). Then it would follow the gradient again and the cycle repeats.

3.2 Line search direction for shadow casting

Here we describe the procedure to obtain the line search direction for the shadow casting scheme. Detailed explanation is presented in Sect. 4.

Let $l$ be the length of the geodesic, $\gamma$. We define the jumping direction to be along the vector

$$\Delta x := \frac{1}{T} \int_\gamma \hat{f}(x)(x - A) \, dx - \frac{1}{T} \int_\gamma (x - A) \, dx,$$

for some fixed point $A$ we call the spotlight, where the integral is over the geodesic and $\hat{f} = f/\sqrt{\sum_t \hat{f}(x_t)^2}$ being the normalized $f$ over $t$. In practice, this is approximated by the sum over all steps along the geodesic

$$\Delta x \approx \frac{1}{T} \sum_{t=1}^T [\hat{f}(x_t)(x_t - A) - (x_t - A)],$$

where $T$ is the total number of steps and $x_t \in \gamma$. This is just the difference of the weighted mean and the mean position vectors along the geodesic.

3.3 Solving the geodesic equation with a quadratic approximation

In this subsection we discuss the quadratic approximation used to solve the geodesic equation iteratively. We give an estimation of the adaptive step sizes to ensure that the approximation is valid.

In the neighborhood of $x_t$, the (discretized) approximation to the solution of the geodesic Eq. (1) is

$$x_{t+1} = x_t + v_t \delta t + c_t (\delta t)^2,$$

where $v_t$ is the unit tangent vector of the geodesic at $x_t$, $\delta t$ is the step size of the geodesic parameter (henceforth the “parametric step size”), and

$$c_t = \frac{1}{2} \frac{dv}{dt} = \frac{1}{2} \left[ \nabla f(x_t) - 2(v_t \cdot \nabla f(x_t))v_t \right].$$

The tangent vector is estimated by the (normalized) difference $x_t - x_{t-1}$ and we set the initial tangent vector to be the gradient, $v_{t-1} = \nabla f(x_{t-1})$. Note that the quadratic approximation (7) is not valid when there exist a component $i$ such that $O(v_{ti}\delta t) \gg O(c_{ti}(\delta t)^2)$, as the
approximation is only accurate when the quadratic term is small. We choose the value of $\delta t$ when the linear term approximately equals to the quadratic term in magnitude $\delta t = t_C$, where

$$t_C = \frac{1}{D} \sum_{i=1}^{D} \frac{v_i}{c_{ii}}.$$  

(9)

Note that $||c_i|| = ||\nabla f(x_t)||/4$ and $v_i \sim O(1)$ for a normalized tangent vector, in regions of large gradients $t_C \propto ||\nabla f||^{-1}$. Consequently the step size also scales with the inverse gradient,

$$||x_{t+1} - x_t|| \propto ||\nabla f||^{-1}.$$  

(10)

The result is intuitive, as the gradient becomes large, the curvature at the point in question on the manifold is large, and a smaller step size is needed so that the quadratic approximation is accurate.

4 Shadow casting algorithm with conformal geodesics

This section illustrates the idea of shadow casting. We denote the location of the spotlight by the point $A$. The algorithm has two phases. A global phase and a local phase. The global phase combines the geodesic and a jumping mechanism to explore the search region. The local phase consists of a local search to pinpoint a nearby maximum and a random walk move to relocate point $A$. In this section we describe the global phase in detail and show conceptually that the method has a less severe dependence on dimensionality. We first describe the behaviors of the geodesics, then we discuss the procedure for shadow casting and describe the scheme of our method. Numerical results that support our claims are shown in the next section.

The global phase comprises of a geodesic search and line search jumping mechanism. Figure 1 demonstrates the estimated geodesic moving through three local maxima. At the beginning, the geodesic was not able to locate a nearby maximum. But once it escapes its basin of attraction, it turns around when it was not able to find another maximum. The same
happens again before it reaches the first maximum in the top-left corner. Once it passes the maximum and escapes its basin, it is able to reach another one in the bottom left. The same process repeats until the third maximum is found.

There are cases where the geodesic fails to reach multiple maxima, for instance, as in Fig. 2. This occurs when $||\nabla f||$ is sufficiently large so that the geodesics turn backwards before it escapes a basin of attraction [c.f. Eq. (5)]. Furthermore, Eq. (10) shows that the geodesic step size is inversely proportional to $||\nabla f||$. In regions of sufficiently large gradients the step size would become impractically small. On the other hand, when the landscape of the objective function is sufficiently flat, the step size would become so large that the geodesic would miss a maximum. In order to control the behaviors of the geodesics, we rescale the gradient in constructing the geodesics, i.e.

$$\nabla f \rightarrow \beta \frac{\nabla f}{||\nabla f||} := \beta \hat{\nabla} f,$$

where $\beta$ follows a log-normal distribution, $\log \beta \sim N(0, 1)$. Each geodesic is assigned a specific value of $\beta$.

The line search direction for the shadow casting scheme is introduced in accordance with Sect. 3.2. When the geodesic is trapped, the line search direction $\Delta x$ will point approximately towards $A$. The situation is illustrated in Fig. 3. The region $S$ contains points $x$ such that $f(x) > f(A)$ with the surrounding dashed region corresponding to the basin of attraction of the maximum contained in $S$. The blue line corresponding to a geodesic trapped inside a local maximum, denoted by the surrounding dashed circle. The arrow represents the direction of line search from the geodesic. The region $C$ denotes the region spanned by the solid angle $\Omega$ subtended by the cross-section of $S$ from point $A$. Up to the line search step size, as long as the geodesic is trapped inside $C$, the line search is guaranteed to discover a point $\hat{x}^*$ in $S$ where $f(\hat{x}^*) > f(A)$. If the solid angle $\Omega$ is large, the region $C$ can cover a substantial region of the search place. This happens when the size of $S$ is large and the global (or any

---

3 Note that $\sum_t f(x_t) \neq 1$ and the vector $\sum_t \hat{f}(x_t)(x_t - A)$ will be approximately parallel to $(x_t - A)$ in Eq. (6).
better) maximum is close to the point $A$. Furthermore, implementing local searches along the line effectively widens the solid angle resulting in a higher probability of reaching $S$. As an example, let $S$ be the volume bounded by a unit sphere in $\mathbb{R}^2$ with center at $(1, 0)$, and point $A$ at $(0, 0)$. Then $C = \mathbb{R}^2$ as every line passing through $A$ must contain a segment inside the unit circle. The same is true in any dimensions.

If we set point $A$ to be the point that gives the best value of $f(x)$ found thus far, then the shadow casting scheme can be viewed as jumping from a maximum to a better one, while forbidding jumps to a lower maximum. The situation is illustrated on the right of Fig. 3. We see that as the number of maxima increases, the number of paths to get to the global maximum also increases, in fact, exponentially. In this case the set $C = \bigcup_{i=1}^{N} C_i$, where $i$ is an index for each maximum and $N$ is the number of maxima better than the best known maximum. In other words the set $C$ is enlarged by each maximum having a better value. This suggests multistart methods may be appropriate for locating the global maximum - even when a bad starting point is chosen on the first try, a different path is likely to be adopted by another starting point which may have a higher probability in reaching the global optimum. In turn leading to a lowered average function evaluations to solve a problem and a higher success rate. Finally, we note that the convergence of our algorithm to the global maximum is guaranteed, as point $A$ can only jump to better and better maxima.

The initial position of point $A$ is set to be the center of the search space. In fact, this is the optimal initial condition given no prior information of the global optimum. This is from the observation that when point $A$ lies at the global optimum the first line search through $A$ would find the solution. And that, in the language of Bayesian statistics, as we have no information on the location of the global optimum, we can assume a non-informative prior distribution for its location, which is just a uniform distribution over the search space. This gives the expected location of the global maximum at the center of the search space. Therefore setting $A$ at the center of the search space initially is expected to be a good choice.

The sketch of the algorithm, Sequential Geodesic Optimization (SGEO), is presented as follows. In essence, sequences of geodesic and line search is repeated with local searches, and restarted with possibly different points of $A$ in each iteration. One iteration is defined as the block in the while loop. The set of input parameters is

- $T$, the number of geodesic steps in one direction,
- $N_{LS}$, the number of steps where local searches are performed along the geodesic,
– \( \alpha_L \) the line search step size,
– \( N^{LS}_L \), the number of steps where local searches are performed along the line,
– \( \sigma \), the random walk jump size for \( A \),
– \( No\text{ImproveMax} = 1000 \), maximum number of iterations that fail to find a better optimum, stopping criterion.

1. Set \( A = \frac{1}{2}(u + v) \) and generate \( x_0 \) uniformly in \([u, v]^D\).
2. while \( No\text{Improve} < No\text{ImproveMax} \)
   (a) Generate \( \beta \sim \exp(\mathcal{N}(0, 1)) \).
   (b) Construct geodesic \( \gamma \) with parameters \((\beta, T, x_0)\) and obtain \( \hat{x}_\gamma^* \) and \( \hat{\Delta}x = \Delta x / ||\Delta x||^2 \).
   (c) Perform line search with step size \( \alpha_L \) along the line \( L \), where
   \[ L = \{ x = \hat{x}_\gamma^* \pm \alpha \hat{\Delta}x \mid \alpha \in \mathbb{R} \text{ s.t. } x \in [u, v]^D \} \]
   and obtain \( \hat{x}_L^* = \arg \max f(x), x \in L \).
   (d) If \( f(\hat{x}_L^*) > f(\hat{x}_\gamma^*) \) go to step 2(a) with starting point \( x_0 = \hat{x}_L^* \) for \( \gamma \), else
      i. Perform local search with starting point \( x_0 = \hat{x}_L^* \) to obtain the optimizer \( x_{LS}^* \).
      ii. If \( x_{LS}^* \) is the same as the one obtained in the last iteration, set \( No\text{Improve} := No\text{Improve} + 1 \). Else set \( No\text{Improve} = 0 \).
      iii. If local search cannot find a better maximum and \( f(x_{LS}^*) > f(A) \), set \( A = x_{LS}^* \) which is the best optimizer to date.
      iv. Finally, perform a random walk jump on \( A \) followed by a local search.\(^5\) Update \( A \) if a better point is found and go back to step 2(a) with \( x_0 \) generated uniformly in \([u, v]^D\).

Being a stochastic algorithm with multiple relaunches, convergence is guaranteed as the number of iterations goes to infinity. As the search trajectory consisting of the geodesics and the shadow casting scheme enters the basin of the global optimum, the rate of convergence is that of the local search method, quasi-Newton in our case once we are are within a reasonable range of the global optimum. Given an arbitrary starting point, the rate of finding the global basin of attraction is dependent on the geometrical shape of the objective function. We do not believe that a uniform convergence rate can be established for all possible objective functions. For the geodesic, it depends on the distribution of basins of attraction within the search or admissible space. For the shadow casting scheme, it depends on the positions of the spotlight \( A_n \) at each iteration \( n \). However, conclusions can be made for specific types of functions. Recall that at the end of each iteration, \( A_n \) is the best solution found thus far. Then for objective functions of the single funnel type, the sequence \( ||A_n - x^*|| \rightarrow 0 \) monotonically as \( n \rightarrow \infty \). Equivalently, the region \( C \) for the global optimum approaches the whole search space monotonically. Therefore the probability of finding the global basin of attraction by shadow casting increases monotonically at each iteration.

5 Numerical experiments

Many generators exist in the literature as performance benchmarks \([28–31]\). We have chosen to use the large-scale differentiable functions from the same test set as in \([15]\), which comprises of problems from \([32,33]\). The reason is twofold. The test set encompasses various classes of test functions spanning over dimensions of 2–50. Also, the same problem set

\(^4\) Note that \( x_{\gamma}^* \in L \).
\(^5\) This is similar to the monotonic basin hopping algorithm \([27]\) but we use a random walk instead of uniform sampling in a \( D \)-sphere.
was also used to test recent DIRECT variants hence we can compare the performance of SGEO-QN\(^6\) directly. We also perform an analysis on the GKLS class of functions. Lastly, we present our results on the sensitivity analysis with five sets of parameters.

### 5.1 Numerical analysis on differentiable large-scale problems

The same definition of success as [15] is used, which occurs when the follow condition is satisfied

\[
f(x^*) - f^* \leq 10^{-4},
\]

where \(x^*\) is the estimated global optimum and \(f^*\) is the known function value at the optimum. Note that in order to compare with a minimization problem, we have negated our function values. For each function, we record the mean number of function/gradient evaluations, \(\bar{N}_{call}\), its standard deviation, \(\sigma_{N_{call}}\) and the worst-case number \(\max(N_{call})\). We placed the results for SGEO-QN in Tables 1. Tables 2 show the number of function evaluations required for the DIRECT algorithms in [15] and function/gradient evaluations for SGEO-QN including those from local searches. Note that in the comparisons the number of gradients evaluation for SGEO-QN is compared against the number of function evaluations for the derivative-free DIRECT variants, leading roughly to a factor of \(D\) difference between the computational cost of the two metrics with gradient evaluations being more costly.\(^7\) Parameters used in Tables 1 to 2 are the set \(P_2\) described in Sect. 5.3. For the CEC13 test problems we used the set \(P_1\), so that a local search is performed on every step on the line search. The recorded results for SGEO-QN are averaged over 50 times with the exception of the Dixon-Price \((D = 50)\), Perm1 \((D = 5)\) and CEC13 functions, which are averaged over 10 times. Also, we set \(\text{NoImproveMax} = 10,000\) for Dixon-Price \((D = 50)\) to ensure a high success rate. Note that composition functions 4–7 contain the Weierstrass function which is continuous but non-smooth everywhere. Therefore we omit the corresponding composition functions in our analysis.

The numerical results show that SGEO-QN can outperform the DIRECT variants on two classes of functions. 1) Functions with many optima. 2) Functions with the global optima very close to a local optimum. Furthermore, when the basin of attraction of the global optimum is large, SGEO-QN scales rather well with increasing dimensionality. We note that our algorithm is more robust in the sense that it is not affected by a rotation of the objective functions as much as the DIRECT variants. The ratio of the number of function evaluations in the CEC13 Rastrigin to Rotated Rastrigin functions is 2.82 for SGEO-QN whereas that for DIRECT variants it is 24.1. Similar observations can be made in the Schwefel functions. This is expected, since SGEO-QN employs an almost\(^8\) coordinate independent calculation. Overall, SGEO-QN was able to solve all but 2 out of 95 differentiable test functions analyzed (the CEC13 Schaffer F6 and F7 functions) with all but five having an 100% success rate, compared to 10, 13 and 1 for DIRECT variants DFO-DIRMIN-TL, BDF-DIRMIN-TL and DIRFOB-TL, respectively. The comparison of function/gradient evaluations is made difficult by the fact that there exists functions that are not successfully optimized by all of the algorithms. However this can be alleviated by omitting the CEC13 problems and consider the reported

---

6 We implement quasi-Newton (QN) local searches on the geodesics \(\gamma\) and the line \(L\). Hence we call the resulting algorithm SGEO-QN.

7 A recent publication tackling the problem of comparing deterministic and stochastic optimization methods is discussed in [34].

8 The only quantity that depends on the coordinate is \(t_C\) in Eq. (9), where a different choice of basis would result in different values in the components.
Table 1  SGEO-QN results

| Function      | D | $N_{call}$ | $\sigma_{N_{call}}$ | $\text{max}(N_{call})$ | $f(x^*)$       |
|---------------|---|------------|---------------------|--------------------------|----------------|
| Schubert      | 2 | 778        | 415                 | 1807                     | $-186.7309$   |
| Schub. Pen 1  | 2 | 4933       | 4802                | 18,616                   | $-186.7309$   |
| Schub. Pen 2  | 2 | 3068       | 2532                | 11,903                   | $-186.7309$   |
| S-H. Camel B. | 2 | 392        | 54                  | 483                      | $-1.0316$     |
| Goldstein Price | 2 | 335       | 57                  | 612                      | 3             |
| Trecanni mod. | 2 | 285        | 27                  | 341                      | 1.010e$-11$   |
| Quartic       | 2 | 20,573     | 20,044              | 103,122                  | $-0.35237$    |
| Shekel $m = 5$ | 4 | 491        | 258                 | 1405                     | $-10.1532$    |
| Shekel $m = 7$ | 4 | 484        | 306                 | 1853                     | $-10.4029$    |
| Shekel $m = 10$ | 4 | 499       | 249                 | 1440                     | $-10.5364$    |
| Espon. Mod.   | 2 | 284        | 34                  | 364                      | $-1$          |
| Espon. Mod.   | 4 | 327        | 46                  | 426                      | $-1$          |
| Cos-mix mod. | 2 | 495        | 248                 | 1453                     | $-0.2$        |
| Cos-mix mod. | 4 | 1626       | 1144                | 5777                     | $-0.4$        |
| Hartman 3     | 3 | 241        | 141                 | 627                      | $-3.8628$     |
| Hartman 6     | 6 | 436        | 280                 | 1501                     | $-3.3224$     |
| 5n loc-min    | 2 | 339        | 93                  | 667                      | 5.316e$-10$   |
| 5n loc-min    | 5 | 553        | 148                 | 1200                     | 1.087e$-08$   |
| 5n loc-min    | 10| 890        | 204                 | 1812                     | 2.933e$-09$   |
| 5n loc-min    | 20| 1518       | 265                 | 2180                     | 1.916e$-07$   |
| 10n loc-min   | 2 | 487        | 243                 | 1417                     | 6.221e$-12$   |
| 10n loc-min   | 5 | 1051       | 468                 | 2712                     | 1.566e$-10$   |
| 10n loc-min   | 10| 1863       | 455                 | 3290                     | 1.156e$-09$   |
| 10n loc-min   | 20| 5428       | 1653                | 13,701                   | 4.082e$-08$   |
| 15n loc-min   | 2 | 1619       | 896                 | 3571                     | 4.584e$-09$   |
| 15n loc-min   | 5 | 2677       | 1606                | 6685                     | 1.195e$-08$   |
| 15n loc-min   | 10| 5085       | 4122                | 23,921                   | 3.616e$-08$   |
| 15n loc-min   | 20| 9981       | 6087                | 32,569                   | 7.340e$-09$   |
| Griewank mod  | 2 | 19,998     | 21,516              | 87,419                   | 6.357e$-12$   |
| Griewank mod  | 5 | 521,744    | 349,267             | 1,346,006                | 2.736e$-08$   |
| Griewank mod  | 10| 1847       | 1234                | 6319                     | 8.313e$-08$   |
| Griewank mod  | 20| 1942       | 723                 | 3162                     | 3.482e$-08$   |
| Pinter        | 2 | 466        | 262                 | 1715                     | 1.043e$-09$   |
| Pinter        | 5 | 1277       | 900                 | 4452                     | 2.642e$-09$   |
| Pinter        | 10| 2207       | 1665                | 7923                     | 2.636e$-09$   |
| Pinter        | 20| 2248       | 1024                | 5307                     | 8.487e$-10$   |
| Griewrot2     | 2 | 5888       | 6366                | 27,691                   | $-179.9926$   |
| Griewrot2     | 10| 15,508     | 11,154              | 50,673                   | $-179.9901$   |
| Griewrot2     | 30| 5153       | 2782                | 9976                     | $-179.9970$   |
| Griewrot2     | 50| 5000       | 2517                | 9571                     | $-179.9911$   |
| Ackley        | 2 | 6122       | 5222                | 22,742                   | 8.686e$-05$   |
| Ackley        | 10| 17,280     | 10,952              | 48,755                   | 5.384e$-05$   |
| Function       | D  | \( N_{\text{call}} \) | \( \sigma N_{\text{call}} \) | \( \max(N_{\text{call}}) \) | \( f(x^*) \) |
|---------------|----|------------------|-----------------|-----------------|----------|
| Ackley        | 25 | 40,878           | 22,215          | 11,259          | 7.550e−05|
| Ackley        | 50 | 43,713           | 22,412          | 126,600         | 5.489e−05|
| DixonPrice    | 2  | 341              | 52              | 503             | 1.141e−11|
| DixonPrice    | 10 | 2115             | 1394            | 6503            | 3.262e−09|
| DixonPrice    | 25 | 140,002          | 147,424         | 771,112         | 5.066e−08|
| DixonPrice    | 50 | 37,259,283       | 41,089,123      | 126,527,432     | 5.662e−08|
| Easom         | 2  | 908              | 625             | 2915            | −1       |
| Michalewics   | 2  | 1054             | 823             | 3769            | −1.8013  |
| Michalewics   | 5  | 36,113           | 24,830          | 122,661         | −4.6877  |
| Michalewics   | 10 | 2,550,847        | 4,032,534       | 14,677,922      | −9.6602  |
| Rastrigin     | 2  | 1855             | 1344            | 6827            | 1.391e−08|
| Rastrigin     | 10 | 2189             | 1038            | 5028            | 2.415e−12|
| Rastrigin     | 30 | 2714             | 1337            | 7810            | 2.842e−13|
| Rastrigin     | 50 | 3616             | 1864            | 13,591          | 3.856e−13|
| Beale         | 2  | 360              | 51              | 499             | 7.803e−10|
| Bohachevsky1  | 2  | 646              | 349             | 1523            | 8.356e−13|
| Bohachevsky2  | 2  | 588              | 235             | 1447            | 5.754e−10|
| Bohachevsky3  | 2  | 450              | 120             | 899             | 2.145e−11|
| Booth         | 2  | 296              | 35              | 391             | 5.673e−12|
| Colville      | 4  | 715              | 149             | 1305            | 7.259e−12|
| Perm1         | 2  | 341              | 37              | 407             | 2.865e−11|
| Perm1         | 5  | 10,085,123       | 7,242,795       | 19,447,930      | 2.611e−05|
| Perm2         | 2  | 347              | 42              | 432             | 1.067e−13|
| Perm2         | 5  | 1276             | 655             | 3610            | 1.633e−05|
| Powell        | 4  | 490              | 55              | 610             | 8.072e−09|
| Powell        | 8  | 852              | 138             | 1159            | 7.110e−08|
| Powell        | 16 | 1595             | 228             | 1998            | 1.044e−07|
| Powell        | 24 | 2495             | 288             | 3142            | 2.626e−07|
| Powersum      | 4  | 875              | 290             | 1604            | 1.361e−05|
| Schwefel      | 2  | 2452             | 2370            | 12,963          | 2.882e−07|
| Schwefel      | 5  | 15,340           | 8118            | 36,984          | 6.269e−05|
| Schwefel      | 10 | 53,781           | 31,818          | 131,343         | 7.868e−06|
| Schwefel      | 20 | 117,831          | 49,672          | 223,350         | 4.863e−05|
| CEC13 FUNCTIONS [33] |
| Sphere        | 10 | 1126             | 190             | 1306            | −1400    |
| Rot Elliptic  | 10 | 9712             | 1603            | 11,264          | −1299.996|
| Rot discus    | 10 | 98,739           | 59,044          | 247,293         | −1199.94 |
| Rot Bent Cigar| 10 | 4880             | 1270            | 6797            | −1099.992|
| Different Powers | 10 | 10,027           | 1415            | 12,346          | −1000    |
| Rot Rosenbrock| 10 | 10,671           | 2237            | 13,791          | −900     |
| Rot Schaffers (F7) | 10 | 32,417,595*     | N/A             | N/A             | −799.780 |
| Rot Ackley    | 10 | 1,072,573        | 488,116         | 1,815,822       | −699.966 |
Table 1 continued

| Function                  | D  | $\bar{N}_{call}$ | $\sigma_{N_{call}}$ | $\max(N_{call})$ | $f(x^*)$       |
|---------------------------|----|------------------|---------------------|-------------------|----------------|
| Rot Griewank              | 10 | 15,330           | 3534                | 19,708            | $-499.977$      |
| Rastrigin                 | 10 | 397,366          | 153,339             | 707,675           | $-400$          |
| Rot Rastrigin             | 10 | 140,722          | 32,110              | 197,939           | $-300$          |
| Non-continuous rot. Rastrigin | 10 | 1,366,913        | 1,078,038           | 4,104,849         | $-200$          |
| Schwefel                  | 10 | 12,065,174       | 7,754,747           | 30,362,498        | $-100$          |
| Rot Schwefel              | 10 | 6,545,621        | 3,924,463           | 13,558,876        | 100             |
| Expanded Griewank+ Rosenbrock | 10 | 5,343,742        | 7,121,904           | 20,286,621        | 500.032         |
| Expanded Schaffer (F6)    | 10 | 33,020,382*      | N/A                 | N/A               | 600.060         |
| Comp function 1           | 10 | 3,674,569 (90%)  | 3,929,466           | 12,068,553        | 710.005         |
| Comp function 2           | 10 | 5,418,960 (70%)  | 3,483,559           | 117,358,874       | 830.001         |
| Comp function 3           | 10 | 4,338,355 (30%)  | 1,548,661           | 6,501,845         | 970.001         |
| Comp function 8           | 10 | 16,239,405 (20%) | 5,292,288           | 21,714,538        | 1480.005        |

The mean number of function/gradient calls is denoted by $\bar{N}_{call}$, $\sigma_{N_{call}}$ denotes the standard deviation of the averaged number of function evaluations for each $n_f$, and $\max(N_{call})$ denotes the maximum number evaluations. The success rates are 100% except for comp functions, with the respective success rates listed in parentheses. The standard deviations and worst-case numbers are not available (denoted by “N/A”) for the CEC Rot. Schaffer and the Expanded Schaffer functions as SQEO-QN were unable to solve these functions.

function evaluations for failures for DIRECT variants as a lower bound. Considering only the non-CEC13 test set, SQEO-QN can solve every problem and the averaged function/gradient evaluations for SQEO-QN is $1.86 \times 10^5$. The lower bound for function evaluations is $3.21 \times 10^7$ for DFO-DIRMIN-TL. Averaging over the “hard” non-CEC13 problem, we get $4.39 \times 10^5$ for SQEO-QN, $7.54 \times 10^7$, $1.80 \times 10^6$ and $3.01 \times 10^8$ for lower bounds of DFO-DIRMIN-TL, BDF-DIRMIN-TL and DIRFOB-TL, respectively. In summary, the average number of function/gradient evaluations required for SQEO-QN is less than those for the DIRECT variants over the non-CEC13 problem set.

5.1.1 Functions with many optima

Examples of functions with many optima include Schubert, Griewank, Rastrigin, Michalewics, Easom, Pinter, Ackley, Schwefel and most of the CEC13 functions. Numerical results show that SQEO-QN can outperform the DIRECT variants in the majority of this class of functions, especially in higher dimensions.

We observe that the dependence on dimensionality is less than exponential in some cases. For the Schwefel functions, where the number of function/gradient evaluations grows approximately linearly with the dimension. In the more extreme cases such as the Rastrigin and Pinter functions, where the increase is very small. This occurs when the initial position for the spotlight point $A$ is such that the region $C$ discussed in Sect. 4 extends to most of the search space. So that the functions are optimized by the first few line searches with local searches through $A$. The Rastrigin function is a special case, where the initial point $A$ lies at the global optimum and the region $C$ is the whole search space.

Remarkably, SQEO-QN is able to find the global optimum of the CEC13 non-continuous rotated Rastrigin function with less number of function/gradient evaluations than DIRECT variants. The average function/gradient evaluations required is higher than that for the other
Table 2  Number of function evaluations required to find the optimum for the derivative free DIRECT variants in [15] and function/gradient evaluations for SGEO-QN

| f         | D | Derivative-free DIRECT variants | SGEO-QN |
|-----------|---|---------------------------------|---------|
|           |   | DFO-DIRMIN-TL                  | BDF-DIRMIN-TL | DIRFOB-TL |
| Schubert  | 2 | 365                             | 109      | 2315      | 778      |
| Schub. Pen 1 | 2 | 2520                            | 235      | 6316      | 4933     |
| Schub. Pen 2 | 2 | 881                             | 193      | 3868      | 3068     |
| S-H. Camel B. | 2 | 75                              | –        | –         | 392      |
| Goldstein Price | 2 | 107                             | –        | –         | 335      |
| Trecanni mod. | 2 | 78                              | –        | –         | 285      |
| Quartic   | 2 | 499                             | 96       | 84        | 20,593   |
| Shekel m = 5 | 4 | 142                             | –        | –         | 491      |
| Shekel m = 7 | 4 | 500                             | 126      | 164       | 484      |
| Shekel m = 10 | 4 | 1005                            | 142      | 160       | 499      |
| Espon. Mod. | 2 | 76                              | –        | –         | 284      |
| Espon. Mod. | 4 | 150                             | –        | –         | 327      |
| Cos-mix mod. | 2 | 70                              | –        | –         | 495      |
| Cos-mix mod. | 4 | 138                             | –        | –         | 1626     |
| Hartman 3 | 3 | 105                             | –        | –         | 241      |
| Hartman 6 | 6 | 229                             | –        | –         | 436      |
| 5n loc-min | 2 | 62                              | –        | –         | 339      |
| 5n loc-min | 5 | 152                             | –        | –         | 553      |
| 5n loc-min | 10 | 302                             | –        | –         | 890      |
| 5n loc-min | 20 | 602                             | –        | –         | 1518     |
| 10n loc-min | 2 | 62                              | –        | –         | 487      |
| 10n loc-min | 5 | 152                             | –        | –         | 1051     |
| 10n loc-min | 10 | 302                             | –        | –         | 1863     |
| 15n loc-min | 20 | 602                             | –        | –         | 5428     |
| 15n loc-min | 2 | 62                              | –        | –         | 1619     |
| 15n loc-min | 5 | 152                             | –        | –         | 2677     |
| 15n loc-min | 10 | 302                             | –        | –         | 5085     |
| 15n loc-min | 20 | 602                             | –        | –         | 9981     |
| Griewank mod | 2 | 78,663                          | 8072     | 14,746    | 19,998   |
| Griewank mod | 5 | 490,515                         | 274,190  | 154,689   | 521,744  |
| Griewank mod | 10 | 411,178                        | 14,724   | 182,128   | 1847     |
| Griewank mod | 20 | 942                             | –        | –         | 1942     |
| Pinter    | 2 | 170                             | 197      | 90        | 466      |
| Pinter    | 5 | 25,618                          | 1843     | 10,547    | 1277     |
| Pinter    | 10 | 444,607                         | 16,393   | 24,151,879| 2207     |
| Pinter    | 20 | 42,682,351                      | 10,7872  | 822,928,421 | 2248     |
| Griewrot2 | 2 | 80                              | –        | –         | 5888     |
| Griewrot2 | 10 | 816                             | –        | –         | 15,508   |
| Griewrot2 | 30 | 5063                            | –        | –         | 5153     |
| Griewrot2 | 50 | 10,205                          | 8468     | 5104      | 5000     |
| $f$          | D | Derivative-free DIRECT variants | SGEO-QN |
|-------------|---|---------------------------------|--------|
|             |  | DFO-DIRMIN-TL                  | BDF-DIRMIN-TL | DIRFOB-TL |       |
| Ackley      | 2 | 3348                            | 407     | 466       | 6122   |
| Ackley      | 10| 412                             | –       | –         | 17,280 |
| Ackley      | 25| 1232                            | –       | –         | 40,878 |
| Ackley      | 50| 2052                            | –       | –         | 43,713 |
| Dixon-Price | 2 | 87                              | –       | –         | 341    |
| Dixon-Price | 10| 178,195                         | 12,464  | 137,441   | 2115   |
| Dixon-Price | 25| 283,077,766                     | 5,492,616* | 5,731,881* | 140,002 |
| Dixon-Price | 50| 36,182,567*                     | 1,940,253* | 1,269,313,104 | 37,259,283 |
| Easom       | 2 | 131,165                         | 11,043  | 318,161   | 908    |
| Michalewics | 2 | 69                              | –       | –         | 1054   |
| Michalewics | 5 | 130,058                         | 9538    | 23,308    | 36,113 |
| Michalewics | 10| 21,699,660                      | 67,724  | 30,065,636 | 2,550,847 |
| Rastrigin   | 2 | 336                             | 181     | 1856      | 1855   |
| Rastrigin   | 10| 12,751                          | 1188    | 12,010    | 2189   |
| Rastrigin   | 30| 280,683                         | 13,542  | 6145      | 2714   |
| Rastrigin   | 50| 1,265,672                       | 51,820  | 10,277    | 3616   |
| Beale       | 2 | 137                             | –       | –         | 360    |
| Bohachevsky1| 2 | 96                              | –       | –         | 646    |
| Bohachevsky2| 2 | 96                              | –       | –         | 588    |
| Bohachevsky3| 2 | 122                             | –       | –         | 450    |
| Booth       | 2 | 75                              | –       | –         | 296    |
| Colville    | 4 | 225,066                         | 1335    | 13,120,363 | 715    |
| Perm1       | 2 | 272                             | –       | –         | 341    |
| Perm1       | 5 | 1,568,231,384                   | 36,624,683* | 7,279,772,213* | 10,085,123 |
| Perm2       | 2 | 62                              | –       | –         | 347    |
| Perm2       | 5 | 57,766                          | 5275    | 168,322,925 | 1276   |
| Powell      | 4 | 141                             | –       | –         | 490    |
| Powell      | 8 | 286                             | –       | –         | 852    |
| Powell      | 16| 576                             | –       | –         | 1595   |
| Powell      | 24| 866                             | –       | –         | 2495   |
| Powersum    | 4 | 127                             | –       | –         | 875    |
| Schwefel    | 2 | 1624                            | 515     | 525       | 2452   |
| Schwefel    | 5 | 201,763                         | 11,670  | 91,233    | 15,340 |
| Schwefel    | 10| 24,909,898                      | 822,302 | 483,319   | 53,781 |
| Schwefel    | 20| 431,426,380                     | 12,217,489 | 4,295,690 | 117,831 |
| CEC13 FUNCTIONS [33] | |       |        |           |       |
| Sphere      | 10| 573                             | –       | –         | 1126   |
| Rot Elliptic| 10| 5,642,213,969                   | 581,930,281 | 121,196,868,750 | 9712   |
| Rot discus  | 10| 47,680                          | 18,501  | 2,611,405 | 98,739 |
| Rot Bent Cigar | 10| 10,584,086,678* | 122,143,170* | 4,059,627,002 | 4880   |
| Different Powers | 10| 589                             | –       | –         | 1,0027 |
Table 2  continued

| \( f \)             | D     | DFO-DIRMIN-TL | BDF-DIRMIN-TL | DIRFOB-TL | SGEO-QN |
|---------------------|-------|---------------|---------------|-----------|---------|
| Rot Rosenbrock      | 10    | 36,686        | 96,883        | 110,939   | 10,671  |
| Rot Schaffers (F7)  | 10    | 374,242,440   | 12,735,661    | 425,933,887| 32,417,595*|
| Rot Ackley          | 10    | 943,365,696*  | 15,935,575*   | 650,470,093| 1,072,573|
| Rot Griewank        | 10    | 14,487        | 16,384        | 22,073    | 15,330  |
| Rastrigin           | 10    | 52,620,517    | 22,445,131    | 202,714,074| 397,366 |
| Rot Rastrigin       | 10    | 3,208,062,726*| 87,462,231*   | 8,398,182 | 140,722 |
| Non-continuous rot. Rastrigin | 10  | 1,939,249,826*| 83,304,904*   | 3,864,394 | 1,366,913|
| Schwefel            | 10    | 1,597,692,027*| 353,923,387*  | 186,574,187| 12,065,174|
| Rot Schwefel        | 10    | 189,035,848*  | 5,107,995*    | 546,425,335| 6,545,621|
| Griewank+Rosenbrock | 10    | 12,087,229    | 1,149,771*    | 31,844,643| 5,343,742|
| Expanded Schaffer (F6) | 10  | 2,396,769,984*| 685,417,439*  | 54,060,331| 3,302,038*|
| Comp function 1     | 10    | 104,003       | 29,144        | 196,612   | 36,74,569|
| Comp function 2     | 10    | 1,655,305,428*| 362,203,872*  | 189,617,796| 5,418,960|
| Comp function 3     | 10    | 265,301,785*  | 4,701,450*    | 97,308,583| 4,338,355|
| Comp function 8     | 10    | 33,824,697    | 2,231,849*    | 1,541,217| 16,239,405|

The dashes correspond to cases where the function is minimized by a single local minimization and therefore considered “easy” and not investigated further in [15]. The asterisks next to the results denote that the respective algorithm is not able to find the global optimum after the recorded function evaluations. Best results are indicated in bold.

CEC13 Rastrigin functions, however. The reason is most likely that when the geodesics cross the discontinuities, the function values along the geodesics change suddenly so that the probability of the line search going through point \( A \) is reduced.

The CEC13 Schwefel functions requires much more function/gradient evaluations for SGEO-QN compared to functions with a single funnel such as the CEC13 Ackley or the CEC13 Rastrigin. This is expected, as the optima of the funnels in the Schwefel functions are far apart from each other. During our analysis we observed that the record minimum (that is, the best function value found by SGEO-QN thus far) stays at \(-93.2950\) and \(-96.6475\) for hundreds of iterations before SGEO-QN locates the next optimum.

Regarding the Ackley functions, the analysis in [15] used a skewed search space \([-15, 30]^D\) and we follow their work. This caused a shift in the initial position for \( A \) from \([0, 0]\) and effects of the optimal initial placement for \( A \) discussed in the previous paragraph is no longer observed. However, from \( D = 10 \) to \( D = 50 \) the dependence on dimensionality appears to be better than exponential still. In the CEC13 rotated Ackley function, SGEO-QN successfully located the global optimum using about 606 times fewer number of function/gradient evaluation than DIRFOB-TL which translates to about 60 times efficient in computational cost, while the other two DIRECT variants are unable to find the global optimum.

5.1.2 Functions with global optimum very close to local optima

Examples of test functions where the global optimum lies very close to a local optimum include Dixon-Price, Perm1, Perm2, and Coville [35]. Numerical results show that SGEO-QN
outperform the DIRECT variants even when $D \geq 5$. Especially in higher dimensions, such as Dixon-Price ($D = 50$) and perm1 ($D = 5$), the differences in number of function/gradient evaluations between SGEO-QN and the DIRECT variants can be as large as 2–3 orders of magnitude, and can be at least an order of magnitude efficient in computational cost for the perm1 ($D = 5$) function. SGEO-QN exhibits an exponential increase in number of function/gradient evaluations with dimensionality but the effect is much less severe compared to the DIRECT variants.

5.1.3 Functions with small basins of attraction

Our algorithm does not perform as well in cases when the global optimum has a small basin of attraction. In these cases the dependence on the dimensionality is exponential. However in these cases SGEO-QN can still outperforms DIRECT variants in some cases. For instance, the Michalewicz functions and the composition functions in CEC13, where the algorithm can get trapped in the global optimum of the second best basic function $g_2$. Also, it is inefficient when the parameters are set such that line search step size is much larger than the characteristic distance between optima (which may vary from region to region). This is demonstrated in the analysis of the CEC13 Schaffer functions (F6 and F7). The algorithm would converge very close to the optimum but fails to progress further. We expect an adaption scheme that changes the step size and the number of local searches along the line search would assist in finding the global optimum for such functions.

5.2 Analysis on the GKLS class of functions

The GKLS class of functions presented in [17] has 100 functions for each dimension and difficulty (“Simple” or “Hard”). We test SGEO-QN using parameter set $P_2$ on the set of functions generated by the GKLS generator [28]. The function set spans over dimensions $D = 2$ to $D = 5$ for both difficulties. Each function is optimized five times in our analysis, so that a total of $4 \times 2 \times 100 \times 5 = 4000$ optimizations are performed. The same stopping criterion as [17] is used - the algorithm is stopped when the function/gradient evaluations exceed $10^6$. Note that [17] has a different definition of success that is dependent on vertices and volumes of rectangular partitions than we employ. For each dimension $D$ and difficulty we record the averaged number of function/gradient evaluations and success rates. The results are shown in Table 3 and 4. The comparison in computational cost can be directly inferred from the recorded number of function/gradient evaluations here as both SGEO-QN and the algorithm in [17] both perform gradient calculations.

The results are interesting that the dependence on dimensionality is non-linear for “simple” problems. The number of function/gradient evaluations increases from dimensions two to four, but drops when the dimension is increased to 5 while having roughly the same success rates. However, the performance of SGEO-QN is more dependent on the size of the global basin. For each dimension, the number of function/gradient evaluations increases and the success rates drop when the problems become “hard” from “simple”. Furthermore, the GKLS class of functions has a paraboloid with basin of attraction extending to the whole search space, excluding small regions modified to allow for other optima. Especially in higher dimensions, the geodesic would frequently revisit the optimum of the paraboloid and miss the global optimum. When the geodesic searches fail, the performance of SGEO-QN is mostly

9 The composition functions have the form $f(x) = \sum_i \omega_i g_i(x)$, where $i$ denotes the $i$-th basic function $g_i(x)$ and its weight $\omega_i \in [0, 1]$, such that $\sum_i \omega_i = 1$. 

© Springer
Table 3  SGEO-QN results for the GKLS class of functions

| D  | Simple | Hard   |
|----|--------|--------|
|    | $\tilde{N}_{call}$ | $\sigma_{N_{call}}$ | max($N_{call}$) | $\tilde{N}_{call}$ | $\sigma_{N_{call}}$ | max($N_{call}$) |
| P2 |        |        |        | P4        |        |        |
| 2  | 2259 (100%) | 1349 | 9755 | 6055 (100%) | 5455 | 49,768 |
| 3  | 12,564 (100%) | 8626 | 38,004 | 28,463 (100%) | 24,426 | 152,209 |
| 4  | 103,739 (99%) | 93,628 | 424,102 | 187,575 (92.8%) | 141,374 | 562,571 |
| 5  | 66,336 (99.2%) | 83,931 | 511,685 | 191,156 (80%) | 121,027 | 519,008 |
| P4 |        |        |        | P2        |        |        |
| 2  | 2161 (100%) | 1557 | 12,847 | 5853 (100%) | 5450 | 33,577 |
| 3  | 14,430 (100%) | 12,718 | 76,978 | 29,937 (100%) | 27,660 | 162,728 |
| 4  | 102,678 (98%) | 92,974 | 623,026 | 186,297 (90%) | 130,579 | 506,445 |
| 5  | 61,064 (99.6%) | 67,387 | 331,455 | 175,124 (80%) | 126,993 | 576,298 |

Only two functions were unable to be solved after $10^6$ function/gradient evaluations in any of the five attempts. These were “hard” problems with \(D = 4, n_f = 30\) and \(D = 5, n_f = 61\), respectively, where \(n_f\) denotes the GKLS function number. The parameter sets \(P_2\) (top) and \(P_4\) (bottom) in Table 5 were used. The quantities in parentheses denote the averaged success rate, \(\sigma_{N_{call}}\) denotes the standard deviation of the averaged number of function evaluations for each \(n_f\), and max($N_{call}$) denotes the maximum number evaluations.

Table 4  Comparison in the mean number of function/gradient evaluations between SGEO-QN and the results with different criteria as defined in [17], denoted by DIRECT-KS, C1 and C3, where C1 represents the maximal number of function/gradient evaluations over each class and C3 is the average

| D  | DIRECT-KS | SGEO-QN |
|----|------------|---------|
|    | C1 | C3 | P2 | P4 |
| Simple |     |    |    |    |
| 2  | 335 | 97.22 | 2259 | 2161 |
| 3  | 2043 | 491.28 | 12,564 | 14,430 |
| 4  | 16,976 | 3675.84 | 103,739 | 102,678 |
| 5  | 16,300 | 3759.05 | 66,336 | 61,064 |
| Hard |     |    |    |    |
| 2  | 1075 | 192.00 | 6055 | 5853 |
| 3  | 2352 | 618.32 | 28,463 | 29,937 |
| 4  | 20,866 | 5524.77 | 187,575 | 186,297 |
| 5  | 88,459 | 22,189.47 | 191,156 | 175,124 |

Best results are indicated in bold

dependent on the shadow casting scheme. Indeed, the enlarged region \(C\) corresponding to the global optimum is smaller in the “hard” problems than in the “simple” problems due to the basins of the global optima having a smaller radius. The increase in number of function/gradient evaluations and the drop in success rates show that this is indeed the case. We note that the functions tested here have few local minima (in fact, just ten) with small basins of attraction. Comparing the results between parameter sets \(P_2\) and \(P_4\) we see that \(P_4\) obtains a slightly better performance, especially in higher dimensions. Indeed, the algorithm proposed in [17] outperforms SGEO-QN on the GKLS functions tested. We expect that combining SGEO-QN with a partitioning scheme on the search space would yield better results on the GKLS class of functions.
5.3 Sensitivity analysis

We chose five different sets of parameters for the sensitivity analysis,

\[ P_1 = \left\{ T = 40, N^V_{LS} = 20, \alpha_L = \frac{\Lambda}{80}, N^L_{LS} = 1, \sigma = \frac{\Lambda}{10} \right\}, \]

\[ P_2 = \left\{ T = 40, N^V_{LS} = 20, \alpha_L = \frac{\Lambda}{200}, N^L_{LS} = 40, \sigma = \frac{\Lambda}{10} \right\}, \]

\[ P_3 = \left\{ T = 40, N^V_{LS} = 20, \alpha_L = \frac{\Lambda}{200}, N^L_{LS} = 80, \sigma = \frac{\Lambda}{10} \right\}, \]

\[ P_4 = \left\{ T = 80, N^V_{LS} = 40, \alpha_L = \frac{\Lambda}{200}, N^L_{LS} = 40, \sigma = \frac{\Lambda}{10} \right\}, \] and

\[ P_5 = \left\{ T = 40, N^V_{LS} = 20, \alpha_L = \frac{\Lambda}{200}, N^L_{LS} = 40, \sigma = \frac{\Lambda}{100} \right\}, \]

where the component-wise minimum range of the box bound is \( \Lambda = \min_c (v - u) \).

We recommend that a larger choice of \( T \) to be used on functions with few optima as in \( P_4 \) and a larger number of local searches to be used on functions with many optima, for

Table 5  Number of function/gradient evaluations required for selected functions with various parameter sets

| \( f(x) \) | \( D \) | \( P_1 \) | \( P_2 \) | \( P_3 \) | \( P_4 \) | \( P_5 \) |
|-------------|-----|------|------|------|------|------|
| Schubert    | 2   | 2278 | 847  | 883  | 763.1 | 832  |
| Schub. Pen 1| 2   | 4300 | 5197 | 6468 | 4716 | 5321 |
| Schub. Pen 2| 2   | 3852 | 3549 | 4901 | 4141 | 4842 |
| Shekel m = 7| 4   | 3955 | 540  | 894  | 401  | 493  |
| Shekel m = 10| 4  | 3757 | 612  | 10,596| 487  | 470  |
| 5n loc-min  | 10  | 8630 | 973  | 964  | 735  | 864  |
| 10n loc-min | 10  | 25,244 | 2380 | 1768 | 1901 | 2109 |
| 15n loc-min | 10  | 29,230 | 5310 | 4243 | 4918 | 4720 |
| Griewank mod| 5   | 196,434 | 583,483 | 1,001,159| 439,297 | 589,192 |
| Pinter      | 5   | 8581 | 1294 | 1622 | 1204 | 1301 |
| Pinter      | 10  | 13,755 | 2186 | 2525 | 2371 | 1968 |
| Pinter      | 20  | 20,660 | 2235 | 2726 | 3279 | 2130 |
| Dixon-Price | 10  | 14,130 | 4518 | 10,689| 1641 | 2161 |
| Dixon-Price | 25  | 194,430 | 123,272 | 2,646,136| 269,607 | 185,198 |
| Schwefel    | 10  | 81,941 | 40,211 | 716,080 | 56,386 | 49,102 |
| Schwefel    | 20  | 201,228 | 94,296 | 1,483,708| 119,498 | 132,030 |

\( n_f = 1 \)

| \( n_f = 1 \) | \( n_f = 1 \) | \( n_f = 1 \) |
|---------------|---------------|---------------|
| 2             | 3             | 4             |
| 4981          | 76,197        | 230,319       |
| 4871          | 49,138        | 234,170       |
| 4380          | 56,353        | 440,944       |
| 4432          | 38,150        | 138,627       |

\( \text{Normalized average} \)

| \( f(x) \) | \( D \) | \( P_1 \) | \( P_2 \) | \( P_3 \) | \( P_4 \) |
|-------------|-----|------|------|------|------|
| Schubert    | 2   | 2278 | 847  | 883  | 763.1 | 832  |
| Schub. Pen 1| 2   | 4300 | 5197 | 6468 | 4716 | 5321 |
| Schub. Pen 2| 2   | 3852 | 3549 | 4901 | 4141 | 4842 |
| Shekel m = 7| 4   | 3955 | 540  | 894  | 401  | 493  |
| Shekel m = 10| 4  | 3757 | 612  | 10,596| 487  | 470  |
| 5n loc-min  | 10  | 8630 | 973  | 964  | 735  | 864  |
| 10n loc-min | 10  | 25,244 | 2380 | 1768 | 1901 | 2109 |
| 15n loc-min | 10  | 29,230 | 5310 | 4243 | 4918 | 4720 |
| Griewank mod| 5   | 196,434 | 583,483 | 1,001,159| 439,297 | 589,192 |
| Pinter      | 5   | 8581 | 1294 | 1622 | 1204 | 1301 |
| Pinter      | 10  | 13,755 | 2186 | 2525 | 2371 | 1968 |
| Pinter      | 20  | 20,660 | 2235 | 2726 | 3279 | 2130 |
| Dixon-Price | 10  | 14,130 | 4518 | 10,689| 1641 | 2161 |
| Dixon-Price | 25  | 194,430 | 123,272 | 2,646,136| 269,607 | 185,198 |
| Schwefel    | 10  | 81,941 | 40,211 | 716,080 | 56,386 | 49,102 |
| Schwefel    | 20  | 201,228 | 94,296 | 1,483,708| 119,498 | 132,030 |

In the last row, the normalized average is defined as \( \frac{\sum_i N^i_{\text{call}} / D_i}{D_i} \) for the \( i \)-th function, where \( N^i_{\text{call}} \) is the number of evaluations and \( D_i \) is the dimensionality for the \( i \)-th function, respectively. Best results are indicated in bold.

© Springer
instance, with $P_{1,2}$. The precise values of the parameter need to be tuned which we leave to future research. The results of the sensitivity analysis is shown in Table 5. For $P_1$, which was used for the CEC13 functions, it appears that the optimal number of local searches for functions with few local optima is relatively low. This is intuitive, as any more local searches would just rediscover known optima. In functions with many local optima such as the Schubert and Griewank functions, a high number of local searches results in less function/gradient evaluations. An exception is illustrated in the Pinter functions, when the region $S$ is sufficiently large the line search is most likely to pass through the basin of the global optimum.

The other parameter sets are obtained from $P_2$. Compared to $P_2$, the number of local searches on the line $L$ is twice as less in $P_3$ which leads to an increase in the number of function/gradient evaluations required overall. In $P_4$, the length of geodesic is doubled while keeping the number of local searches from the geodesic the same. In this case, we see the having a longer geodesic can be beneficial in cases with few optima, such as the Shekel functions and the GKLS functions. Finally in $P_5$ we used a smaller random walk step for the point $A$.

6 Discussion and conclusion

We proposed a method, SGEO-QN, for black box optimization with geodesics on a Riemannian manifold defined by the objective functions. Numerical results have shown that the proposed algorithm exhibits better scaling with dimensionality and less affected by rotations. Specifically, we found that SGEO-QN is well suited to solve problems that the recent DIRECT algorithms considered in this paper have difficulties on, such as those with many local optima. Therefore SGEO-QN should be a good alternative to DIRECT type algorithms.

The current algorithm would benefit from an adaptive scheme to tune the parameters. Problems SGEO-QN failed in solving were those having basins of the global optima much smaller than the step sizes. We believe that having a tunable step size would greatly assist in finding the global optima for such functions as well as a partition scheme for the search region. We leave this to future work.

Acknowledgements

We thank the anonymous reviewers for extremely useful feedbacks in assisting the writing of this manuscript.

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

Funding This research is supported by the Discovery Grants and Discovery Accelerator Supplement from Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Diener, I.: Handbook of Global Optimization, Vol. 2 of the Series Nonconvex Optimization and its Applications, pp. 649–668 (1995)
2. Floudas, C.A., Pardalos, P. M.: Encyclopedia of Optimization, Globally Convergent Homotopy Methods, pp. 1272–1277
3. Chow, S.N., Mallet-Paret, J., Yorke, J.A.: Finding zeros of maps: homotopy methods that are constructive with probability one. Math. Comput. 32, 887899 (1978)
4. Watson, L.T.: Globally convergent homotopy algorithms for nonlinear systems of equations. Nonlinear Dyn. 1, 143–191 (1990)
5. Dunlavy, D.M., O’Leary, D.P.: Homotopy Optimization Methods for Global Optimization, Sandia National Laboratories, Report SAND 2005–7495 (2005)
6. Botsaris, C.A.: Constrained optimization along geodesics. J. Math. Anal. Appl. 79(2), 295–306 (1981)
7. Smith, S.T.: Geometric Optimization Methods for Adaptive Filtering. Harvard University, Cambridge (1993)
8. Smith, S.T.: Optimization techniques on Riemannian manifolds. Fields Inst. Commun. 3(3), 113–135 (1994)
9. Rapcsák, T.: Convex Programming on Riemannian Manifolds, Lecture Notes in Control and Information Sciences, vol. 84, pp. 733–740. Springer, Berlin (1986)
10. Rapcsák, T.: Geodesic convexity in nonlinear optimization. J. Optim. Theory Appl. 69(1), 169–183 (1991)
11. Rapcsák, T.: Geodesic convexity on $\mathbb{R}^n_+$. Optimization 37(4), 341–355 (1996)
12. Rapcsák, T.: Local convexity on smooth manifolds. J. Optim. Theory Appl. 127(1), 165–176 (2005)
13. Perttunen, C.D., Jones, D.R., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. J. Optim. Theory Appl. 79(1), 157–181 (1993)
14. Liuzzi, G., Lucidi, S., Piccialli, V.: A DIRECT-based approach exploiting local minimalizations for the solution of large-scale global optimization problems. Comput. Optim. Appl. 45(2), 353–375 (2010)
15. Liuzzi, G., Lucidi, S., Piccialli, V.: Exploiting derivative-free local searches in DIRECT-type algorithms for global optimization. Comput. Optim. Appl. (2015). doi: 10.1007/s10589-015-9741-9
16. Kvasov, D.E., Sergeyev, Y.D.: A univariate global search working with a set of Lipschitz constants for the first derivative. Optim. Lett. 3(2), 303–318 (2009)
17. Kvasov, D.E., Sergeyev, Y.D.: Lipschitz gradients for global optimization in a one-point-based partitioning scheme. J. Comput. Appl. Math. 236(16), 4042–4054 (2012)
18. Butz, A.R.: Space-filling curves and mathematical programming. Inf. Control 12(4), 313–330 (1968)
19. Lera, D., Sergeyev, Y.D.: Deterministic global optimization using space-filling curves and multiple estimates of Lipschitz and Hölder constants. Commun. Nonlinear Sci. Numer. Simul. 23(1–3), 328–342 (2015)
20. Sergeyev, Y.D., Strongin, R.G., Lera, D.: Introduction to Global Optimization Exploiting Space-Filling Curves. Springer, Berlin (2013). doi: 10.1007/978-1-4614-8042-6
21. Schoen, F.: Two-Phase Methods for Global Optimization, Handbook of Global Optimization, Vol. 62 of the Series Nonconvex Optimization and its Applications, pp. 151–177 (2002)
22. Sergeyev, Y., Kvasov, D.E.: Global search based on efficient diagonal partitions and a set of Lipschitz constants. SIAM J. Optim. 16(3), 910–937 (2006). doi: 10.1137/040611320
23. Boothby, M.W.: An Introduction to Differentiable Manifolds and Riemannian Geometry. Springer, Berlin (2003)
24. Lee, M.J.: Introduction to Topological Manifolds. Springer, Berlin (2010)
25. Foster, J.: A Short Course in General Relativity. Springer, Berlin (2006)
26. Hawking, S., Ellis, G.F.R.: The Large Scale Structure of Space-Time. Cambridge University Press, Cambridge (1973)
27. Wales, D.J., Doye, J.P.K.: Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones Clusters containing up to 110 atoms. Phys. Chem. A 101, 5111 (1997)
28. Gaviano, M., Kvasov, D.E., Lera, D., Sergeyev, Y.D.: Algorithm 829: software for generation of classes of test functions with known local and global minima for global optimization. ACM Trans. Math. Softw. 29(4), 469–480 (2003)
29. Addis, B., Locatelli, M.: A new class of test functions for global optimization. J. Glob. Optim. 38(3), 479–501 (2007)
30. Zilinskas, A.: A class of test functions for global optimization. J. Glob. Optim. 5(2), 195–199 (1994)
31. Schoen, F.: A wide class of test functions for global optimization. J. Glob. Optim. 3, 133–138 (1993)
32. Hedar, A.R.: Test Problems for Unconstrained Optimization, http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm
33. Liang, J.J., Qu, B.Y., Suganthan, P.N., Hernandez-Daz, A.G.: Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session and Competition on Real-parameter Optimization, Technical Report 201212. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China (2013)
34. Sergeyev, Y., Kvasov, D.E., Mukhametzhanov, M.S.: Operational zones for comparing metaheuristic and deterministic one-dimensional global optimization algorithms. Math. Comput. Simul. (2016). doi: 10.1016/j.matcom.2016.05.006
35. Jamil, M., Yang, X.S.: A literature survey of benchmark functions for global optimisation problems. Int. J. Math. Model. Numer. Optim. 4(2), 150–194 (2013)