Acta Numerica (2019), pp. 287–404
doi:10.1017/S0962492919000060

Derivative-free optimization methods

Jeffrey Larson, Matt Menickelly and Stefan M. Wild
Mathematics and Computer Science Division,
Argonne National Laboratory, Lemont, IL 60439, USA
E-mail: jmlarson@anl.gov, mmenickelly@anl.gov, wild@anl.gov

Dedicated to the memory of Andrew R. Conn for his inspiring enthusiasm and his many contributions to the renaissance of derivative-free optimization methods.

In many optimization problems arising from scientific, engineering and artificial intelligence applications, objective and constraint functions are available only as the output of a black-box or simulation oracle that does not provide derivative information. Such settings necessitate the use of methods for derivative-free, or zeroth-order, optimization. We provide a review and perspectives on developments in these methods, with an emphasis on highlighting recent developments and on unifying treatment of such problems in the non-linear optimization and machine learning literature. We categorize methods based on assumed properties of the black-box functions, as well as features of the methods. We first overview the primary setting of deterministic methods applied to unconstrained, non-convex optimization problems where the objective function is defined by a deterministic black-box oracle. We then discuss developments in randomized methods, methods that assume some additional structure about the objective (including convexity, separability and general non-smooth compositions), methods for problems where the output of the black-box oracle is stochastic, and methods for handling different types of constraints.

CONTENTS

1. Introduction

The growth in computing for scientific, engineering and social applications has long been a driver of advances in methods for numerical optimization. The development of derivative-free optimization methods – those methods that do not require the availability of derivatives – has especially been driven by the need to optimize increasingly complex and diverse problems. One of the earliest calculations on MANIAC,[1] an early computer based on the von Neumann architecture, was the approximate solution of a six-dimensional non-linear least-squares problem using a derivative-free coordinate search (Fermi and Metropolis 1952). Today, derivative-free methods are used routinely, for example by Google (Golovin *et al.* 2017), for the automation and tuning needed in the artificial intelligence era.

In this paper we survey methods for derivative-free optimization and key results for their analysis. Since the field – also referred to as black-box optimization, gradient-free optimization, optimization without derivatives, simulation-based optimization and zeroth-order optimization – is now far too expansive for a single survey, we focus on methods for local optimization of continuous-valued, single-objective problems. Although Section 8 illustrates further connections, here we mark the following notable omissions.

- We focus on methods that seek a local minimizer. Despite users understandably desiring the best possible solution, the problem of global optimization raises innumerably more mathematical and computational challenges than do the methods presented here. We instead point to the survey by Neumaier (2004), which importantly addresses general

[1] Mathematical Analyzer, Integrator, And Computer. Other lessons learned from this application are discussed by Anderson (1986).

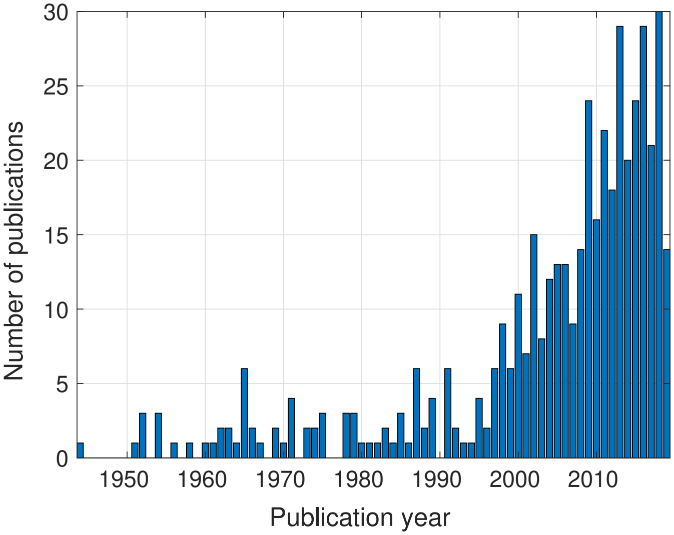

Figure 1.1. Histogram of the references cited in the bibliography.

constraints, and to the textbook by Forrester, Sobester and Keane (2008), which lays a foundation for global surrogate modelling.

- Multi-objective optimization and optimization in the presence of discrete variables are similarly popular tasks among users. Such problems possess fundamental challenges as well as differences from the methods presented here.
- In focusing on methods, we cannot do justice to the application problems that have driven the development of derivative-free methods and benefited from implementations of these methods. The recent textbook by Audet and Hare (2017) contains a number of examples and references to applications; Rios and Sahinidis (2013) and Auger *et al.* (2009) both reference a diverse set of implementations. At the persistent page

  https://archive.org/services/purl/dfomethods

  we intend to link all works that cite the entries in our bibliography and those that cite this survey; we hope this will provide a coarse, but dynamic, catalogue for the reader interested in potential uses of these methods.

Given these limitations, we particularly note the intersection with the foundational books by Kelley (1999*b*) and Conn, Scheinberg and Vicente (2009*b*). Our intent is to highlight recent developments in, and the evolution of, derivative-free optimization methods. Figure 1.1 summarizes our bias; over half of the references in this survey are from the past ten years.

Many of the fundamental inspirations for the methods discussed in this survey are detailed to a lesser extent. We note in particular the activity in the United Kingdom in the 1960s (see *e.g.* the works by Rosenbrock 1960, Powell 1964, Nelder and Mead 1965, Fletcher 1965 and Box 1966, and the later exposition and expansion by Brent 1973) and the Soviet Union (as evidenced by Rastrigin 1963, Matyas 1965, Karmanov 1974, Polyak 1987 and others). In addition to those mentioned later, we single out the work of Powell (1975), Wright (1995), Davis (2005) and Leyffer (2015) for insight into some of these early pioneers.

With our focus clear, we turn our attention to the deterministic optimization problem

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{x} \in \boldsymbol{\Omega} \subseteq \mathbb{R}^n \end{aligned} \tag{DET}$$

and the stochastic optimization problem

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[\tilde{f}(\boldsymbol{x};\boldsymbol{\xi})] \\ \text{subject to} \quad & \boldsymbol{x} \in \boldsymbol{\Omega}. \end{aligned} \tag{STOCH}$$

Although important exceptions are noted throughout this survey, the majority of the methods discussed assume that the objective function f in (DET) and (STOCH) is differentiable. This assumption may cause readers to pause (and some readers may never resume). The methods considered here do not necessarily address non-smooth optimization; instead they address problems where a (sub)gradient of the objective f or a constraint function defining $\boldsymbol{\Omega}$ is not available to the optimization method. Note that similar naming confusion has existed in non-smooth optimization, as evidenced by the introduction of Lemarechal and Mifflin (1978):

> This workshop was held under the name Nondifferentiable Optimization, but it has been recognized that this is misleading, because it suggests 'optimization without derivatives'.

1.1. Alternatives to derivative-free optimization methods

Derivative-free optimization methods are sometimes employed for convenience rather than by necessity. Since the decision to use a derivative-free method typically limits the performance – in terms of accuracy, expense or problem size – relative to what one might expect from gradient-based optimization methods, we first mention alternatives to using derivative-free methods.

The design of derivative-free optimization methods is informed by the alternatives of algorithmic and numerical differentiation. For the former, the purpose seems clear: since the methods use only function values, they apply even in cases when one cannot produce a computer code for the function's derivative. Similarly, derivative-free optimization methods should be designed in order to outperform (typically measured in terms of the number of function evaluations) gradient-based optimization methods that employ numerical differentiation.

1.1.1. Algorithmic differentiation

Algorithmic differentiation[2] (AD) is a means of generating derivatives of mathematical functions that are expressed in computer code (Griewank 2003, Griewank and Walther 2008). The forward mode of AD may be viewed as performing differentiation of elementary mathematical operations in each line of source code by means of the chain rule, while the reverse mode may be seen as traversing the resulting computational graph in reverse order.

Algorithmic differentiation has the benefit of automatically exploiting function structure, such as partial separability or other sparsity, and the corresponding ability of producing a derivative code whose computational cost is comparable to the cost of evaluating the function code itself.

AD has seen significant adoption and advances in the past decade (Forth *et al.* 2012). Tools for algorithmic differentiation cover a growing set of compiled and interpreted languages, with an evolving list summarized on the community portal at

http://www.autodiff.org.

Progress has also been made on algorithmic differentiation of piecewise smooth functions, such as those with breakpoints resulting from absolute values or conditionals in a code; see, for example, Griewank, Walther, Fiege and Bosse (2016). The machine learning renaissance has also fuelled demand and interest in AD, driven in large part by the success of algorithmic differentiation in backpropagation (Baydin, Pearlmutter, Radul and Siskind 2018).

1.1.2. Numerical differentiation

Another alternative to derivative-free methods is to estimate the derivative of f by numerical differentiation and then to use the estimates in a derivative-based method. This approach has the benefit that only zeroth-order information (*i.e.* the function value) is needed; however, depending on the derivative-based method used, the quality of the derivative estimate may be a limiting factor. Here we remark that for the finite-precision (or even fixed-precision) functions encountered in scientific applications, finite-difference estimates of derivatives may be sufficient for many purposes; see Section 2.3.1.

[2] *Algorithmic differentiation* is sometimes referred to as *automatic differentiation*, but we follow the preferred convention of Griewank (2003).

When numerical derivative estimates are used, the optimization method must tolerate inexactness in the derivatives. Such methods have been classically studied for both non-linear equations and unconstrained optimization; see, for example, the works of Powell (1965), Brown and Dennis, Jr (1971) and Mifflin (1975) and the references therein. Numerical derivatives continue to be employed by recent methods (see *e.g.* the works of Cartis, Gould and Toint 2012 and Berahas, Byrd and Nocedal 2019). Use in practice is typically determined by whether the limit on the derivative accuracy and the expense in terms of function evaluations are acceptable.

1.2. Organization of the paper

This paper is organized principally by problem class: unconstrained domain (Sections 2 and 3), convex objective (Section 4), structured objective (Section 5), stochastic optimization (Section 6) and constrained domain (Section 7).

Section 2 presents deterministic methods for solving (DET) when $\mathbf{\Omega} = \mathbb{R}^n$. The section is split between direct-search methods and model-based methods, although the lines between these are increasingly blurred; see, for example, Conn and Le Digabel (2013), Custódio, Rocha and Vicente (2009), Gramacy and Le Digabel (2015) and Gratton, Royer and Vicente (2016). Direct-search methods are summarized in far greater detail by Kolda, Lewis and Torczon (2003) and Kelley (1999*b*), and in the more recent survey by Audet (2014). Model-based methods that employ trust regions are given full treatment by Conn *et al.* (2009*b*), and those that employ stencils are detailed by Kelley (2011).

In Section 3 we review randomized methods for solving (DET) when $\mathbf{\Omega} = \mathbb{R}^n$. These methods are often variants of the deterministic methods in Section 2 but require additional notation to capture the resulting stochasticity; the analysis of these methods can also deviate significantly from their deterministic counterparts.

In Section 4 we discuss derivative-free methods intended primarily for convex optimization. We make this delineation because such methods have distinct lines of analysis and can often solve considerably higher-dimensional problems than can general methods for non-convex derivative-free optimization.

In Section 5 we survey methods that address particular structure in the objective f in (DET). Examples of such structure include non-linear least-squares objectives, composite non-smooth objectives and partially separable objectives.

In Section 6 we address derivative-free stochastic optimization, that is, when methods have access only to a stochastic realization of a function in pursuit of solving (STOCH). This topic is increasingly intertwined with simulation optimization and Monte Carlo-based optimization; for these areas we refer to the surveys by Homem-de-Mello and Bayraksan (2014), Fu, Glover and April (2005), Amaran, Sahinidis, Sharda and Bury (2015) and Kim, Pasupathy and Henderson (2015).

Section 7 presents methods for deterministic optimization problems with constraints (*i.e.* $\mathbf{\Omega} \subset \mathbb{R}^n$). Although many of these methods rely on the foundations laid in Sections 2 and 3, we highlight particular difficulties associated with constrained derivative-free optimization.

In Section 8 we briefly highlight related problem areas (including global and multi-objective derivative-free optimization), methods and other implementation considerations.

2. Deterministic methods for deterministic objectives

We now address deterministic methods for solving (DET). We discuss direct-search methods in Section 2.1, model-based methods in Section 2.2 and other methods in Section 2.3. At a coarse level, direct-search methods use comparisons of function values to directly determine candidate points, whereas model-based methods use a surrogate of f to determine candidate points. Naturally, some hybrid methods incorporate ideas from both model-based and direct-search methods and may not be so easily categorized. An early survey of direct-search and model-based methods is given in Powell (1998*a*).

2.1. Direct-search methods

Although Hooke and Jeeves (1961) are credited with originating the term ‘direct search’, there is no agreed-upon definition of what constitutes a direct-search method. We follow the convention of Wright (1995), wherein a direct-search method is a method that uses only function values and ‘does not “in its heart” develop an approximate gradient’.

We first discuss simplex methods, including the Nelder–Mead method – perhaps the most widely used direct-search method. We follow this discussion with a presentation of directional direct-search methods; hybrid direct-search methods are discussed in Section 2.3. (The global direct-search method DIRECT is discussed in Section 8.3.)

2.1.1. Simplex methods

Simplex methods (not to be confused with Dantzig’s simplex method for linear programming) move and manipulate a collection of $n+1$ affinely independent points (*i.e.* the vertices of a simplex in $\mathbb{R}^n$) when solving (DET). The method of Spendley, Hext and Himsworth (1962) involves either taking the point in the simplex with the largest function value and reflecting it through the hyperplane defined by the remaining n points or moving the

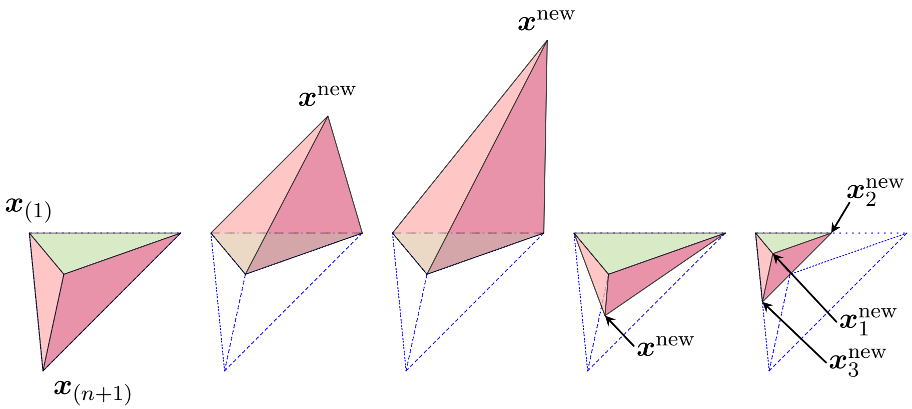
$\boldsymbol{x}_{(1)}$
$\boldsymbol{x}_{(n+1)}$
$\boldsymbol{x}^{\text{new}}$
$\boldsymbol{x}^{\text{new}}$
$\boldsymbol{x}^{\text{new}}$
$\boldsymbol{x}_2^{\text{new}}$
$\boldsymbol{x}_1^{\text{new}}$
$\boldsymbol{x}_3^{\text{new}}$

Figure 2.1. Primary Nelder–Mead simplex operations: original simplex, reflection, expansion, inner contraction, and shrink.

n worst points toward the best vertex of the simplex. In this manner, the geometry of all simplices remains the same as that of the starting simplex. (That is, all simplices are similar in the geometric sense.)

Nelder and Mead (1965) extend the possible simplex operations, as shown in Figure 2.1, by allowing the 'expansion' and 'contraction' operations in addition to the 'reflection' and 'shrink' operations of Spendley *et al.* (1962). These operations enable the Nelder–Mead simplex method to distort the simplex in order to account for possible curvature present in the objective function.

Nelder and Mead (1965) propose stopping further function evaluations when the standard error of the function values at the simplex vertices is small. Others, Woods (1985) for example, propose stopping when the size of the simplex's longest side incident to the best simplex vertex is small.

Nelder–Mead is an incredibly popular method, in no small part due to its inclusion in *Numerical Recipes* (Press, Teukolsky, Vetterling and Flannery 2007), which has been cited over 125 000 times and no doubt used many times more. The method (as implemented by Lagarias, Poonen and Wright 2012) is also the algorithm underlying fminsearch in MATLAB. Benchmarking studies highlight Nelder–Mead performance in practice (Moré and Wild 2009, Rios and Sahinidis 2013).

The method's popularity from its inception was not diminished by the lack of theoretical results proving its ability to identify stationary points. Woods (1985) presents a non-convex, two-dimensional function where Nelder–Mead converges to a non-stationary point (where the function's Hessian is singular). Furthermore, McKinnon (1998) presents a class of thrice-continuously differentiable, strictly convex functions on $\mathbb{R}^2$ where the Nelder–Mead simplex fails to converge to the lone stationary point. The only operation that Nelder–Mead performs on this relatively routine function is repeated 'inner contraction' of the initial simplex.

Researchers have continued to develop convergence results for modified or limited versions of Nelder–Mead. Kelley (1999*a*) addresses Nelder–Mead's theoretical deficiencies by restarting the method when the objective decrease on consecutive iterations is not larger than a multiple of the simplex gradient norm. Such restarts do not ensure that Nelder–Mead will converge: Kelley (1999*a*) shows an example of such behaviour. Price, Coope and Byatt (2002) embed Nelder–Mead in a different (convergent) algorithm using positive spanning sets. Nazareth and Tseng (2002) propose a clever, though perhaps superfluous, variant that connects Nelder–Mead to golden-section search.

Lagarias, Reeds, Wright and Wright (1998) show that Nelder–Mead (with appropriately chosen reflection and expansion coefficients) converges to the global minimizer of strictly convex functions when $n = 1$. Gao and Han (2012) show that the contraction and expansion steps of Nelder–Mead satisfy a descent condition on uniformly convex functions. Lagarias, Poonen and Wright (2012) show that a restricted version of the Nelder–Mead method – one that does not allow an expansion step – can converge to minimizers of any twice-continuously differentiable function with a positive-definite Hessian and bounded level sets. (Note that the class of functions from McKinnon (1998) have singular Hessians at only one point – their minimizers – and not at the point to which the simplex vertices are converging.)

The simplex method of Rykov (1980) includes ideas from model-based methods. Rykov varies the number of reflected vertices from iteration to iteration, following one of three rules that depend on the function value at the simplex centroid $\boldsymbol{x}_c$. Rykov considers both evaluating f at the centroid and approximating f at the centroid using the values of f at the vertex. The non-reflected vertices are also moved in parallel with the reflected subset of vertices. In general, the number of reflected vertices is chosen so that $\boldsymbol{x}_c$ moves in a direction closest to $-\nabla f(\boldsymbol{x}_c)$. This, along with a test of sufficient decrease in f, ensures convergence of the modified simplex method to a minimizer of convex, continuously differentiable functions with bounded level sets and Lipschitz-bounded gradients. (The sufficient-decrease condition is also shown to be efficient for the classical Nelder–Mead algorithm.)

Tseng (1999) proposes a modified simplex method that keeps the b_k best simplex vertices on a given iteration k and uses them to reflect the remaining vertices. Their method prescribes that 'the rays emanating from the reflected vertices toward the b_k best vertices should contain, in their convex hull, the rays emanating from a weighted centroid of the b_k best vertices toward the to-be-reflected vertices'. Their method also includes a *fortified descent* condition that is stronger than common sufficient-decrease conditions. If f is continuously differentiable and bounded below and b_k is fixed for all

Algorithm 1: $\boldsymbol{x}^+ = \texttt{test\_descent}(f, \boldsymbol{x}, \boldsymbol{P})$

```
1 Initialize $\boldsymbol{x}^+ \leftarrow \boldsymbol{x}$
2 for $\boldsymbol{p}_i \in \boldsymbol{P}$ do
3     Evaluate $f(\boldsymbol{p}_i)$
4     if $f(\boldsymbol{p}_i) - f(\boldsymbol{x})$ acceptable then
5         $\boldsymbol{x}^+ \leftarrow \boldsymbol{p}_i$
6         optional break
```

iterations, Tseng (1999) prove that every cluster point of the sequence of candidate points generated by their method is a stationary point.

Bűrmen, Puhan and Tuma (2006) propose a convergent version of a simplex method that does not require a sufficient descent condition to be satisfied. Instead, they ensure that evaluated points lie on a grid of points, and they show that this grid will be refined as the method proceeds.

2.1.2. Directional direct-search methods

Broadly speaking, each iteration of a directional direct-search (DDS) method generates a finite set of points near the current point $\boldsymbol{x}_k$; these *poll points* are generated by taking $\boldsymbol{x}_k$ and adding terms of the form $\alpha_k \boldsymbol{d}$, where α_k is a positive step size and $\boldsymbol{d}$ is an element from a finite set of directions $\boldsymbol{D}_k$. Kolda *et al.* (2003) propose the term *generating set search methods* to encapsulate this class of methods.[3] The objective function f is then evaluated at all or some of the poll points, and $\boldsymbol{x}_{k+1}$ is selected to be some poll point that produces a (sufficient) decrease in the objective and the step size is possibly increased. If no poll point provides a sufficient decrease, $\boldsymbol{x}_{k+1}$ is set to $\boldsymbol{x}_k$ and the step size is decreased. In either case, the set of directions $\boldsymbol{D}_k$ can (but need not) be modified to obtain $\boldsymbol{D}_{k+1}$.

A general DDS method is provided in Algorithm 2, which includes a *search step* where f is evaluated at any finite set of points $\boldsymbol{Y}_k$, including $\boldsymbol{Y}_k = \emptyset$. The search step allows one to (potentially) improve the performance of Algorithm 2. For example, points could be randomly sampled during the search step from the domain in the hope of finding a better local minimum, or a person running the algorithm may have problem-specific knowledge that can generate candidate points given the observed history of evaluated points and their function values. While the search step allows for this insertion of such heuristics, rigorous convergence results are driven by the more disciplined poll step. When testing for objective decrease in Algorithm 1, one can stop evaluating points in $\boldsymbol{P}$ (line 6) as soon as the first point is

[3] The term generating set arises from a need to generate a cone from the nearly active constraint normals when $\boldsymbol{\Omega}$ is defined by linear constraints.

Algorithm 2: Directional direct-search method

```
 1  Set parameters $0 < \gamma_{\mathrm{dec}} < 1 \le \gamma_{\mathrm{inc}}$
 2  Choose initial point $\boldsymbol{x}_0$ and step size $\alpha_0 > 0$
 3  for $k = 0, 1, 2, \ldots$ do
 4      Choose and order a finite set $\boldsymbol{Y}_k \subset \mathbb{R}^n$            // (search step)
 5      $\boldsymbol{x}_k^+ \leftarrow$ test_descent($f, \boldsymbol{x}_k, \boldsymbol{Y}_k$)
 6      if $\boldsymbol{x}_k^+ = \boldsymbol{x}_k$ then
 7          Choose and order poll directions $\boldsymbol{D}_k \subset \mathbb{R}^n$      // (poll step)
 8          $\boldsymbol{x}_k^+ \leftarrow$ test_descent($f, \boldsymbol{x}_k, \{\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_i : \boldsymbol{d}_i \in \boldsymbol{D}_k\}$)
 9      if $\boldsymbol{x}_k^+ = \boldsymbol{x}_k$ then
10          $\alpha_{k+1} \leftarrow \gamma_{\mathrm{inc}} \alpha_k$
11      else
12          $\alpha_{k+1} \leftarrow \gamma_{\mathrm{dec}} \alpha_k$
13      $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k^+$
```

identified where there is (sufficient) decrease in f. In this case, the polling (or search) step is considered *opportunistic*.

DDS methods are largely distinguished by how they generate the set of poll directions $\boldsymbol{D}_k$ at line 7 of Algorithm 2. Perhaps the first approach is *coordinate search*, in which the poll directions are defined as $\boldsymbol{D}_k = \{\pm \boldsymbol{e}_i : i = 1, 2, \ldots, n\}$, where $\boldsymbol{e}_i$ denotes the ith elementary basis vector (*i.e.* column i of the identity matrix in n dimensions). The first known description of coordinate search appears in the work of Fermi and Metropolis (1952) where the smallest positive integer l is sought such that $f(\boldsymbol{x}_k + l\alpha \boldsymbol{e}_1/2) > f(\boldsymbol{x}_k + (l-1)\alpha \boldsymbol{e}_1/2)$. If an increase in f is observed at $\boldsymbol{e}_1/2$ then $-\boldsymbol{e}_1/2$ is considered. After such an integer l is identified for the first coordinate direction, $\boldsymbol{x}_k$ is updated to $\boldsymbol{x}_k \pm l\boldsymbol{e}_1/2$ and the second coordinate direction is considered. If $\boldsymbol{x}_k$ is unchanged after cycling through all coordinate directions, then the method is repeated but with $\pm \boldsymbol{e}_i/2$ replaced with $\pm \boldsymbol{e}_i/16$, terminating when no improvement is observed for this smaller α. In terms of Algorithm 2 the search set $\boldsymbol{Y}_k = \emptyset$ at line 4, and the descent test at line 4 of Algorithm 1 merely tests for simple decrease, that is, $f(\boldsymbol{p}_i) - f(\boldsymbol{x}) < 0$. Other versions of acceptability in line 4 of Algorithm 1 are employed by methods discussed later.

Proofs that DDS methods converge first appeared in the works of Céa (1971) and Yu (1979), although both require the sequence of step-size parameters to be non-increasing. Lewis, Torczon and Trosset (2000) attribute the first global convergence proof for coordinate search to Polak (1971, p. 43). In turn, Polak cites the 'method of local variation' of Banichuk, Petrov and Chernous'ko (1966); although Banichuk *et al.* (1966) do develop parts of a convergence proof, they state in Remark 1 that ‘the question of the strict formulation of the general sufficient conditions for convergence of the algorithm to a minimum remains open’.

Typical convergence results for DDS require that the set $\boldsymbol{D}_k$ is a *positive spanning set* (PSS) for the domain $\boldsymbol{\Omega}$; that is, any point $\boldsymbol{x} \in \boldsymbol{\Omega}$ can be written as

$$\boldsymbol{x} = \sum_{i=1}^{|\boldsymbol{D}_k|} \lambda_i \boldsymbol{d}_i,$$

where $\boldsymbol{d}_i \in \boldsymbol{D}_k$ and $\lambda_i \geq 0$ for all i. Some of the first discussions of properties of positive spanning sets were presented by Davis (1954) and McKinney (1962), but recent treatments have also appeared in Regis (2016). In addition to requiring positive spanning sets during the poll step, earlier DDS convergence results depended on f being continuously differentiable. When f is non-smooth, no descent direction is guaranteed for these early DDS methods, even when the step size is arbitrarily small. See, for example, the modification of the Dennis–Woods (Dennis, Jr and Woods 1987) function by Kolda *et al.* (2003, Figure 6.2) and a discussion of why coordinate-search methods (for example) will not move when started at a point of non-differentiability; moreover, when started at differentiable points, coordinate-search methods tend to converge to a point that is not (Clarke) stationary.

The pattern-search method of Torczon (1991) revived interest in direct-search methods. The method therein contains ideas from both DDS and simplex methods. Given a simplex defined by $\boldsymbol{x}_k, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ (where $\boldsymbol{x}_k$ is the simplex vertex with smallest function value), the polling directions are given by $\boldsymbol{D}_k = \{\boldsymbol{y}_i - \boldsymbol{x}_k : i = 1, \ldots, n\}$. If a decrease is observed at the best poll point in $\boldsymbol{x}_k + \boldsymbol{D}_k$, the simplex is set to either $\boldsymbol{x}_k \bigcup \boldsymbol{x}_k + \boldsymbol{D}_k$ or some expansion thereof. If no improvement is found during the poll step, the simplex is contracted. Torczon (1991) shows that if f is continuous on the level set of $\boldsymbol{x}_0$ and this level set is compact, then a subsequence of $\{\boldsymbol{x}_k\}$ converges to a stationary point of f, a point where f is non-differentiable, or a point where f is not continuously differentiable.

A generalization of pattern-search methods is the class of *generalized pattern-search* (GPS) *methods*. Early GPS methods did not allow for a search step; the search-poll paradigm was introduced by Audet (2004). GPS methods are characterized by fixing a positive spanning set $\boldsymbol{D}$ and selecting $\boldsymbol{D}_k \subseteq \boldsymbol{D}$ during the poll step at line 7 on each iteration of Algorithm 2. Torczon (1997) assumes that the test for decrease in line 4 in Algorithm 1 is simple decrease, that is, that $f(\boldsymbol{p}_i) < f(\boldsymbol{x})$. Early analysis of GPS methods using simple decrease required the step size α_k to remain rational (Audet and Dennis, Jr 2002, Torczon 1997). Audet (2004) shows that such an assumption is necessary by constructing small-dimensional examples where

GPS methods do not converge if α_k is irrational. Works below show that if a sufficient (instead of simple) decrease is ensured, α_k can take irrational values.

A refinement of the analysis of GPS methods was made by Dolan, Lewis and Torczon (2003), which shows that when ∇f is Lipschitz-continuous, the step-size parameter α_k scales linearly with $\|\nabla f(\boldsymbol{x}_k)\|$. Therefore α_k can be considered a reliable measure of first-order stationarity and justifies the traditional approach of stopping a GPS method when α_k is small. Second-order convergence analyses of GPS methods have also been considered. Abramson (2005) shows that, when applied to a twice-continuously differentiable f, a GPS method that infinitely often has $\boldsymbol{D}_k$ include a fixed orthonormal basis and its negative will have a limit point satisfying a 'pseudo-second-order' stationarity condition. Building off the use of curvature information in Frimannslund and Steihaug (2007), Abramson, Frimannslund and Steihaug (2013) show that a modification of the GPS framework that constructs approximate Hessians of f will converge to points that are second-order stationary provided that certain conditions on the Hessian approximation hold (and a fixed orthonormal basis and its negative are in $\boldsymbol{D}_k$ infinitely often).

In general, first-order convergence results (there exists a limit point $\boldsymbol{x}_*$ of $\{\boldsymbol{x}_k\}$ generated by a GPS method such that $\nabla f(\boldsymbol{x}_*) = \boldsymbol{0}$) for GPS methods can be demonstrated when f is continuously differentiable. For general Lipschitz-continuous (but non-smooth) functions f, however, one can only demonstrate that on a particular subsequence $\mathcal{K}$, satisfying $\{\boldsymbol{x}_k\}_{k\in\mathcal{K}} \to \boldsymbol{x}_*$, for each $\boldsymbol{d}$ that appears infinitely many times in $\{\boldsymbol{D}_k\}_{k\in\mathcal{K}}$, it holds that $f'(\boldsymbol{x}_*;\boldsymbol{d}) \geq 0$; that is, the directional derivative at $\boldsymbol{x}_*$ in the direction $\boldsymbol{d}$ is non-negative.

The flexibility of GPS methods inspired various extensions. Abramson, Audet and Dennis, Jr (2004) consider adapting GPS to utilize derivative information when it is available in order to reduce the number of points evaluated during the poll step. Abramson, Audet, Dennis, Jr and Le Digabel (2009*b*) and Frimannslund and Steihaug (2011) re-use previous function evaluations in order to determine the next set of directions. Custódio and Vicente (2007) consider re-using previous function evaluations to compute simplex gradients; they show that the information obtained from simplex gradients can be used to reorder the poll points $\boldsymbol{P}$ in Algorithm 1. A similar use of simplex gradients in the non-smooth setting is considered by Custódio, Dennis, Jr and Vicente (2008). Hough, Kolda and Torczon (2001) discuss modifications to Algorithm 2 that allow for increased efficiency when concurrent, asynchronous evaluations of f are possible; an implementation of the method of Hough *et al.* (2001) is presented by Gray and Kolda (2006).

The early analysis of Torczon (1991, Section 7) of pattern-search methods when f is non-smooth carries over to GPS methods as well; such methods may converge to a non-stationary point. This motivated a further generalization of GPS methods, *mesh adaptive direct search* (MADS) methods (Audet and Dennis, Jr 2006, Abramson and Audet 2006). Inspired by Coope and Price (2000), MADS methods augment GPS methods by incorporating a mesh parametrized by a *mesh parameter* $\beta_k^m > 0$. In the kth iteration, given the fixed PSS $\boldsymbol{D}$ and the mesh parameter β_k^m, the MADS mesh around the current point $\boldsymbol{x}_k$ is

$$\mathcal{M}_k = \bigcup_{\boldsymbol{x}\in\boldsymbol{S}_k} \left\{ \boldsymbol{x} + \beta_k^m \sum_{j=1}^{|\boldsymbol{D}|} \lambda_j \boldsymbol{d}_j : \boldsymbol{d}_j \in \boldsymbol{D}, \lambda_j \in \mathbb{N} \bigcup \{0\} \right\},$$

where $\boldsymbol{S}_k$ is the set of points at which f has been evaluated prior to the kth iteration of the method.

MADS methods additionally define a frame

$$\mathcal{F}_k = \{\boldsymbol{x}_k + \beta_k^m \boldsymbol{d}^f : \boldsymbol{d}^f \in \boldsymbol{D}_k^f\},$$

where $\boldsymbol{D}_k^f$ is a finite set of directions, each of which is expressible as

$$\boldsymbol{d}^f = \sum_{j=1}^{|\boldsymbol{D}|} \lambda_j \boldsymbol{d}_j,$$

with each $\lambda_j \in \mathbb{N} \bigcup \{0\}$ and $\boldsymbol{d}_j \in \boldsymbol{D}$. Additionally, MADS methods define a *frame parameter* β_k^f and require that each $\boldsymbol{d}^f \in \boldsymbol{D}_k^f$ satisfies $\beta_k^m \|\boldsymbol{d}^f\| \leq \beta_k^f \max\{\|\boldsymbol{d}\| : \boldsymbol{d} \in \boldsymbol{D}\}$. Observe that in each iteration, $\mathcal{F}_k \subsetneq \mathcal{M}_k$. Note that the mesh is never explicitly constructed nor stored over the domain. Rather, points are evaluated only at what *would be* nodes of some implicitly defined mesh via the frame.

In the poll step of Algorithm 2, the set of poll directions $\boldsymbol{D}_k$ is chosen as $\{\boldsymbol{y} - \boldsymbol{x}_k : \boldsymbol{y} \in \mathcal{F}_k\}$. The role of the step-size parameter α_k in Algorithm 2 is completely replaced by the behaviour of β_k^f, β_k^m. If there is no improvement at a candidate solution during the poll step, β_k^m is decreased, resulting in a finer mesh; likewise β_k^f is decreased, resulting in a finer local mesh around $\boldsymbol{x}_k$. MADS intentionally allows the parameters β_k^m and β_k^f to be decreased at different rates; roughly speaking, by driving β_k^m to zero faster than β_k^f is driven to zero, and by choosing the sequence $\{\boldsymbol{D}_k^f\}$ to satisfy certain conditions, the directions in $\mathcal{F}_k$ become asymptotically dense around limit points of $\boldsymbol{x}_k$. That is, it is possible to decrease β_k^m, β_k^f at rates such that poll directions will be arbitrarily close to any direction. This ensures that the Clarke directional derivative is non-negative in all directions around any limit point of the sequence of $\boldsymbol{x}_k$ generated by MADS; that is,

$$f_C'(\boldsymbol{x}_*; \boldsymbol{d}) \geq 0 \quad \text{for all directions } \boldsymbol{d}, \tag{2.1}$$

with an analogous result also holding for constrained problems, with (2.1) reduced to all *feasible* directions $\boldsymbol{d}$. (DDS methods for constrained optimization will be discussed in Section 7.) This powerful result highlights the ability of directional direct-search methods to address non-differentiable functions f.

MADS does not prescribe any one approach for adjusting β_k^m, β_k^f so that the poll directions are dense, but Audet and Dennis, Jr (2006) demonstrate an approach where randomized directions are completed to be a PSS and β_k^f either is $n\sqrt{\beta_k^m}$ or $\sqrt{\beta_k^m}$ results in a asymptotically dense poll directions for any convergent subsequence of $\{\boldsymbol{x}_k\}$. MADS does not require a sufficient-decrease condition.

Recent advances to MADS-based algorithms have focused on reducing the number of function evaluations required in practice by adaptively reducing the number of poll points queried; see, for example, Audet, Ianni, Le Digabel and Tribes (2014) and Alarie *et al.* (2018). Smoothing-based extensions to noisy deterministic problems include Audet, Ihaddadene, Le Digabel and Tribes (2018*b*). Vicente and Custódio (2012) show that MADS methods converge to local minima even for a limited class of *discontinuous* functions that satisfy some assumptions concerning the behaviour of the disconnected regions of the epigraph at limit points.

Worst-case complexity analysis. Throughout this survey, when discussing classes of methods, we will refer to their worst-case complexity (WCC). Generally speaking, WCC refers to an upper bound on the number of function evaluations N_ϵ required to attain an ϵ-accurate solution to a problem drawn from a problem class. Correspondingly, the definition of ϵ-accurate varies between different problem classes. For instance, and of particular immediate importance, if an objective function is assumed Lipschitz-continuously differentiable (which we denote by $f \in \mathcal{LC}^1$), then an appropriate notion of first-order ϵ-accuracy is

$$\|\nabla f(\boldsymbol{x}_k)\| \le \epsilon. \tag{2.2}$$

That is, the WCC of a method applied to the class $\mathcal{LC}^1$ is characterized by N_ϵ, an upper bound on the number of function evaluations the method requires before (2.2) is satisfied for *any* $f \in \mathcal{LC}^1$. Similarly, we can define a notion of second-order ϵ-accuracy as

$$\max\{\|\nabla f(\boldsymbol{x}_k)\|, -\lambda_k\} \le \epsilon, \tag{2.3}$$

where λ_k denotes the minimum eigenvalue of $\nabla^2 f(\boldsymbol{x}_k)$.

Note that WCCs can only be derived for methods for which convergence results have been established. Indeed, in the problem class $\mathcal{LC}^1$, first-order convergence results canonically have the form

$$\lim_{k\to\infty} \|\nabla f(\boldsymbol{x}_k)\| = 0. \tag{2.4}$$

The convergence in (2.4) automatically implies the weaker lim-inf-type result

$$\liminf_{k\to\infty} \|\nabla f(\boldsymbol{x}_k)\| = 0, \tag{2.5}$$

from which it is clear that for any $\epsilon > 0$, there must exist finite N_ϵ so that (2.2) holds. In fact, in many works, demonstrating a result of the form (2.5) is a stepping stone to proving a result of the form (2.4). Likewise, demonstrating a second-order WCC of the form (2.3) depends on showing

$$\lim_{k\to\infty} \max\{\|\nabla f(\boldsymbol{x}_k)\|, -\lambda_k\} = 0, \tag{2.6}$$

which guarantees the weaker lim-inf-type result

$$\liminf_{k\to\infty} \max\{\|\nabla f(\boldsymbol{x}_k)\|, -\lambda_k\} = 0. \tag{2.7}$$

Proofs of convergence for DDS methods applied to functions $f \in \mathcal{LC}^1$ often rely on a (sub)sequence of positive spanning sets $\{\boldsymbol{D}_k\}$ satisfying

$$\mathrm{cm}(\boldsymbol{D}_k) = \min_{\boldsymbol{v}\in\mathbb{R}^n\setminus\{\boldsymbol{0}\}} \max_{\boldsymbol{d}\in\boldsymbol{D}_k} \frac{\boldsymbol{d}^\top \boldsymbol{v}}{\|\boldsymbol{d}\|\|\boldsymbol{v}\|} \geq \kappa > 0, \tag{2.8}$$

where $\mathrm{cm}(\cdot)$ is the *cosine measure* of a set. Under Assumption (2.8), Vicente (2013) obtains a WCC of type (2.2) for a method in the Algorithm 2 framework. In that work, it is assumed that $\boldsymbol{Y}_k = \emptyset$ at every search step. Moreover, *sufficient* decrease is tested at line 4 of Algorithm 1; in particular, Vicente (2013) checks in this line whether $f(\boldsymbol{p}_i) < f(\boldsymbol{x}) - c\alpha_k^2$ for some $c > 0$, where α_k is the current step size in Algorithm 2. Under these assumptions, Vicente (2013) demonstrates a WCC in $O(\epsilon^{-2})$. Throughout this survey, we will refer to Table A.1 for more details concerning specific WCCs. In general, though, we will often summarize WCCs in terms of their ϵ-dependence, as this provides an asymptotic characterization of a method's complexity in terms of the accuracy to which one wishes to solve a problem.

When $f \in \mathcal{LC}^2$, work by Gratton *et al.* (2016) essentially augments the DDS method analysed by Vicente (2013), but forms an approximate Hessian via central differences from function evaluations obtained (for free) by using a particular choice of $\boldsymbol{D}_k$. Gratton *et al.* (2016) then demonstrate that this augmentation of Algorithm 2 has a subsequence that converges to a second-order stationary point. That is, they prove a convergence result of the form (2.7) and demonstrate a WCC result of type (2.3) in $O(\epsilon^{-3})$ (see Table A.1).

We are unaware of WCC results for MADS methods; this situation may be unsurprising since MADS methods are motivated by non-smooth problems, which depend on the generation of a countably infinite number of poll directions. However, WCC results are not necessarily impossible to obtain in *structured* non-smooth cases, which we discuss in Section 5. We will discuss a special case where *smoothing functions* of a non-smooth function are assumed to be available in Section 5.3.2.

2.2. Model-based methods

In the context of derivative-free optimization, model-based methods are methods whose updates are based primarily on the predictions of a model that serves as a surrogate of the objective function or of a related merit function. We begin with basic properties and construction of popular models; readers interested in algorithmic frameworks such as trust-region methods and implicit filtering can proceed to Section 2.2.4. Throughout this section, we assume that models are intended as a surrogate for the function f; in future sections, these models will be extended to capture functions arising, for example, as constraints or separable components. The methods in this section assume some smoothness in f and therefore operate with smooth models; in Section 5, we examine model-based methods that exploit knowledge of non-smoothness.

2.2.1. Quality of smooth model approximation

A natural first indicator of the quality of a model used for optimization is the degree to which the model locally approximates the function f and its derivatives. To say anything about the quality of such approximation, one must make an assumption about the smoothness of both the model and function. For the moment, we leave this assumption implicit, but it will be formalized in subsequent sections.

A function $m : \mathbb{R}^n \to \mathbb{R}$ is said to be a $\boldsymbol{\kappa}$-fully linear model of f on $\mathcal{B}(\boldsymbol{x};\Delta) = \{\boldsymbol{y} : \|\boldsymbol{x}-\boldsymbol{y}\| \le \Delta\}$ if

$$|f(\boldsymbol{x}+\boldsymbol{s}) - m(\boldsymbol{x}+\boldsymbol{s})| \le \kappa_{\text{ef}}\Delta^2, \quad \text{for all } \boldsymbol{s} \in \mathcal{B}(\boldsymbol{0};\Delta), \tag{2.9a}$$

$$\|\nabla f(\boldsymbol{x}+\boldsymbol{s}) - \nabla m(\boldsymbol{x}+\boldsymbol{s})\| \le \kappa_{\text{eg}}\Delta, \quad \text{for all } \boldsymbol{s} \in \mathcal{B}(\boldsymbol{0};\Delta), \tag{2.9b}$$

for $\boldsymbol{\kappa} = (\kappa_{\text{ef}}, \kappa_{\text{eg}})$. Similarly, for $\boldsymbol{\kappa} = (\kappa_{\text{ef}}, \kappa_{\text{eg}}, \kappa_{\text{eH}})$, m is said to be a $\boldsymbol{\kappa}$-fully quadratic model of f on $\mathcal{B}(\boldsymbol{x};\Delta)$ if

$$|f(\boldsymbol{x}+\boldsymbol{s}) - m(\boldsymbol{x}+\boldsymbol{s})| \le \kappa_{\text{ef}}\Delta^3, \quad \text{for all } \boldsymbol{s} \in \mathcal{B}(\boldsymbol{0};\Delta), \tag{2.10a}$$

$$\|\nabla f(\boldsymbol{x}+\boldsymbol{s}) - \nabla m(\boldsymbol{x}+\boldsymbol{s})\| \le \kappa_{\text{eg}}\Delta^2, \quad \text{for all } \boldsymbol{s} \in \mathcal{B}(\boldsymbol{0};\Delta), \tag{2.10b}$$

$$\|\nabla^2 f(\boldsymbol{x}+\boldsymbol{s}) - \nabla^2 m(\boldsymbol{x}+\boldsymbol{s})\| \le \kappa_{\text{eH}}\Delta, \quad \text{for all } \boldsymbol{s} \in \mathcal{B}(\boldsymbol{0};\Delta). \tag{2.10c}$$

Extensions to higher-degree approximations follow a similar form, but the computational expense associated with achieving higher-order guarantees is not a strategy pursued by derivative-free methods that we are aware of.

Models satisfying (2.9) or (2.10) are called Taylor-like models. To understand why, consider the second-order Taylor model

$$m(\boldsymbol{x}+\boldsymbol{s}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\nabla^2 f(\boldsymbol{x})\boldsymbol{s}. \tag{2.11}$$

This model is a $\boldsymbol{\kappa}$-fully quadratic model of f, with

$$(\kappa_{\mathrm{ef}}, \kappa_{\mathrm{eg}}, \kappa_{\mathrm{eH}}) = (L_{\mathrm{H}}/6, L_{\mathrm{H}}/2, L_{\mathrm{H}}),$$

on any $\mathcal{B}(\boldsymbol{x};\Delta)$, where f has a Lipschitz-continuous second derivative with Lipschitz constant L_{H}.

As illustrated in the next section, one also can guarantee that models that do not employ derivative information satisfy these approximation bounds in (2.9) or (2.10). This approximation quality is used by derivative-free algorithms to ensure that a sufficient reduction predicted by the model m yields an attainable reduction in the function f as Δ becomes smaller.

2.2.2. Polynomial models

Polynomial models are the most commonly used models for derivative-free local optimization. We let $\mathcal{P}^{d,n}$ denote the space of polynomials of n variables of degree d and $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^{\dim(\mathcal{P}^{d,n})}$ define a basis for this space. For example, quadratic models can be obtained by using the monomial basis

$$\boldsymbol{\phi}(\boldsymbol{x}) = [1,\, x_1,\, \ldots,\, x_n,\, x_1^2,\, \ldots x_n^2,\, x_1x_2,\, \ldots, x_{n-1}x_n]^{\mathrm{T}}, \tag{2.12}$$

for which $\dim(\mathcal{P}^{2,n}) = (n+1)(n+2)/2$; linear models can be obtained by using the first $\dim(\mathcal{P}^{1,n}) = n+1$ components of (2.12); quadratic models with diagonal Hessians, which are considered by Powell (2003), can be obtained by using the first $2n+1$ components of (2.12).

Any polynomial model $m \in \mathcal{P}^{d,n}$ is defined by $\boldsymbol{\phi}$ and coefficients $\boldsymbol{a} \in \mathbb{R}^{\dim(\mathcal{P}^{d,n})}$ through

$$m(\boldsymbol{x}) = \sum_{i=1}^{\dim(\mathcal{P}^{d,n})} a_i\phi_i(\boldsymbol{x}). \tag{2.13}$$

Given a set of p points $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p\}$, a model that interpolates f on $\boldsymbol{Y}$ is defined by the solution $\boldsymbol{a}$ to

$$\boldsymbol{\Phi}(\boldsymbol{Y})\boldsymbol{a} = \begin{bmatrix}\boldsymbol{\phi}(\boldsymbol{y}_1) & \cdots & \boldsymbol{\phi}(\boldsymbol{y}_p)\end{bmatrix}^{\mathrm{T}}\boldsymbol{a} = \begin{bmatrix} f(\boldsymbol{y}_1) \\ \vdots \\ f(\boldsymbol{y}_p)\end{bmatrix}. \tag{2.14}$$

The existence, uniqueness and conditioning of a solution to (2.14) depend on the location of the sample points $\boldsymbol{Y}$ through the matrix $\boldsymbol{\Phi}(\boldsymbol{Y})$. We note that when $n > 1$, $|\boldsymbol{Y}| = \dim(\mathcal{P}^{d,n})$ is insufficient for guaranteeing that $\boldsymbol{\Phi}(\boldsymbol{Y})$ is non-singular (Wendland 2005). Instead, additional conditions, effectively on the geometry of the sample points $\boldsymbol{Y}$, must be satisfied.

Simplex gradients and linear interpolation models. The geometry conditions needed to uniquely define a linear model are relatively straightforward: the sample points $\boldsymbol{Y}$ must be affinely independent; that is, the columns of

$$\boldsymbol{Y}_{-1} = \begin{bmatrix} \boldsymbol{y}_2 - \boldsymbol{y}_1 & \cdots & \boldsymbol{y}_{n+1} - \boldsymbol{y}_1 \end{bmatrix} \tag{2.15}$$

must be linearly independent. Such sample points define what is referred to as a simplex gradient $\boldsymbol{g}$ through $\boldsymbol{g} = [a_2, \dots, a_{n+1}]^{\mathrm{T}}$, when the monomial basis ϕ is used in (2.14).

Simplex gradients can be viewed as a generalization of first-order finite-difference estimates (*e.g.* the forward differences based on evaluations at the points $\{\boldsymbol{y}_1, \boldsymbol{y}_1 + \Delta\boldsymbol{e}_1, \dots, \boldsymbol{y}_1 + \Delta\boldsymbol{e}_n\}$); their use in optimization algorithms dates at least back to the work of Spendley *et al.* (1962) that inspired Nelder and Mead (1965). Other example usage includes pattern search (Custódio and Vicente 2007, Custódio *et al.* 2008) and noisy optimization (Kelley 1999*b*, Bortz and Kelley 1998); the study of simplex gradients continues with recent works such as those of Regis (2015) and Coope and Tappenden (2019).

Provided that (2.15) is non-singular, it is straightforward to show that linear interpolation models are $\boldsymbol{\kappa}$-fully linear model of f in a neighbourhood of $\boldsymbol{y}_1$. In particular, if $\boldsymbol{Y} \subset \mathcal{B}(\boldsymbol{y}_1; \Delta)$ and f has an L_{g}-Lipschitz-continuous first derivative on an open domain containing $\mathcal{B}(\boldsymbol{y}_1; \Delta)$, then (2.9) holds on $\mathcal{B}(\boldsymbol{y}_1; \Delta)$ with

$$\kappa_{\mathrm{eg}} = L_{\mathrm{g}}(1 + \sqrt{n}\Delta\|\boldsymbol{Y}_{-1}^{-1}\|/2) \quad \text{and} \quad \kappa_{\mathrm{ef}} = L_{\mathrm{g}}/2 + \kappa_{\mathrm{eg}}. \tag{2.16}$$

The expressions in (2.16) also provide a recipe for obtaining a model with a potentially tighter error bound over $\mathcal{B}(\boldsymbol{y}_1; \Delta)$: modify $\boldsymbol{Y} \subset \mathcal{B}(\boldsymbol{y}_1; \Delta)$ to decrease $\|\boldsymbol{Y}_{-1}^{-1}\|$. We note that when $\boldsymbol{Y}_{-1}$ contains orthonormal directions scaled by Δ, one recovers $\kappa_{\mathrm{eg}} = L_{\mathrm{g}}(1 + \sqrt{n}/2)$ and $\kappa_{\mathrm{ef}} = L_{\mathrm{g}}(3 + \sqrt{n})/2$, which is the least value one can obtain from (2.16) given the restriction that $\boldsymbol{Y} \subset \mathcal{B}(\boldsymbol{y}_1; \Delta)$. Hence, by performing LU or QR factorization with pivoting, one can obtain directions (which are then scaled by Δ) in order to improve the conditioning of $\boldsymbol{Y}_{-1}^{-1}$ and hence the approximation bound. Such an approach is performed by Conn, Scheinberg and Vicente (2008*a*) for linear models and by Wild and Shoemaker (2011) for fully linear radial basis function models.

The geometric conditions on $\boldsymbol{Y}$, induced by the approximation bounds in (2.9) or (2.10), can be viewed as playing a similar role to the geometric conditions (*e.g.* positive spanning) imposed on $\boldsymbol{D}$ in directional direct-search methods. Naturally, the choice of basis function used for any model affects the quantitative measure of that model's quality.

Note that many practical methods employ interpolation sets contained within a constant multiple of the trust-region radius (*i.e.* $\boldsymbol{Y} \subset \mathcal{B}(\boldsymbol{y}_1; c_1\Delta)$ for a constant $c_1 \in [1, \infty)$).

Quadratic interpolation models. Quadratic interpolation models have been used for derivative-free optimization for at least fifty years (Winfield 1969, Winfield 1973) and were employed by a series of methods that revitalized interest in model-based methods; see, for example, Conn and Toint (1996), Conn, Scheinberg and Toint (1997*b*), Conn, Scheinberg and Toint (1997*a*) and Powell (1998*b*, 2002).

Of course, the quality of an interpolation model (quadratic or otherwise) in a region of interest is determined by the position of the underlying points being interpolated. For example, if a model m interpolates a function f at points far away from a certain region of interest, the model value may differ greatly from the value of f in that region. Λ-poisedness is a concept to measure how well a set of points is dispersed through a region of interest, and ultimately how well a model will estimate the function in that region.

The most commonly used metric for quantifying how well points are positioned in a region of interest is based on Lagrange polynomials. Given a set of p points $\boldsymbol{Y} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_p\}$, a basis of Lagrange polynomials satisfies

$$\ell_j(\boldsymbol{y}_i) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases} \tag{2.17}$$

We now define Λ-poisedness. A set of points $\boldsymbol{Y}$ is said to be Λ-poised on a set $\boldsymbol{B}$ if $\boldsymbol{Y}$ is linearly independent and the Lagrange polynomials $\{\ell_1, \dots, \ell_p\}$ associated with $\boldsymbol{Y}$ satisfy

$$\Lambda \geq \max_{1 \leq i \leq p} \max_{\boldsymbol{x} \in B} |\ell_i(\boldsymbol{x})|. \tag{2.18}$$

(For an equivalent definition of Λ-poisedness, see Conn *et al.* (2009*b*, Definition 3.6).) Note that the definition of Λ-poisedness is independent of the function being modelled. Also, the points $\boldsymbol{Y}$ need not necessarily be elements of the set $\boldsymbol{B}$. Also, note that if a model is poised on a set $\boldsymbol{B}$, it is poised on any subset of $\boldsymbol{B}$. One is usually interested in the least value of Λ so that (2.18) holds.

Powell's unconstrained optimization by quadratic approximation method (UOBYQA) follows such an approach in maximizing the Lagrange polynomials. In Powell (1998*b*), Powell (2001) and Powell (2002), significant care is given to the linear algebra expense associated with this maximization and the associated change of basis as the methods change their interpolation sets. For example, in Powell (1998*b*), particular sparsity in the Hessian approximation is employed with the aim of capturing curvature while keeping linear algebraic expenses low.

Maintaining, and the question of to what extent it is necessary to maintain, this geometry for quadratic models has been intensely studied; see, for example, Fasano, Morales and Nocedal (2009), Marazzi and Nocedal (2002), D'Ambrosio, Nannicini and Sartor (2017) and Scheinberg and Toint (2010).

Underdetermined quadratic interpolation models. A fact not to be overlooked in the context of derivative-free optimization is that employing an interpolation set $\boldsymbol{Y}$ requires availability of the $|\boldsymbol{Y}|$ function values $\{f(\boldsymbol{y}_i) : \boldsymbol{y}_i \in \boldsymbol{Y}\}$. When the function f is computationally expensive to evaluate, the $(n+1)(n+2)/2$ points required by fully quadratic models can be a burden, potentially with little benefit, to obtain repeatedly in an optimization algorithm.

Beginning with Powell (2003), Powell investigated quadratic models constructed from fewer than $(n+1)(n+2)/2$ points. The most successful of these strategies was detailed in Powell (2004*a*) and Powell (2004*b*) and resolved the $(n+1)(n+2)/2 - |\boldsymbol{Y}|$ remaining degrees of freedom by solving problems of the form

$$\begin{aligned} &\underset{m \in \mathcal{P}^{2,n}}{\text{minimize}} && \|\nabla^2 m(\check{\boldsymbol{x}}) - \boldsymbol{H}\|_F^2 \\ &\text{subject to} && m(\boldsymbol{y}_i) = f(\boldsymbol{y}_i), \quad \text{for all } \boldsymbol{y}_i \in \boldsymbol{Y} \end{aligned} \tag{2.19}$$

to obtain a model m about a point of interest $\check{\boldsymbol{x}}$. Solutions to (2.19) are models with a Hessian closest in Frobenius norm to a specified $\boldsymbol{H} = \boldsymbol{H}^{\mathrm{T}}$ among all models that interpolate f on $\boldsymbol{Y}$. A popular implementation of this strategy is the NEWUOA solver (Powell 2006).

By using the basis

$$\begin{aligned} \boldsymbol{\phi}(\check{\boldsymbol{x}} + \boldsymbol{x}) &= \big[\boldsymbol{\phi}_{\mathrm{fg}}(\check{\boldsymbol{x}} + \boldsymbol{x})^{\mathrm{T}} \quad \big| \, \boldsymbol{\phi}_{\mathrm{H}}(\check{\boldsymbol{x}} + \boldsymbol{x})^{\mathrm{T}}\big]^{\mathrm{T}} \\ &= \left[1, x_1, \ldots, x_n \quad \middle| \, \frac{1}{2}x_1^2, \ldots, \frac{1}{2}x_n^2, \frac{1}{\sqrt{2}}x_1x_2, \ldots, \frac{1}{\sqrt{2}}x_{n-1}x_n\right]^{\mathrm{T}}, \end{aligned} \tag{2.20}$$

the problem (2.19) is equivalent to the problem

$$\begin{aligned} &\underset{\boldsymbol{a}_{\mathrm{fg}}, \boldsymbol{a}_{\mathrm{H}}}{\text{minimize}} && \|\boldsymbol{a}_{\mathrm{H}}\|_2^2 \\ &\text{subject to} && \boldsymbol{a}_{\mathrm{fg}}^{\mathrm{T}}\boldsymbol{\phi}_{\mathrm{fg}}(\boldsymbol{y}_i) + \boldsymbol{a}_{\mathrm{H}}^{\mathrm{T}}\boldsymbol{\phi}_{\mathrm{H}}(\boldsymbol{y}_i) = f(\boldsymbol{y}_i) - \frac{1}{2}\boldsymbol{y}_i^{\mathrm{T}}\boldsymbol{H}\boldsymbol{y}_i, \quad \text{for all } \boldsymbol{y}_i \in \boldsymbol{Y}. \end{aligned} \tag{2.21}$$

Existence and uniqueness of solutions to (2.21) again depend on the positioning of the points in $\boldsymbol{Y}$. Notably, a necessary condition for there to be a unique minimizer of the seminorm is that at least $n+1$ of the points in $\boldsymbol{Y}$ be affinely independent. Lagrange polynomials can be defined for this case; Conn, Scheinberg and Vicente (2008*b*) establish conditions for Λ-poisedness (and hence a fully linear, or better, approximation quality) of such models.

Powell (2004*c*, 2007, 2008) develops efficient solution methodologies for (2.21) when $\boldsymbol{H}$ and m are constructed from interpolation sets that differ by at most one point, and employ these updates in NEWUOA and subsequent solvers. Wild (2008*b*) and Custódio *et al.* (2009) use $\boldsymbol{H} = \boldsymbol{0}$ in order to obtain tighter fully linear error bounds of models resulting from (2.21). A strategy of using even fewer interpolation points (including those in a proper subspace of $\mathbb{R}^n$) is developed by Powell (2013) and Zhang (2014).

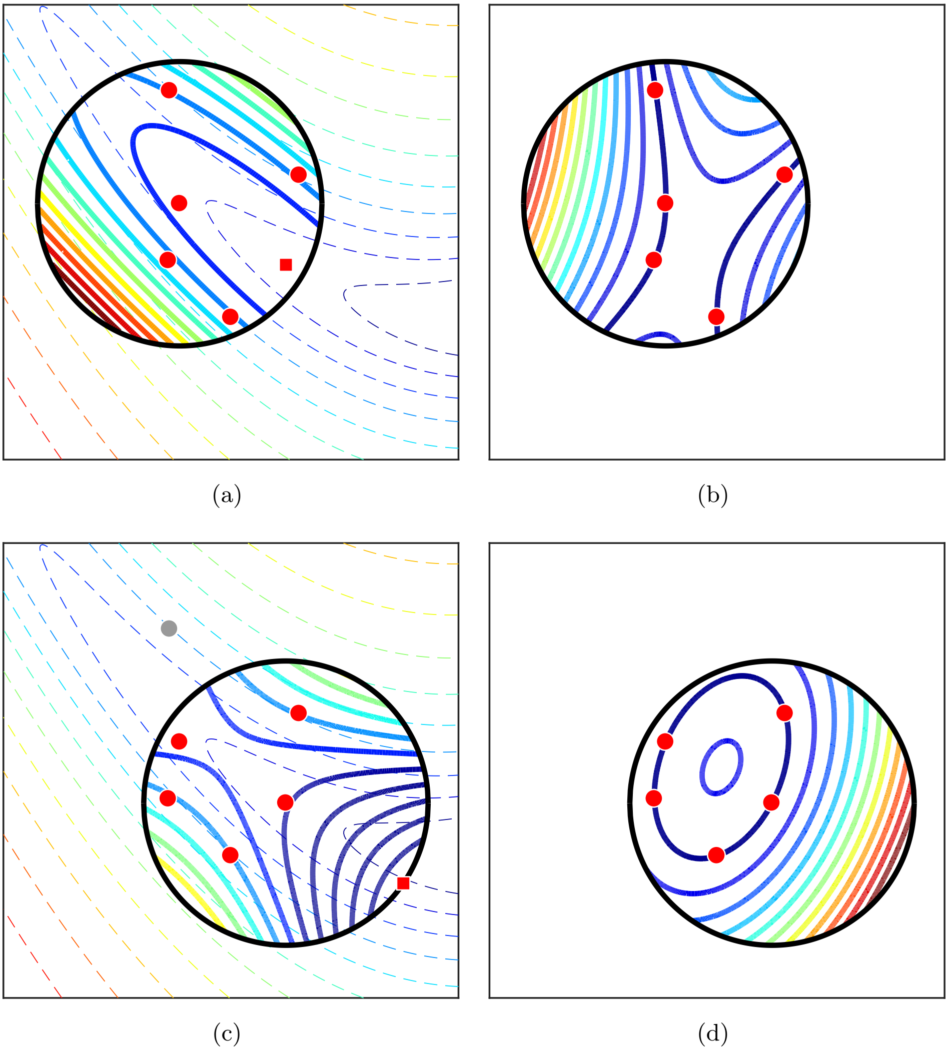

Figure 2.2. (a) Minimum-norm-Hessian model through five points in $\mathcal{B}(\boldsymbol{x}_k; \Delta_k)$ and its minimizer. (b) Absolute value of a sixth Lagrange polynomial for the five points. (c) Minimum-norm-Hessian model through five points in $\mathcal{B}(\boldsymbol{x}_{k+1}; \Delta_{k+1})$ and its minimizer. (d) Absolute value of a sixth Lagrange polynomial for the five points.

In Section 5.2, we summarize approaches that exploit knowledge of sparsity of the derivatives of f in building quadratic models that interpolate fewer than $(n+1)(n+2)/2$ points.

Figure 2.2 shows quadratic models in two dimensions that interpolate $(n+1)(n+2)/2 - 1 = 5$ points as well as the associated magnitude of the remaining Lagrange polynomial (note that this polynomial vanishes at the five interpolated points).

Regression models. Just as one can establish approximation bounds and geometry conditions when $\boldsymbol{Y}$ is linearly independent, the same can be done for overdetermined regression models (Conn *et al.* 2008*b*, Conn *et al.* 2009*b*). This can be accomplished by extending the definition of Lagrange polynomials from (2.17) to the regression case. That is, given a basis $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^{\dim(\mathcal{P}^{d,n})}$ and points $\boldsymbol{Y} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_p\} \subset \mathbb{R}^n$ with $p > \dim(\mathcal{P}^{d,n})$, the set of polynomials satisfies

$$\ell_j(\boldsymbol{y}_i) \overset{\text{l.s.}}{=} \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \tag{2.22}$$

where $\overset{\text{l.s.}}{=}$ denotes the least-squares solution. The regression model can be recovered finding the least-squares solution (now overdetermined) system from (2.14), and the definition of Λ-poisedness (in the regression sense) is equivalent to (2.18). Ultimately, given a linear regression model through a set of Λ-poised points $\boldsymbol{Y} \subset \mathcal{B}(\boldsymbol{y}_1; \Delta)$, and if f has an L_{g}-Lipschitz-continuous first derivative on an open domain containing $\mathcal{B}(\boldsymbol{y}_1; \Delta)$, then (2.9) holds on $\mathcal{B}(\boldsymbol{y}_1; \Delta)$ with

$$\kappa_{\mathrm{eg}} = \frac{5}{2}\sqrt{p} L_{\mathrm{g}} \Lambda \quad \text{and} \quad \kappa_{\mathrm{ef}} = \frac{1}{2} L_{\mathrm{g}} + \kappa_{\mathrm{eg}}. \tag{2.23}$$

Conn *et al.* (2008*b*) note the fact that the extension of Lagrange polynomials does not apply to the 1-norm or infinity-norm case. Billups, Larson and Graf (2013) show that the definition of Lagrange polynomials can be extended to the weighted regression case. Verdério, Karas, Pedroso and Scheinberg (2017) show that (2.9) can also be recovered for support vector regression models.

Efficiently minimizing the model (regardless of type) over a trust region is integral to the usefulness of such models within an optimization algorithm. In fact, this necessity is a primary reason for the use of low-degree polynomial models by the majority of derivative-free trust-region methods. For quadratic models, the resulting subproblem remains one of the most difficult non-convex optimization problems solvable in polynomial time, as illustrated by Moré and Sorensen (1983). As exemplified by Powell (1997), the implementation of subproblem solvers is a key concern in methods seeking to perform as few algebraic operations between function evaluations as possible.

2.2.3. Radial basis function interpolation models

An additional way to model non-linearity with potentially less restrictive geometric conditions is by using radial basis functions (RBFs). Such models take the form

$$m(\boldsymbol{x}) = \sum_{i=1}^{|\boldsymbol{Y}|} b_i \psi(\|\boldsymbol{x} - \boldsymbol{y}_i\|) + \boldsymbol{a}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}), \tag{2.24}$$

where $\psi : \mathbb{R}_+ \to \mathbb{R}$ is a conditionally positive-definite univariate function and $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})$ represents a (typically low-order) polynomial as before; see, for example, Buhmann (2000). Given a sample set $\boldsymbol{Y}$, RBF model coefficients $(\boldsymbol{a}, \boldsymbol{b})$ can be obtained by solving the augmented interpolation equations

$$\begin{bmatrix} \psi(\|\boldsymbol{y}_1 - \boldsymbol{y}_1\|) & \cdots & \psi(\|\boldsymbol{y}_1 - \boldsymbol{y}_{|\boldsymbol{Y}|}\|) & \boldsymbol{\phi}(\boldsymbol{y}_1)^{\mathrm{T}} \\ \vdots & & \vdots & \vdots \\ \psi(\|\boldsymbol{y}_{|\boldsymbol{Y}|} - \boldsymbol{y}_1\|) & \cdots & \psi(\|\boldsymbol{y}_{|\boldsymbol{Y}|} - \boldsymbol{y}_{|\boldsymbol{Y}|}\|) & \boldsymbol{\phi}(\boldsymbol{y}_{|\boldsymbol{Y}|})^{\mathrm{T}} \\ \boldsymbol{\phi}(\boldsymbol{y}_1) & \cdots & \boldsymbol{\phi}(\boldsymbol{y}_{|\boldsymbol{Y}|}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} f(\boldsymbol{y}_1) \\ \vdots \\ f(\boldsymbol{y}_{|\boldsymbol{Y}|}) \\ \mathbf{0} \end{bmatrix}. \tag{2.25}$$

That RBFs are conditionally positive-definite ensures that (2.25) is non-singular provided that the degree d of the polynomial $\boldsymbol{\phi}$ is sufficiently large and that $\boldsymbol{Y}$ is poised for degree-d polynomial interpolation. For example, cubic ($\psi(r) = r^3$) RBFs require a linear polynomial; multiquadric ($\psi(r) = -(\gamma^2 + r^2)^{1/2}$) RBFs require a constant polynomial; and inverse multiquadric ($\psi(r) = (\gamma^2 + r^2)^{-1/2}$) and Gaussian ($\psi(r) = \exp(-\gamma^{-2}r^2)$) RBFs do not require a polynomial. Consequently, RBFs have relatively unrestrictive geometric requirements on the interpolation points $\boldsymbol{Y}$ while allowing for modelling a wide range of non-linear behaviour.

This feature is typically exploited in global optimization (see *e.g.* Björkman and Holmström 2000, Gutmann 2001 and Regis and Shoemaker 2007), whereby an RBF surrogate model is employed to globally approximate f. However, works such as Oeuvray and Bierlaire (2009), Oeuvray (2005), Wild (2008*a*) and Wild and Shoemaker (2013) establish and use local approximation properties of these models. This approach is typically performed by relying on a linear polynomial $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})$, which can be used to establish that the RBF model in (2.24) can be a fully linear local approximation of smooth f.

2.2.4. Trust-region methods

Having discussed issues of model construction, we are now ready to present a general statement of a model-based trust-region method in Algorithm 3.

A distinguishing characteristic of derivative-free model-based trust-region methods is how they manage $\boldsymbol{Y}_k$, the set of points used to construct the model m_k. Some methods ensure that $\boldsymbol{Y}_k$ contains a scaled stencil of points around $\boldsymbol{x}_k$; such an approach can be attractive since the objective at such points can be evaluated in parallel. A fixed stencil can also ensure that all models sufficiently approximate the objective. Other methods construct $\boldsymbol{Y}$ by using previously evaluated points near $\boldsymbol{x}_k$, for example, those points

Algorithm 3: Derivative-free model-based trust-region method

```
Set parameters $\epsilon > 0$, $0 < \gamma_{\mathrm{dec}} < 1 \leq \gamma_{\mathrm{inc}}$, $0 < \eta_0 \leq \eta_1 < 1$, $\Delta_{\max}$
Choose initial point $\boldsymbol{x}_0$, trust-region radius $0 < \Delta_0 \leq \Delta_{\max}$, and set of
  previously evaluated points $\boldsymbol{Y}_k$
for $k = 0, 1, 2 \ldots$ do
    Select a subset of $\boldsymbol{Y}_k$ (or augment $\boldsymbol{Y}_k$ and evaluate $f$ at new points)
      for model building
    Build a model $m_k$ using points in $\boldsymbol{Y}_k$ and their function values
    while $\|\nabla m_k(\boldsymbol{x}_k)\| < \epsilon$ do
        if $m_k$ is accurate on $\mathcal{B}(\boldsymbol{x}_k; \Delta_k)$ then
            $\Delta_k \leftarrow \gamma_{\mathrm{dec}} \Delta_k$
        else
            By updating $\boldsymbol{Y}_k$, make $m_k$ accurate on $\mathcal{B}(\boldsymbol{x}_k; \Delta_k)$
    Generate a direction $\boldsymbol{s}_k \in \mathcal{B}(\boldsymbol{0}; \Delta_k)$ so that $\boldsymbol{x}_k + \boldsymbol{s}_k$ approximately
      minimizes $m_k$ on $\mathcal{B}(\boldsymbol{x}_k; \Delta_k)$
    Evaluate $f(\boldsymbol{x}_k + \boldsymbol{s}_k)$ and $\rho_k \leftarrow \dfrac{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}$
    if $\rho_k < \eta_1$ and $m_k$ is inaccurate on $\mathcal{B}(\boldsymbol{x}_k; \Delta_k)$ then
        Add model improving point(s) to $\boldsymbol{Y}_k$
    if $\rho_k \geq \eta_1$ then
        $\Delta_{k+1} \leftarrow \min\{\gamma_{\mathrm{inc}} \Delta_k, \Delta_{\max}\}$
    else if $m_k$ is accurate on $\mathcal{B}(\boldsymbol{x}_k; \Delta_k)$ then
        $\Delta_{k+1} \leftarrow \gamma_{\mathrm{dec}} \Delta_k$
    else
        $\Delta_{k+1} \leftarrow \Delta_k$
    if $\rho_k \geq \eta_0$ then $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \boldsymbol{s}_k$ else $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k$
    $\boldsymbol{Y}_{k+1} \leftarrow \boldsymbol{Y}_k$
```

within $\mathcal{B}(\boldsymbol{x}_k; c_1\Delta_k)$ for some constant $c_1 \in [1, \infty)$. Depending on the set of previously evaluated points, such methods may need to add points to $\boldsymbol{Y}_k$ that most improve the model quality. Determining which additional points to add to $\boldsymbol{Y}_k$ can be computationally expensive, but the method should be willing to do so in the hope of needing fewer evaluations of the objective function at new points in $\boldsymbol{Y}_k$. Most methods do not ensure that models are valid on every iteration but rather make a single step toward improving the model. Such an approach can ensure a high-quality model in a finite number of improvement steps. (Exceptional methods that ensure model quality before $\boldsymbol{s}_k$ is calculated are the methods of Powell and manifold sampling of Khan, Larson and Wild (2018).) The ORBIT method (Wild, Regis and Shoemaker 2008) places a limit on the size of $\boldsymbol{Y}_k$ (*e.g.* in order to limit the amount of linear algebra or to prevent overfitting). In the end, such restrictions on $\boldsymbol{Y}_k$ may determine whether m_k is an interpolation or regression model.

Derivative-free trust-region methods share many similarities with traditional trust-region methods, for example, the use of a ρ-test to determine whether a step is taken or rejected. As in a traditional trust-region method, the ρ-test measures the ratio of actual decrease observed in the objective versus the decrease predicted by the model.

On the other hand, the management of the trust-region radius parameter Δ_k in Algorithm 3 differs remarkably from traditional trust-region methods. Derivative-free variants require an additional test of model quality, the failure of which results in shrinking Δ_k. When derivatives are available, Taylor's theorem ensures model accuracy for small Δ_k. In the derivative-free case, such a condition must be explicitly checked in order to ensure that Δ_k does not go to zero merely because the model is poor, hence the inclusion of tests of model quality. As a direct result of these considerations, $\Delta_k \to 0$ as Algorithm 3 converges; this is generally not the case in traditional trust-region methods.

As in derivative-based trust-region methods, the solution to the trust-region subproblem in line 11 of Algorithm 3 must satisfy a *Cauchy decrease condition.* Given the model m_k used in Algorithm 3, we define the optimal step length in the direction $-\nabla m_k(\boldsymbol{x}_k)$ by

$$t_k^C = \operatorname*{arg\,min}_{t \geq 0 : \boldsymbol{x}_k - t\nabla m_k(\boldsymbol{x}_k) \in \mathcal{B}(\boldsymbol{x}_k;\Delta_k)} m_k(\boldsymbol{x}_k - t\nabla m_k(\boldsymbol{x}_k)),$$

and the corresponding *Cauchy step*

$$\boldsymbol{s}_k^C = -t_k^C \nabla m_k(\boldsymbol{x}_k).$$

It is straightforward to show (see *e.g.* Conn, Scheinberg and Vicente 2009*b*, Theorem 10.1) that

$$m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k^C) \geq \frac{1}{2}\|\nabla m_k(\boldsymbol{x}_k)\| \min\left\{\frac{\|\nabla m_k(\boldsymbol{x}_k)\|}{\|\nabla^2 m_k(\boldsymbol{x}_k)\|}, \Delta_k\right\}. \tag{2.26}$$

That is, (2.26) states that, provided that both $\Delta_k \approx \|\nabla m_k(\boldsymbol{x}_k)\|$ and a uniform bound exists on the norm of the model Hessian, the model decrease attained by the Cauchy step $\boldsymbol{s}_k^C$ is of the order of Δ_k^2. In order to prove convergence, it is desirable to ensure that each step $\boldsymbol{s}_k$ generated in line 11 of Algorithm 3 decreases the model m_k by no less than $\boldsymbol{s}_k^C$ does, or at least some fixed positive fraction of the decrease achieved by $\boldsymbol{s}_k^C$. Because successful iterations ensure that the actual decrease attained in an iteration is at least a constant fraction of the model decrease, the sequence of decreases of Algorithm 3 are square-summable, provided that $\Delta_k \to 0$. (This is indeed the case for derivative-free trust-region methods.) Hence, in most theoretical treatments of these methods, it is commonly stated as an assumption that the subproblem solution $\boldsymbol{s}_k$ obtained in line 11 of Algorithm 3 satisfies

$$m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k) \geq \kappa_{\text{fcd}}(m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k^C)), \tag{2.27}$$

where $\kappa_{\text{fcd}} \in (0, 1]$ is the fraction of the Cauchy decrease. In practice, when m_k is a quadratic model, subproblem solvers have been well studied and often come with guarantees concerning the satisfaction of (2.27) (Conn, Gould and Toint 2000). Wild *et al.* (2008, Figure 4.3) demonstrate the satisfaction of an assumption like (2.27) when the model m_k is a radial basis function.

Under reasonable smoothness assumptions, most importantly $f \in \mathcal{LC}^1$, algorithms in the Algorithm 3 framework have been shown to be first-order convergent (*i.e.* (2.4)) and second-order convergent (*i.e.* (2.6)), with the (arguably) most well-known proof given by Conn, Scheinberg and Vicente (2009*a*). In more recent work, Garmanjani, Jùdice and Vicente (2016) provide a WCC bound of the form (2.2) for Algorithm 3, recovering essentially the same upper bound on the number of function evaluations required by DDS methods found in Vicente (2013), that is, a WCC bound in $O(\epsilon^{-2})$ (see Table A.1). When $f \in \mathcal{LC}^2$, Gratton, Royer and Vicente (2019*a*) demonstrate a second-order WCC bound of the form (2.3) in $O(\epsilon^{-3})$; in order to achieve this result, fully quadratic models m_k are required. In Section 3.3, a similar result is achieved by using randomized variants that do not require a fully quadratic model in every iteration.

Early analysis of Powell's UOBYQA method shows that, with minor modifications, the algorithm can converge superlinearly in neighbourhoods of strict convexity (Han and Liu 2004). A key distinction between Powell's methods and other model-based trust-region methods is the use of separate neighbourhoods for model quality and trust-region steps, with each of these neighbourhoods changing dynamically. Convergence of such methods is addressed by Powell (2010, 2012).

The literature on derivative-free trust-region methods is extensive. We mention in passing several additional classes of trust-region methods that have not fallen neatly into our discussion thus far. *Wedge methods* (Marazzi and Nocedal 2002) explicitly enforce geometric properties (Λ-poisedness) of the sample set between iterations by adding additional constraints to the trust-region subproblem. Alexandrov, Dennis, Jr, Lewis and Torczon (1998) consider a trust-region method utilizing a hierarchy of model approximations. In particular, if derivatives can be obtained but are expensive, then the method of Alexandrov *et al.* (1998) uses a model that interpolates not only zeroth-order information but also first-order (gradient) information. For problems with deterministic noise, Elster and Neumaier (1995) propose a method that projects the solutions of a trust-region subproblem onto a dynamically refined grid, encouraging better practical behaviour. Similarly, for problems with deterministic noise, Maggiar, Wächter, Dolinskaya and Staum (2018) propose a model-based trust-region method that implicitly convolves the objective function with a Gaussian kernel, again yielding better practical behaviour.

2.3. Hybrid methods and miscellanea

While the majority of work in derivative-free methods for deterministic problems can be classified as direct-search or model-based methods, some work defies this simple classification. In fact, several works (Conn and Le Digabel 2013, Custódio *et al.* 2009, Dennis, Jr and Torczon 1997, Frimannslund and Steihaug 2011) propose methods that seem to hybridize these two classes, existing somewhere in the intersection. For example, Custódio and Vicente (2005) and Custódio *et al.* (2009) develop the SID-PSM method, which extends Algorithm 2 so that the search step consists of minimizing an approximate quadratic model of the objective (obtained either by minimum-Frobenius norm interpolation or by regression) over a trust region. Here, we highlight methods that do not neatly belong to the two aforementioned classes of methods.

2.3.1. Finite differences

As noted in Section 1.1.2, many of the earliest derivative-free methods employed finite-difference-based estimates of derivatives. The most popular first-order directional derivative estimates include the forward/reverse difference

$$\delta_{\mathrm{f}}(f;\boldsymbol{x};\boldsymbol{d};h) = \frac{f(\boldsymbol{x}+h\boldsymbol{d}) - f(\boldsymbol{x})}{h} \tag{2.28}$$

and central difference

$$\delta_{\mathrm{c}}(f;\boldsymbol{x};\boldsymbol{d};h) = \frac{f(\boldsymbol{x}+h\boldsymbol{d}) - f(\boldsymbol{x}-h\boldsymbol{d})}{2h}, \tag{2.29}$$

where $h \neq 0$ is the difference parameter and the non-trivial $\boldsymbol{d} \in \mathbb{R}^n$ defines the direction. Several recent methods, including the methods described in Sections 2.3.2, 2.3.3 and 3.1.2, use such estimates and employ difference parameters or directions that dynamically change.

As an example of a potentially dynamic choice of difference parameter, we consider the usual case of roundoff errors. We denote by $f'_\infty(\boldsymbol{x};\boldsymbol{d})$ the directional derivative at $\boldsymbol{x}$ of the infinite-precision (*i.e.* based on real arithmetic) objective function f_∞ in the unit direction $\boldsymbol{d}$ (*i.e.* $\|\boldsymbol{d}\| = 1$). We then have the following error for forward or reverse finite-difference estimates based on the function f available through computation:

$$|\delta_{\mathrm{f}}(f;\boldsymbol{x};\boldsymbol{d};h) - f'_\infty(\boldsymbol{x};\boldsymbol{d})| \leq \frac{1}{2}L_{\mathrm{g}}(\boldsymbol{x})|h| + 2\frac{\epsilon_\infty(\boldsymbol{x})}{|h|}, \tag{2.30}$$

provided that $|f''_\infty(\cdot;\boldsymbol{d})| \leq L_{\rm g}(\boldsymbol{x})$ and $|f_\infty(\cdot) - f(\cdot)| \leq \epsilon_\infty(\boldsymbol{x})$ on the interval $[\boldsymbol{x}, \boldsymbol{x} + h\boldsymbol{d}]$. In Gill, Murray and Wright (1981) and Gill, Murray, Saunders and Wright (1983), the recommended difference parameter is $h = 2\sqrt{\epsilon_\infty(\boldsymbol{x})/L_{\rm g}(\boldsymbol{x})}$, which yields the minimum value $2\sqrt{\epsilon_\infty(\boldsymbol{x})L_{\rm g}(\boldsymbol{x})}$ of the upper bound in (2.30); when ϵ_∞ is a bound on the roundoff error and $L_{\rm g}$ is of order one, then the familiar $h \in O(\sqrt{\epsilon_\infty})$ is obtained.

Similarly, if one models the error between f_∞ and f as a stationary stochastic process (through the ansatz denoted by $f_{\boldsymbol{\xi}}$) with variance $\epsilon_{\rm f}(\boldsymbol{x})^2$, minimizing the upper bound on the mean-squared error,

$$\mathbb{E}_{\boldsymbol{\xi}}[(\delta_{\rm f}(f_{\boldsymbol{\xi}};\boldsymbol{x};\boldsymbol{d};h) - f'_\infty(\boldsymbol{x};\boldsymbol{d}))^2] \leq \frac{1}{4}L_{\rm g}(\boldsymbol{x})^2h^2 + 2\frac{\epsilon_{\rm f}(\boldsymbol{x})^2}{h^2}, \tag{2.31}$$

yields the choice $h = (\sqrt{8}\epsilon_{\rm f}(\boldsymbol{x})/L_{\rm g}(\boldsymbol{x}))^{1/2}$ with an associated root-mean-squared error of $(\sqrt{2}\epsilon_\infty(\boldsymbol{x})L_{\rm g}(\boldsymbol{x}))^{1/2}$; see, for example, Moré and Wild (2012, 2014). A rough procedure for computing $\epsilon_{\rm f}$ is provided in Moré and Wild (2011) and used in recent methods such as that of Berahas, Byrd and Nocedal (2019).

In both cases (2.30) and (2.31), the first-order error is $c\sqrt{\epsilon(\boldsymbol{x})L_{\rm g}(\boldsymbol{x})}$ (for a constant $c \leq 2$), which can be used to guide the decision on whether the derivatives estimates are of sufficient accuracy.

2.3.2. Implicit filtering

Implicit filtering is a hybrid of a grid-search algorithm (evaluating all points on a lattice) and a Newton-like local optimization method. The gradient (and possible Hessian) estimates for local optimization are approximated by the central differences $\{\delta_{\rm c}(f;\boldsymbol{x}_k;\boldsymbol{e}_i;\Delta_k) : i = 1,\dots,n\}$. The difference parameter Δ_k decreases when implicit filtering encounters a *stencil failure* at $\boldsymbol{x}_k$, that is,

$$f(\boldsymbol{x}_k) \leq f(\boldsymbol{x}_k \pm \Delta_k\boldsymbol{e}_i), \tag{2.32}$$

where $\boldsymbol{e}_i$ is the ith elementary basis vector. This is similar to direct-search methods, but notice that implicit filtering is not polling opportunistically: all polling points are evaluated on each iteration. The basic version of implicit filtering from Kelley (2011) is outlined in Algorithm 4. Note that most implementations of implicit filtering require a bound-constrained domain.

Considerable effort has been devoted to extensions of Algorithm 4 when f is 'noisy'. Gilmore and Kelley (1995) show that implicit filtering converges to local minima of (DET) when the objective f is the sum of a smooth function f_s and a high-frequency, low-amplitude function f_n, with $f_n \to 0$ quickly in a neighbourhood of all minimizers of f_s. Under similar assumptions, Choi and Kelley (2000) show that Algorithm 4 converges superlinearly if the step sizes Δ_k are defined as a power of the norm of the previous iteration's gradient approximation.

Algorithm 4: Implicit-filtering method

```
Set parameters feval_max $> 0$, $\Delta_{\min} > 0$, $\gamma_{\text{dec}} \in (0,1)$ and $\tau > 0$
Choose initial point $\boldsymbol{x}_0$ and step size $\Delta_0 \geq \Delta_{\min}$
$k \leftarrow 0$; evaluate $f(\boldsymbol{x}_0)$ and set fevals $\leftarrow 1$
while fevals $\leq$ feval_max and $\Delta_k \geq \Delta_{\min}$ do
    Evaluate $f(\boldsymbol{x}_k \pm \Delta_k \boldsymbol{e}_i)$ for $i \in \{1, \dots, n\}$ and approximate $\nabla f(\boldsymbol{x}_k)$
      via $\{\delta_{\text{c}}(f; \boldsymbol{x}_k; \boldsymbol{e}_i; \Delta_k) : i = 1, \dots, n\}$
    if equation (2.32) is satisfied or $\|\nabla f(\boldsymbol{x}_k)\| \leq \tau \Delta_k$ then
        $\Delta_{k+1} \leftarrow \gamma_{\text{dec}} \Delta_k$
        $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k$
    else
        Update Hessian estimate $\boldsymbol{H}_k$ (or set $\boldsymbol{H}_k \leftarrow \boldsymbol{I}$)
        $\boldsymbol{s}_k \leftarrow -\boldsymbol{H}_k^{-1} \nabla f(\boldsymbol{x}_k)$
        Perform a line search in the direction $\boldsymbol{s}_k$ to generate $\boldsymbol{x}_{k+1}$
        $\Delta_{k+1} \leftarrow \Delta_k$
    $k \leftarrow k + 1$
```

2.3.3. Adaptive regularized methods

Cartis, Gould and Toint (2012) perform an analysis of adaptive regularized cubic (ARC) methods and propose a derivative-free method, ARC-DFO. ARC-DFO is an extension of ARC whereby gradients are replaced with central finite differences of the form (2.29), with the difference parameter monotonically decreasing within a single iteration of the method. ARC-DFO is an intrinsically model-based method akin to Algorithm 3, but the objective within each subproblem regularizes third-order behaviour of the model. Thus, like a trust-region method, ARC-DFO employs trial steps and model gradients. During the main loop of ARC-DFO, if the difference parameter exceeds a constant factor of the minimum of the trial step norm or the model gradient norm, then the difference parameter is shrunk by a constant factor, and the iteration restarts to obtain a new trial step. This mechanism is structurally similar to a derivative-free trust-region method's checks on model quality. Cartis *et al.* (2012) show that ARC-DFO demonstrates a WCC result of type (2.2) in $O(\epsilon^{-3/2})$, the same asymptotic result (in terms of ϵ-dependence) that the authors demonstrate for derivative-based variants of ARC methods. In terms of dependence on ϵ, this result is a strict improvement over the WCC results of the same type demonstrated for DDS and trust-region methods, although this result is proved under the stronger assumption that $f \in \mathcal{LC}^2$.

In a different approach, Hare and Lucet (2013) show convergence of a derivative-free method that penalizes large steps via a proximal regularizer, thereby removing the necessity for a trust region. Lazar and Jarre (2016) regularize their line-search with a term seeking to minimize a weighted change of the model's third derivatives.

2.3.4. Line-search-based methods

Several line-search-based methods for derivative-free optimization have been developed. Grippo, Lampariello and Lucidi (1988) and De Leone, Gaudioso and Grippo (1984) (two of the few papers appearing in the 1980s concerning derivative-free optimization) both analyse conditions on the step sizes used in a derivative-free line-search algorithm, and provide methods for constructing such steps. Lucidi and Sciandrone (2002*b*) present methods that combine pattern-search and line-search approaches in a convergent framework. The VXQR method of Neumaier, Fendl, Schilly and Leitner (2011) performs a line search on a direction computed from a QR factorization of previously evaluated points. Neumaier *et al.* (2011) apply VXQR to problems with $n = 1000$, a large problem dimension among the methods considered here.

Consideration has also been given to non-monotone line-search-based derivative-free methods. Since gradients are not available in derivative-free optimization, the search direction in a line-search method may not be a descent direction. Non-monotone methods allow one to still employ such directions in a globally convergent framework. Grippo and Sciandrone (2007) extend line-search strategies based on coordinate search and the method of Barzilai and Borwein (1988) to develop a globally convergent non-monotone derivative-free method. Grippo and Rinaldi (2014) extend such non-monotone strategies to broader classes of algorithms that employ simplex gradients, hence further unifying direct-search and model-based methods. Another non-monotone line-search method is proposed by Diniz-Ehrhardt, Martínez and Raydan (2008), who encapsulate early examples of randomized DDS methods (Section 3.2).

2.3.5. Methods for non-smooth optimization

In Section 2.1.2, we discuss how MADS handles non-differentiable objective functions by densely sampling directions on a mesh, thereby ensuring that all Clarke directional derivatives are non-negative (*i.e.* (2.1)). Another early analysis of a DDS method on a class of non-smooth objectives was performed by García-Palomares and Rodríguez (2002).

Gradient sampling methods are a developing class of algorithms for general non-smooth non-convex optimization; see the recent survey by Burke *et al.* (2018). These methods attempt to estimate the *ϵ-subdifferential* at a point $\boldsymbol{x}$ by evaluating a random sample of gradients in the neighbourhood of $\boldsymbol{x}$ and constructing the convex hull of these gradients. In a derivative-free setting, the approximation of these gradients is not as immediately obvious in the presence of non-smoothness, but there exist gradient-sampling methods that use finite-difference estimates with specific smoothing techniques (Kiwiel 2010).

In another distinct line of research, Bagirov, Karasözen and Sezer (2007) analyse a derivative-free variant of subgradient descent, where subgradients are approximated via so-called discrete gradients. In Section 5.3, we will further discuss methods for minimizing *composite non-smooth objective functions* of the form $f = h \circ F$, where h is non-smooth but a closed-form expression is known and F is assumed smooth. These methods are characterized by their exploitation of the knowledge of h, making them less general than the methods for non-smooth optimization discussed so far.

3. Randomized methods for deterministic objectives

We now summarize randomized methods for solving (DET). Such methods often have promising theoretical properties, although some practitioners may dislike the non-deterministic behaviour of these methods. We discuss randomization within direct-search methods in Section 3.2 and within trust-region methods in Section 3.3, but we first begin with a discussion of random search as applied to deterministic objectives.

In any theoretical treatment of randomized methods, one must be careful to distinguish between random variables and their realizations. For the sake of terseness in this survey, we will intentionally conflate variables with realizations and refer to respective papers for more careful statements of theoretical results.

3.1. Random search

We highlight two randomized methods for minimizing a deterministic objective: pure random search and Nesterov random search.

3.1.1. Pure random search

Pure random search is a natural method to start with for randomized derivative-free optimization. Pure random search is popular for multiple reasons; in particular, it is easy to implement (with few or no user-defined tolerances), and (if the points generated are independent of one another) it exhibits perfect scaling in terms of evaluating f at many points simultaneously.

A pure random-search method is given in Algorithm 5, where points are generated randomly from $\boldsymbol{\Omega}$. For example, if $\boldsymbol{\Omega} = \{\boldsymbol{x} : \boldsymbol{c}(\boldsymbol{x}) \leq \boldsymbol{0}, \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$, line 3 of Algorithm 5 may involve drawing points uniformly at random from $[\boldsymbol{l}, \boldsymbol{u}]$ and checking whether they satisfy $\boldsymbol{c}(\boldsymbol{x}) \leq \boldsymbol{0}$. If the procedure for generating points in line 3 of Algorithm 5 is independent of the function values observed, then the entire set of points used within pure random

```
Algorithm 5: Pure random search
1 Choose initial point $\hat{\boldsymbol{x}} \in \boldsymbol{\Omega}$, termination test, and point generation
    scheme
2 while Termination test is not satisfied do
3     Generate $\boldsymbol{x} \in \boldsymbol{\Omega}$
4     if $f(\boldsymbol{x}) < f(\hat{\boldsymbol{x}})$ then
5         $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}$
```

search can be generated beforehand: we intentionally omit the index k in the statement of Algorithm 5.

Nevertheless, for the sake of analysis, it is useful to consider an ordering of the sequence of random points generated by Algorithm 5. With such a sequence $\{\boldsymbol{x}_k\}$, one can analyse the best points after N evaluations,

$$\hat{\boldsymbol{x}}_N \in \operatorname*{arg\,min}_{k=1,\ldots,N} f(\boldsymbol{x}_k).$$

If f_* is the global minimum value, then

$$\mathbb{P}[f(\hat{\boldsymbol{x}}_N) \leq f_* + \epsilon] = 1 - \prod_{k=1}^{N} (1 - \mathbb{P}[\boldsymbol{x}_k \in \boldsymbol{\mathcal{L}}_{f_*+\epsilon}(f)]),$$

where $\epsilon \geq 0$ and $\boldsymbol{\mathcal{L}}_\alpha(f) = \{\boldsymbol{x} : f(\boldsymbol{x}) \leq \alpha\}$. Provided that the procedure used to generate points at line 3 of Algorithm 5 satisfies

$$\lim_{N\to\infty} \prod_{k=1}^{N} (1 - \mathbb{P}[\boldsymbol{x}_k \in \boldsymbol{\mathcal{L}}_{f_*+\epsilon}(f)]) = 0$$

for all $\epsilon > 0$, then $f(\hat{\boldsymbol{x}}_k)$ converges in probability to f_*. For example, if each $\boldsymbol{x}_k$ is drawn independently and uniformly over $\boldsymbol{\Omega}$, then one can calculate the number of evaluations required to ensure that the $\hat{\boldsymbol{x}}_k$ returned by Algorithm 5 satisfies $\hat{\boldsymbol{x}}_k \in \boldsymbol{\mathcal{L}}_{f_*+\epsilon}$ with probability $p \in (0, 1)$, that is,

$$N \geq \frac{\log(p)}{\log\left(1 - \dfrac{\mu(\boldsymbol{\mathcal{L}}_{f_*+\epsilon} \bigcap \boldsymbol{\Omega})}{\mu(\boldsymbol{\Omega})}\right)},$$

provided $\mu(\boldsymbol{\mathcal{L}}_{f(\boldsymbol{x}_*)+\epsilon} \cap \boldsymbol{\Omega}) > 0$ and $\boldsymbol{\Omega}$ is measurable.

Random-search methods typically make few assumptions about f; see Zhigljavsky (1991) for further discussion about the convergence of pure random search. Naturally, a method that assumes only that f is measurable on $\boldsymbol{\Omega}$ is likely to produce function values that converge more slowly to f_* when applied to an $f \in \mathcal{C}^0$ than does a method that exploits the continuity of f. Heuristic modifications of random search have sought to improve empirical performance on certain classes of problems, while still maintaining random search's global optimization property; see, for example, the work of Zabinsky and Smith (1992) and Patel, Smith and Zabinsky (1989).

3.1.2. Nesterov random search

We refer to the method discussed in this section as Nesterov random search because of the seminal article by Nesterov and Spokoiny (2017), but the idea driving this method is much older. A similar method, for instance, is discussed in Polyak (1987, Chapter 3.4).

The method of Nesterov random search is largely motivated by Gaussian smoothing. In particular, given a covariance (*i.e.* symmetric positive-definite) matrix $\boldsymbol{B}$ and a smoothing parameter $\mu > 0$, consider the Gaussian smoothed function

$$f_\mu(\boldsymbol{x}) = \sqrt{\frac{\det(\boldsymbol{B})}{(2\pi)^n}} \int_{\mathbb{R}^n} f(\boldsymbol{x} + \mu\boldsymbol{u}) \exp\left(-\frac{1}{2}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{u}\right) \mathrm{d}\boldsymbol{u}.$$

This smoothing has many desirable properties; for instance, if f is Lipschitz-continuous with constant L_{f}, then f_μ is Lipschitz-continuous with a constant no worse than L_{f} for all $\mu > 0$. Likewise, if f has Lipschitz-continuous gradients with constant L_{g}, then f_μ has Lipschitz-continuous gradients with a constant no worse than L_{g} for all $\mu > 0$. If f is convex, then f_μ is convex.

One can show that

$$\nabla f_\mu(\boldsymbol{x}) = \frac{1}{\mu}\sqrt{\frac{\det(\boldsymbol{B})}{(2\pi)^n}} \int_{\mathbb{R}^n} (f(\boldsymbol{x} + \mu\boldsymbol{u}) - f(\boldsymbol{x})) \exp\left(-\frac{1}{2}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{u}\right) \boldsymbol{B}\boldsymbol{u} \, \mathrm{d}\boldsymbol{u}.$$

In other words, $\nabla f_\mu(\boldsymbol{x})$, which can be understood as an approximation of $\nabla f(\boldsymbol{x})$ in the smooth case, can be computed via an expectation over $\boldsymbol{u} \in \mathbb{R}^n$ weighted by the finite difference $f(\boldsymbol{x} + \mu\boldsymbol{u}) - f(\boldsymbol{x})$ and inversely weighted by a radial distance from $\boldsymbol{x}$. With this interpretation in mind, Nesterov and Spokoiny (2017) propose a collection of *random gradient-free oracles*, where one first generates a Gaussian random vector $\boldsymbol{u} \in \mathcal{N}(\boldsymbol{0}, \boldsymbol{B}^{-1})$ and then uses one of

$$\begin{aligned} \boldsymbol{g}_\mu(\boldsymbol{x};\boldsymbol{u}) &= \delta_{\mathrm{f}}(f;\boldsymbol{x};\boldsymbol{u};\mu)\boldsymbol{B}\boldsymbol{u}, \quad \text{or} \\ \hat{\boldsymbol{g}}_\mu(\boldsymbol{x};\boldsymbol{u}) &= \delta_{\mathrm{c}}(f;\boldsymbol{x};\boldsymbol{u};\mu)\boldsymbol{B}\boldsymbol{u}, \end{aligned} \tag{3.1}$$

for a difference parameter $\mu > 0$. Nesterov and Spokoiny also propose a third oracle, $\boldsymbol{g}_0(\boldsymbol{x};\boldsymbol{u}) = f'(\boldsymbol{x};\boldsymbol{u})\boldsymbol{B}\boldsymbol{u}$, intended for the optimization of non-smooth functions; this oracle assumes the ability to compute directional derivatives $f'(\boldsymbol{x};\boldsymbol{u})$. For this reason, Nesterov and Spokoiny refer to all oracles as gradient-free instead of derivative-free. Given the scope of this survey, we focus on the derivative-free oracles $\boldsymbol{g}_\mu$ and $\hat{\boldsymbol{g}}_\mu$ displayed in (3.1).

With an oracle $\boldsymbol{g}$ chosen as either oracle in (3.1), Nesterov random-search methods are straightforward to define, and we do so in Algorithm 6. In Algorithm 6, $\mathrm{proj}(\cdot;\boldsymbol{\Omega})$ denotes projection onto a domain $\boldsymbol{\Omega}$.

Algorithm 6: Nesterov random search

```
Choose initial point $\boldsymbol{x}_0 \in \boldsymbol{\Omega}$, sequence of step sizes $\{\alpha_k\}_{k=0}^{\infty}$, oracle $\boldsymbol{g}$
  from (3.1), smoothing parameter $\mu > 0$ and covariance matrix $\boldsymbol{B}$
for $k = 0, 1, 2, \ldots$ do
    $\hat{\boldsymbol{x}}_k \leftarrow \arg\min_{j \in \{0,1,\ldots,k\}} f(\boldsymbol{x}_j)$
    Generate $\boldsymbol{u}_k \in \mathcal{N}(\boldsymbol{0}, \boldsymbol{B}^{-1})$; compute $\boldsymbol{g}(\boldsymbol{x}_k; \boldsymbol{u}_k)$
    $\boldsymbol{x}_{k+1} \leftarrow \operatorname{proj}(\boldsymbol{x}_k - \alpha_k \boldsymbol{B}^{-1} \boldsymbol{g}(\boldsymbol{x}_k; \boldsymbol{u}_k), \boldsymbol{\Omega})$
```

A particularly striking result proved in Nesterov and Spokoiny (2017) was perhaps the first WCC result for an algorithm (Algorithm 6) in the case where $f \in \mathcal{LC}^0$ – that is, f may be both non-smooth and non-convex. Because of the randomized nature of iteratively sampling from a Gaussian distribution in Algorithm 6, complexity results are given as expectations. That is, letting $\boldsymbol{U}_k = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\}$ denote the random variables associated with the first k iterations of Algorithm 6, complexity results are stated in terms of expectations with respect to the filtration defined by these variables. A WCC is given as an upper bound on the number of f evaluations needed to attain the approximate ($\epsilon > 0$) optimality condition

$$\mathbb{E}_{\boldsymbol{U}_{k-1}}[\|\nabla f_{\check{\mu}}(\hat{\boldsymbol{x}}_k)\|] \leq \epsilon, \tag{3.2}$$

where $\hat{\boldsymbol{x}}_k = \arg\min_{j=0,1,\ldots,k-1} f(\boldsymbol{x}_j)$. By fixing a particular choice of $\check{\mu}$ (dependent on ϵ, n and Lipschitz constants), Nesterov and Spokoiny (2017) demonstrate that the number of f evaluations needed to attain (3.2) is in $O(\epsilon^{-3})$; see Table A.1. For $f \in \mathcal{LC}^1$ (but still non-convex), Nesterov and Spokoiny (2017) prove a WCC result of type (3.2) in $O(\epsilon^{-2})$ for the same method. WCC results of Algorithm 6 under a variety of stronger assumptions on the convexity and differentiability of f are also shown in Table A.1 and discussed in Section 4. We further note that some randomized methods of the form Algorithm 6 have also been developed for (STOCH), which we discuss in Section 6.

We remark on an undesirable feature of the convergence analysis for variants of Algorithm 6: the analysis of these methods supposes that the sequence $\{\alpha_k\}$ is chosen as a constant that depends on parameters, including L_{f}, that may not be available to the method. Similar assumptions concerning the preselection of $\{\alpha_k\}$ also appear in the convex cases discussed in Section 4, and we highlight these dependencies in Table A.1.

3.2. Randomized direct-search methods

Randomization has also been used in the DDS framework discussed in Section 2.1 in the hope of more efficiently using evaluations of f. Polling every point in a PSS requires at least $n+1$ function evaluations; if f is expensive to evaluate and n is relatively large, this can be wasteful. A deterministic strategy for performing fewer evaluations on many iterations is opportunistic polling. Work in randomized direct-search methods attempts to address, formalize and analyse the situation where polling directions are *randomly sampled* from some distribution in each iteration. The ultimate goal is to replace the $O(n)$ per-iteration function evaluation cost with an $O(1)$ per-iteration cost,[4] while still guaranteeing some form of global convergence.

In Section 2.1, we mentioned MADS methods that consider the random generation of polling directions in each iteration (in order to satisfy the asymptotic density required of search directions for the minimization of non-smooth, but Lipschitz-continuous, f). Examples include Audet and Dennis, Jr (2006) and Van Dyke and Asaki (2013), which implement LTMADS and QRMADS, respectively. While this direction of research is within the scope of randomized methods, the purpose of randomization in MADS methods is to overcome particular difficulties encountered when optimizing general non-smooth objectives. This particular randomization does not fall within the scope of this section, where randomization is intended to decrease a method's dependence on n. In the remainder of this section, we focus on a body of work that seems to exist entirely for the unconstrained case where f is assumed sufficiently smooth.

Gratton, Royer, Vicente and Zhang (2015) extend the direct-search framework (Algorithm 2) by assuming that the set of polling directions $\boldsymbol{D}_k$ includes only a descent direction with probability p (as opposed to assuming $\boldsymbol{D}_k$ always includes a descent direction, which comes for free when $f \in \mathcal{LC}^1$ provided $\boldsymbol{D}_k$ is, for example, a PSS). To formalize, given $p \in (0,1)$, a random sequence of polling directions $\{\boldsymbol{D}_k\}$ is said to be p-probabilistically κ_{d}-descent provided that, given a deterministic starting point $\boldsymbol{x}_0$,

$$\mathbb{P}[\mathrm{cm}([\boldsymbol{D}_0, -\nabla f(\boldsymbol{x}_0)]) \geq \kappa_{\mathrm{d}}] \geq p, \tag{3.3}$$

and for all $k \geq 1$,

$$\mathbb{P}[\mathrm{cm}([\boldsymbol{D}_k, -\nabla f(\boldsymbol{x}_k)]) \geq \kappa_{\mathrm{d}} \mid \boldsymbol{D}_0, \ldots, \boldsymbol{D}_{k-1}] \geq p, \tag{3.4}$$

where $\mathrm{cm}(\cdot)$ is the cosine measure in (2.8). A collection of polling directions $\boldsymbol{D}_k$ satisfying (3.3) and (3.4) can be obtained by drawing directions uniformly on the unit ball.

As with the other methods in this section, $\boldsymbol{x}_k$ in (3.4) is in fact a random variable due to the random sequence $\{\boldsymbol{D}_k\}$ generated by the algorithm, and hence it makes sense to view (3.4) as a probabilistic statement. In words, (3.4) states that with probability at least p, the set of polling directions

[4] We note that a $O(1)$ cost can naturally be achieved by deterministic DDS methods when derivatives are available (Abramson *et al.* 2004).

used in iteration k has a positive cosine measure with the steepest descent direction $-\nabla f(\boldsymbol{x}_k)$, regardless of the past history of the algorithm. Gratton *et al.* (2015) use Chernoff bounds in order to bound the worst-case complexity *with high probability*. Roughly, they show that if $f \in \mathcal{LC}^1$, and if (3.3) and (3.4) hold with $p > 1/2$ (this constant changing when $\gamma_{\text{dec}} \neq 1/\gamma_{\text{inc}}$), then $\|\nabla f(\boldsymbol{x}_k)\| \leq \epsilon$ holds within $O(\epsilon^{-2})$ function evaluations with a probability that increases exponentially to 1 as $\epsilon \to 0$. The WCC result of Gratton *et al.* (2015) demonstrates that as $p \to 1$ (*i.e.* $\boldsymbol{D}_k$ almost always includes a descent direction), the known WCC results for Algorithm 2 discussed in Section 2.1.2 are recovered. A more precise statement of this WCC result is included in Table A.1.

Bibi *et al.* (2019) propose a randomized direct-search method in which the two poll directions in each iteration are $\boldsymbol{D}_k = \{\boldsymbol{e}_i, -\boldsymbol{e}_i\}$, where $\boldsymbol{e}_i$ is the ith elementary basis vector. In the kth iteration, $\boldsymbol{e}_i$ is selected from $\{\boldsymbol{e}_1, \dots, \boldsymbol{e}_n\}$ with a probability proportional to the Lipschitz constant of the ith partial derivative of f. Bibi *et al.* (2019) perform WCC analysis of this method assuming a known upper bound on Lipschitz constants of partial derivatives; this assumption leads to improved constant factors, but they essentially prove an upper bound on the number of iterations needed to attain $\mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|] \leq \epsilon$ in $O(\epsilon^{-2})$, where the expectation is with respect to the random sequence of $\boldsymbol{D}_k$. Bibi *et al.* (2019) prove additional WCC results in cases where f is convex or c-strongly convex.

An early randomized DDS derivative-free method that only occasionally employs a descent direction is developed by Diniz-Ehrhardt *et al.* (2008). There, a non-monotone line-search strategy is used to accommodate search directions along which descent may not be initially apparent. Belitz and Bewley (2013) develop a randomized DDS method that employs surrogates and an adaptive lattice.

3.3. Randomized trust-region methods

Whereas the theoretical convergence of a DDS method depends on the set of polling directions satisfying some spanning property (*e.g.* a cosine measure bounded away from zero), the theoretical convergence of a trust-region method (*e.g.* Algorithm 3) depends on the use of fully linear models. Analogous to how randomized DDS methods relax the requirement of the use of a positive spanning set in *every* iteration, (3.4), it is reasonable to ask whether one can relax the requirement of being fully linear in every iteration of a trust-region method. Practically speaking, in the unconstrained case it may not be necessary to ensure that every model is built by using a Λ-poised set of points (therefore ensuring that the model is fully linear) on every iteration, since ensuring Λ-poised sets entails additional function evaluations.

Bandeira, Scheinberg and Vicente (2014) consider a sequence of random models $\{m_k\}$ and a random sequence of trust-region centres and radii $\{\boldsymbol{x}_k, \Delta_k\}$. They say that the sequence of random models is p-probabilistically $\boldsymbol{\kappa}$-fully linear provided

$$\mathbb{P}[m_k \text{ is a } \boldsymbol{\kappa}\text{-fully linear model of } f \text{ on } \mathcal{B}(\boldsymbol{x}_k; \Delta_k) \mid \mathcal{H}_{k-1}] \geq p, \tag{3.5}$$

where $\mathcal{H}_{k-1}$ is the filtration of the random process prior to the current iteration. That is, $\mathcal{H}_{k-1}$ is the σ-algebra generated by the algorithm's history. Under additional standard assumptions concerning Algorithm 3 (*e.g.* $\gamma_{\text{dec}} = 1/\gamma_{\text{inc}}$), the authors show that if (3.5) holds with $p > 1/2$, then $\lim_{k\to\infty} \|\nabla f(\boldsymbol{x}_k)\| = 0$ almost surely (*i.e.* with probability one). Gratton, Royer, Vicente and Zhang (2018) build on this result; they demonstrate that, up to constants, the same (with high probability) WCC bound that was proved for DDS methods in Gratton *et al.* (2015) holds for the randomized trust-region method proposed by Bandeira *et al.* (2014). Higher-order versions of (3.5) also exist; in Section 5.2 we discuss settings for which Bandeira *et al.* (2014) obtain probabilistically $\boldsymbol{\kappa}$-fully quadratic models by interpolating f on a set of fewer than $(n+1)(n+2)/2$ points.

4. Methods for convex objectives

As is true of derivative-based optimization, convexity in the objective of (DET) or (STOCH) can be exploited either when designing new methods or when analysing existing methods. Currently, this split falls neatly into two categories. The majority of work considering (DET) when f is convex sharpens the WCCs for frameworks already discussed in this survey. On the other hand, the influence of machine learning, particularly large-scale empirical risk minimization, has led to entirely new derivative-free methods for solving (STOCH) when f is convex.

4.1. Methods for deterministic convex optimization

We first turn our attention to the solution of (DET). In convex optimization, one can prove WCC bounds on the difference between a method's estimate of the global minimum of f and the value of f at a global minimizer $\boldsymbol{x}_* \in \boldsymbol{\Omega}$ (*i.e.* a point satisfying $f(\boldsymbol{x}_*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \boldsymbol{\Omega}$). This differs from the local WCC bounds on the objective gradient, namely (2.4), that are commonly shown when f is not assumed to be convex. For convex f, under appropriate additional assumptions, one typically demonstrates that a method satisfies

$$\lim_{k\to\infty} f(\boldsymbol{x}_k) - f(\boldsymbol{x}_*) = 0, \tag{4.1}$$

where $\boldsymbol{x}_k$ is the kth point of the method. Hence, an appropriate measure of ϵ-optimality when minimizing an unconstrained convex objective is the satisfaction of

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_*) \leq \epsilon. \tag{4.2}$$

For the DDS methods discussed in Section 2.1.2, Dodangeh and Vicente (2016) and Dodangeh, Vicente and Zhang (2016) analyse the worst-case complexity of Algorithm 2 when there is no search step and when f is smooth and convex, and has a bounded level set. By imposing an appropriate upper bound on the step sizes α_k, and (for $c > 0$) using a test of the form

$$f(\boldsymbol{p}_i) \leq f(\boldsymbol{x}_k) - c\alpha_k^2, \tag{4.3}$$

in line 4 of Algorithm 2, Dodangeh and Vicente (2016) show that the worst-case number of f evaluations to achieve (2.2) is in $O(n^2 L_{\mathrm{g}}^2 \epsilon^{-1})$. Dodangeh *et al.* (2016) show that this n^2-dependence is optimal (in the sense that it cannot be improved) within the class of deterministic methods that employ positive spanning sets. Recall from Section 3.2 that randomized methods allow one to reduce this dependence to be linear in n. Under additional assumptions, which are satisfied, for example, when f is strongly convex, Dodangeh and Vicente (2016) prove R-linear convergence of Algorithm 2, yielding a WCC of type (4.2) with the dependence on ϵ reduced to $\log(\epsilon^{-1})$. We note that, in the convex setting, R-linear convergence had been previously established for a DDS method by Dolan *et al.* (2003). It is notable that the method analysed in Dolan *et al.* (2003) requires only strict decrease, whereas, to the authors' knowledge, the DDS methods for which WCC results have been established all require sufficient decrease.

Konečný and Richtárik (2014) propose a DDS method that does not allow for increases in the step size α_k and analyse the method on strongly convex, convex and non-convex objectives. Although Konečný and Richtárik (2014) demonstrate WCC bounds with the same dependence on n and ϵ as do Dodangeh and Vicente (2016), they additionally assume that one has explicit knowledge of a Lipschitz gradient constant L_{g} and can thus replace the test (4.3) explicitly with

$$f(\boldsymbol{p}_i) \leq f(\boldsymbol{x}_k) - \frac{L_{\mathrm{g}}}{2}\alpha_k. \tag{4.4}$$

Exploiting this additional knowledge of L_{g}, the WCC result in Konečný and Richtárik (2014) exhibits a strictly better dependence on L_{g} than does the WCC result in Dodangeh and Vicente (2016), with a WCC of type (4.2) in $O(n^2 L_{\mathrm{g}} \epsilon^{-1})$. Additionally assuming f is c-strongly convex, Konečný and Richtárik (2014) provide a result showing a WCC of type (4.2) in $O(\log(\epsilon^{-1}))$.

In the non-convex case, Konečný and Richtárik (2014) recover the same WCC of type (2.2) from Vicente (2013); see the discussion in Section 2.1. Once again, however, the result by Konečný and Richtárik (2014) assumes knowledge of L_{g} and again recovers a strictly better dependence on L_{g} by using the test (4.4) in line 4 of Algorithm 2.

Recalling the Nesterov random search methods discussed in Section 3.1.2, we remark that Nesterov and Spokoiny (2017) explicitly give results for deterministic, convex f. In particular, Nesterov and Spokoiny prove WCCs of a specific type. Because of the randomized nature of Algorithm 6, WCCs are given as expectations of the form

$$\mathbb{E}_{\boldsymbol{U}_{k-1}}[f(\hat{\boldsymbol{x}}_k)] - f(\boldsymbol{x}_*) \leq \epsilon, \tag{4.5}$$

where $\hat{\boldsymbol{x}}_k = \arg\min_{j\in\{0,1,\dots,k-1\}} f(\boldsymbol{x}_j)$ and where $\boldsymbol{U}_{k-1} = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \dots, \boldsymbol{u}_{k-1}\}$ is the filtration of Gaussian samples. The form of ϵ-optimality represented by (4.5) can be interpreted as a probabilistic variant of (4.2). The WCC results of Nesterov and Spokoiny show that the worst-case number of f evaluations to achieve (4.5) is in $O(\epsilon^{-1})$ when $f \in \mathcal{LC}^1$. Additionally assuming that f is c-strongly convex yields an improved result; the WCC of type (4.5) is now in $O(\log(\epsilon^{-1}))$. Moreover, by mimicking the method of accelerated gradient descent (see *e.g.* Nesterov 2004, Chapter 2.2), Nesterov and Spokoiny present a variant of Algorithm 6 with a WCC of type (4.5) in $O(\epsilon^{-1/2})$. When $f \in \mathcal{LC}^0$, Nesterov and Spokoiny provide a WCC of type (4.5) in $O(\epsilon^{-2})$, but this result assumes that Algorithm 6 uses an oracle with access to exact directional derivatives of f. Thus, the method achieves the $O(\epsilon^{-2})$ result when $f \in \mathcal{LC}^0$ is not a derivative-free method.

As remarked in Section 3, these convergence results depend on preselecting a sequence of step sizes $\{\alpha_k\}$ for Algorithm 6; in the convex case, the necessary $\{\alpha_k\}$ depends not only on the Lipschitz constants but also on a bound $R_{\boldsymbol{x}}$ on the distance between the initial point and the global minimizer (*i.e.* $\|\boldsymbol{x}_0 - \boldsymbol{x}_*\| \leq R_{\boldsymbol{x}}$). The aforementioned WCC results will hold only if one chooses $\{\alpha_k\}$ and μ (the difference parameter of the oracle used in Algorithm 6) that scale with L_{g} and $R_{\boldsymbol{x}}$ appropriately. When additionally assuming f is c-strongly convex, $\{\alpha_k\}$ and μ also depend on c. Stich, Müller and Gärtner (2013)[5] extend the framework of Algorithm 6 with an approximate line search that avoids the need for predetermined sequences of step sizes.

In general, the existing WCC results for derivative-free methods match the WCC results for their derivative-based counterparts in ϵ-dependence. The WCC results for derivative-free methods tend to involve an additional factor of n when compared with their derivative-based counterparts. This observation mirrors a common intuition in derivative-free optimization: since a number of f evaluations in $O(n)$ can guarantee a suitable approximation

[5] A careful reader may be caught off guard by the fact that Stich *et al.* (2013) was published before Nesterov and Spokoiny (2017). This is not a typo; Stich *et al.* (2013) build on the results from an early preprint of Nesterov and Spokoiny (2017).

of the gradient, then for any class of gradient-based method for which we replace gradients with approximate gradients or model gradients, one should expect to recover the WCC of that method, but with an extra factor of n. WCC results such as those discussed thus far in this survey add credence to this intuition. As we will see in Section 4.2, however, this optimistic trend does not always hold: we will see problem classes for which derivative-free methods are provably worse than their derivative-based counterparts by a factor of ϵ^{-1}.

Bauschke, Hare and Moursi (2014) offer an alternative approach to Algorithm 6 for the solution of (DET) when f is assumed convex and, additionally, lower-$\mathcal{C}^2$. (Such an assumption on f is obviously stronger than plain convexity but contains, for example, functions that are defined as the pointwise maximum over a collection of convex functions.) Bauschke *et al.* (2014) show that linear interpolation through function values is sufficient for obtaining approximate subgradients of convex, lower-$\mathcal{C}^2$ function; these approximate subgradients are used in lieu of subgradients in a mirror-descent algorithm (see *e.g.* Srebro, Sridharan and Tewari 2011) similar to Algorithm 7 in Section 4.2.2. Bauschke *et al.* (2014) establish convergence of their method in the sense of (4.1), and they demonstrate the performance of their method when applied to pointwise maxima of collections of convex quadratics.

4.2. Methods for convex stochastic optimization

We now turn our attention to the solution of the problem (STOCH) when $\tilde{f}(\boldsymbol{x};\boldsymbol{\xi})$ is assumed convex in $\boldsymbol{x}$ for each realization $\boldsymbol{\xi}$. Up to this point in the survey, we have typically assumed that (STOCH) is unconstrained, that is, $\boldsymbol{\Omega} = \mathbb{R}^n$. In this section, however, it will become more frequent that $\boldsymbol{\Omega}$ is a compact set.

In the machine learning community, zeroth-order information (*i.e.* evaluations of $\tilde{f}$ only) is frequently referred to as *bandit feedback*,[6] due to the concept of multi-armed bandits from reinforcement learning. Multi-armed bandit problems are sequential allocation problems defined by a prescribed set of actions. Robbins (1952) formulates a multi-armed bandit problem as a gambler's desire to minimize the total losses accumulated from pulling discrete sequence (of length T) of $A < \infty$ slot machine arms. The gambler does not have to decide the full length-T sequence up front. Rather, at time $k \leq T$, the losses associated with the first $k-1$ pulls are known to the gambler when deciding which of the A arms to pull next. The gambler's decision of the kth arm to pull is represented by the scalar variable $x_k \in \boldsymbol{\Omega}$. Given

[6] A 'one-armed bandit' is an American colloquialism for a casino slot machine, the arm of which must be pulled to reveal a player's losses or rewards.

additional environmental variables outside the gambler's control, $\boldsymbol{\xi}_k \in \Xi$, which represents the stochastic nature of the slot machine,[7] the environment makes a decision $\boldsymbol{\xi}$ simultaneously with the gambler's decision x. The gambler's loss is then $\tilde{f}(x;\boldsymbol{\xi})$.

Within this multi-armed bandit setting, the typical metric of the gambler's performance is the *cumulative regret*. Provided that the expectation $\mathbb{E}_{\boldsymbol{\xi}}[\tilde{f}(x;\boldsymbol{\xi})] = f(x)$ exists for each $x \in \{1,\dots,A\}$, then the gambler's best long-run strategy in terms of minimizing the expected total losses is to constantly play $x_* \in \arg\min_{x\in\{1,\dots,A\}} f(x)$. If, over the course of T pulls, the gambler makes a sequence of decisions $x_1,\dots,x_T$ and the environment makes a sequence of decisions $\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_T$ resulting in a sequence of realized losses $\tilde{f}(x_1;\boldsymbol{\xi}_1),\dots,\tilde{f}(x_T;\boldsymbol{\xi}_T)$, then the cumulative regret r_T associated with the gambler's sequence of decisions is the difference between the cumulative loss incurred by the gambler's strategy $(x_1,\dots,x_T)$ and the loss incurred by the best possible long-run strategy $(x_*,\dots,x_*)$. Analysis of methods for bandit problems in this set-up is generally concerned with *expected cumulative regret*

$$\mathbb{E}_{\boldsymbol{\xi}}[r_T(x_1,\dots,x_T)] = \mathbb{E}_{\boldsymbol{\xi}}\left[\sum_{k=1}^{T}\tilde{f}(x_k;\boldsymbol{\xi}_k)\right] - Tf(x_*), \tag{4.6}$$

where the expectation is computed over $\boldsymbol{\xi} \in \Xi$, since we assume here that the sequence of $\boldsymbol{\xi}_k$ is independent and identically distributed.

This particular treatment of the bandit problem with a discrete space of actions $x \in \{1,\dots,A\}$ was the one considered by Robbins (1952) and has been given extensive treatment (Auer, Cesa-Bianchi, Freund and Schapire 2003, Lai and Robbins 1985, Agrawal 1995, Auer, Cesa-Bianchi and Fischer 2002).

Extending multi-armed bandit methods to *infinite-armed* bandits makes the connections to derivative-free optimization – particularly derivative-free convex optimization – readily apparent. Auer (2002) extends the multi-armed bandit problem to allow for a compact (as opposed to discrete) set of actions for the gambler $\boldsymbol{x} \in \boldsymbol{\Omega} \subset \mathbb{R}^n$ as well as a compact set of vectors for the environment, $\boldsymbol{\xi} \in \Xi \subset \mathbb{R}^n$. The vectors $\boldsymbol{\xi}$ in this set-up define linear functions; that is, if the gambler chooses $\boldsymbol{x}_k$ in their kth pull, and the environment chooses $\boldsymbol{\xi}_k \in \Xi$, then the gambler incurs loss $\tilde{f}(\boldsymbol{x}_k;\boldsymbol{\xi}_k) = \boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{x}_k$. In this linear regime, expected regret takes a form remarkably similar to

[7] Depending on the problem set-up, the environment of a bandit problem may be either stochastic or adversarial. Because this section is discussing stochastic convex optimization, we will assume that losses are stochastic; that is, the $\boldsymbol{\xi}_k$ are i.i.d. and independent of the gambler's decisions x_k. See Bubeck and Cesa-Bianchi (2012) for a survey of bandit problems more general than those discussed in this survey.

(4.6), that is,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\xi}}[r_T(\boldsymbol{x}_1,\dots,\boldsymbol{x}_T)] &= \mathbb{E}_{\boldsymbol{\xi}}\left[\sum_{k=1}^{T}\tilde{f}(\boldsymbol{x}_k;\boldsymbol{\xi}_k)\right] - \min_{\boldsymbol{x}\in\boldsymbol{\Omega}}\mathbb{E}_{\boldsymbol{\xi}}\left[\sum_{k=1}^{T}\tilde{f}(\boldsymbol{x}_k;\boldsymbol{\xi}_k)\right]\\ &= \left(\sum_{k=1}^{T}f(\boldsymbol{x}_k)\right) - Tf(\boldsymbol{x}_*),\end{aligned}\tag{4.7}$$

with $\boldsymbol{x}_* \in \arg\min_{\boldsymbol{x}\in\boldsymbol{\Omega}} f(\boldsymbol{x})$. As in (4.6), (4.7) defines the expected regret with respect to the best long-run strategy $\boldsymbol{x}\in\boldsymbol{\Omega}$ that the gambler could have played for the T rounds (*i.e.* the strategy that would minimize the expected cumulative losses).[8]

By using a bandit method known as Thompson sampling, one can show (under appropriate additional assumptions) that if $\tilde{f}(\boldsymbol{x};\boldsymbol{\xi}) = \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{x}$, then (4.7) can be bounded as

$$\mathbb{E}_{\boldsymbol{\xi}}[r_T(\boldsymbol{x}_1,\dots,\boldsymbol{x}_T)] \in O(n\sqrt{T\log(T)})$$

(Russo and Van Roy 2016). Analysis of bounds on (4.7) in the linear case raises an interesting question: to what extent can similar analysis be performed for classes of functions $\tilde{f}(\boldsymbol{x};\boldsymbol{\xi})$ that are non-linear in $\boldsymbol{x}$?

In much of the bandit literature, the additional structure defining a class of non-linear functions is convexity; here, by convexity, we mean that $\tilde{f}(\boldsymbol{x};\boldsymbol{\xi})$ is convex in $\boldsymbol{x}$ for each realization $\boldsymbol{\xi}\in\Xi$.

Regret bounds on (4.7) automatically imply WCC results for stochastic convex optimization. To see this, define an *average point*

$$\bar{\boldsymbol{x}}_k = \frac{1}{k}\sum_{t=1}^{k}\boldsymbol{x}_t.$$

Because of the convexity of f,

$$f(\bar{\boldsymbol{x}}_T) - f(\boldsymbol{x}_*) \le \frac{1}{T}\sum_{k=1}^{T}f(\boldsymbol{x}_k) - f(\boldsymbol{x}_*) = \frac{\mathbb{E}_{\boldsymbol{\xi}}[r_T(\boldsymbol{x}_1,\dots,\boldsymbol{x}_T)]}{T},\tag{4.8}$$

where the inequality follows from an application of Jensen's inequality. We see in (4.8) that, given an upper bound of $\bar{r}(T)$ on the expected regret $\mathbb{E}_{\boldsymbol{\xi}}[r_T]$, we can automatically derive, provided $\bar{r}(T)/T\le\epsilon$, a WCC result for stochastic convex optimization of the form

$$\mathbb{E}_{\boldsymbol{\xi}}[\tilde{f}(\bar{\boldsymbol{x}}_T;\boldsymbol{\xi}) - \tilde{f}(\boldsymbol{x}_*;\boldsymbol{\xi})] = f(\bar{\boldsymbol{x}}_T) - f(\boldsymbol{x}_*) \le \epsilon.\tag{4.9}$$

[8] We remark that many of the references we provide here may also refer to methods minimizing (4.7) as methods of *online optimization*; one reference in particular (Bach and Perchet 2016) even suggests a taxonomy identifying bandit learning as a restrictive case of online optimization.

Equation (4.9) corresponds to a stochastic version of a WCC result of type (4.2) where $N_\epsilon = T$.

The converse implication, however, is not generally true: a small optimization error does not imply a small regret. That is, WCC results of type (4.9) do not generally imply bounds on expected regret (4.7). This is particularly highlighted by Shamir (2013), who considers a class of strongly convex quadratic objectives f. For such objectives, Shamir (2013) establishes a strict gap between the upper bound on the optimization error (the left-hand side of (4.9)) and the lower bound on the expected regret (4.7) attainable by any method in the bandit setting. Such a gap, between expected regret and optimization error, has also been proved for bandit methods applied to problems of logistic regression (Hazan, Koren and Levy 2014).

Results such as those of Shamir (2013), Jamieson, Nowak and Recht (2012) and Hazan *et al.* (2014) have led researchers to consider methods of derivative-free (stochastic) optimization within the paradigm of convex bandit learning. In this survey, we group these methods into two categories of assumptions on the type of bandit feedback (*i.e.* the observed realizations of $\tilde{f}$) available to the method: one-point bandit feedback or two-point (multi-point) bandit feedback. Although many of the works we cite have also analysed regret bounds for these methods, we focus on optimization error.

4.2.1. One-point bandit feedback

In one-point bandit feedback, a method is assumed to have access to an oracle $\tilde{f}$ that returns unbiased estimates of f. In particular, given two points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{\Omega}$, two separate calls to the oracle will return $\tilde{f}(\boldsymbol{x}_1; \boldsymbol{\xi}_1)$ and $\tilde{f}(\boldsymbol{x}_2; \boldsymbol{\xi}_2)$; methods do not have control over the selection of the random variables $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$. Many one-point bandit methods do not fall neatly into the frameworks discussed in this survey (Agarwal *et al.* 2011, Belloni, Liang, Narayanan and Rakhlin 2015, Bubeck, Lee and Eldan 2017). See Table 4.1 for a summary of best known WCC results of type (4.9) for one-point bandit feedback methods.

One example of a method using one-point bandit feedback, whose development falls naturally into our discussion thus far, is given by Flaxman, Kalai and McMahan (2005); they analyse a method resembling Algorithm 6, but the gradient-free oracle is chosen as

$$\boldsymbol{g}_\mu(\boldsymbol{x}; \boldsymbol{u}; \boldsymbol{\xi}) = \frac{\tilde{f}(\boldsymbol{x} + \mu\boldsymbol{u}; \boldsymbol{\xi})}{\mu}\boldsymbol{u}, \tag{4.10}$$

where $\boldsymbol{u}$ is drawn uniformly from the unit n-dimensional sphere. That is, given a realization $\boldsymbol{\xi} \in \boldsymbol{\Xi}$, a stochastic gradient estimator based on a stochastic evaluation at the point $\boldsymbol{x} + \mu\boldsymbol{u}$ is computed via (4.10). For this method, and for general convex f, Flaxman *et al.* (2005) demonstrate a

Table 4.1. Best known WCCs for N_ϵ, the number of evaluations required to bound (4.9), for one-point bandit feedback. N_ϵ is given only in terms of n and ϵ. See text for the definition of β-smooth; here $\beta > 2$. Method types include random search (RS), mirror descent (MD) and ellipsoidal.

Assumption on f	N_ϵ	Method type (citation)
convex, $f \in \mathcal{LC}^0$	$n^2\epsilon^{-4}$ $n^{13/2}\epsilon^{-2}$	RS (Flaxman *et al.* 2005) ellipsoidal (Belloni *et al.* 2015)
c-strongly convex, $f \in \mathcal{LC}^0$	$n^2\epsilon^{-3}$	RS (Flaxman *et al.* 2005)
convex, $f \in \mathcal{LC}^1$	$n\epsilon^{-3}$ $n^{13/2}\epsilon^{-2}$	MD (Gasnikov *et al.* 2017) ellipsoidal (Belloni *et al.* 2015)
c-strongly convex, $f \in \mathcal{LC}^1$	$n^2\epsilon^{-2}$	MD (Gasnikov *et al.* 2017)
convex, β-smooth	$n^2\epsilon^{-2\beta/(\beta-1)}$	RS (Bach and Perchet 2016)
c-strongly convex, β-smooth	$n^2\epsilon^{-(\beta+1)/(\beta-1)}$	RS (Bach and Perchet 2016)

WCC bound of type (4.9) in $O(n^2\epsilon^{-4})$ for general convex f. For strongly convex f, Flaxman *et al.* (2005) demonstrate a WCC bound of type (4.9) in $O(n^2\epsilon^{-3})$. Various extensions of these results are given in Saha and Tewari (2011), Dekel, Eldan and Koren (2015) and Gasnikov *et al.* (2017). To the best of our knowledge, the best known upper bound on the WCC for smooth, strongly convex problems is in $O(n^2\epsilon^{-2})$ and the best known upper bound on the WCC for smooth, convex problems is in $O(n\epsilon^{-3})$ (Gasnikov *et al.* 2017).

For the solution of (STOCH) when $\boldsymbol{\Omega} = \mathbb{R}^n$, Bach and Perchet (2016) analyse a method resembling Algorithm 6 wherein the gradient-free oracle is chosen as

$$\boldsymbol{g}_\mu(\boldsymbol{x};\boldsymbol{u};\boldsymbol{\xi}_+,\boldsymbol{\xi}_-) = \frac{\tilde{f}(\boldsymbol{x}+\mu\boldsymbol{u};\boldsymbol{\xi}_+) - \tilde{f}(\boldsymbol{x}-\mu\boldsymbol{u};\boldsymbol{\xi}_-)}{\mu}\boldsymbol{u}, \tag{4.11}$$

where $\boldsymbol{u}$ is again drawn uniformly from the unit n-dimensional sphere. We remark that $\boldsymbol{\xi}_+, \boldsymbol{\xi}_- \in \boldsymbol{\Xi}$ in (4.11) are *different* realizations; this distinction will become particularly relevant in Section 4.2.2. For the solution of (STOCH) when $\boldsymbol{\Omega} \subsetneq \mathbb{R}^n$, Bach and Perchet (2016) also consider a gradient-free oracle like (4.10). Bach and Perchet (2016) are particularly interested in how the smoothness of f can be exploited to improve complexity bounds in the one-point bandit feedback paradigm; they define a parameter β and say that a function f is β-smooth provided f is almost everywhere $(\beta-1)$-times Lipschitz-continuously differentiable (a strictly weaker condition than assuming $f \in \mathcal{LC}^{\beta-1}$). When $\beta = 2$, Bach and Perchet (2016) recover similar results to those seen in Table 4.1 for when $f \in \mathcal{LC}^1$. When $\beta > 2$, however, in both the constrained and unconstrained case, Bach and Perchet (2016) prove a WCC result of type (4.9) in $O(n^2\epsilon^{2\beta/(1-\beta)})$ when f is convex and β-smooth. Further assuming that f is c-strongly convex, Bach and Perchet (2016) prove a WCC result of type (4.9) in $O(n^2\epsilon^{(\beta+1)/(1-\beta)}/c)$. Notice that asymptotically, as $\beta \to \infty$, this bound is in $O(n^2\epsilon^{-1}/c)$. This asymptotic result is particularly important because it attains the lower bound on optimization error demonstrated by Shamir (2013) for strongly convex ∞-smooth (quadratic) f.

We also note that Bubeck *et al.* (2017) conjecture that a particular kernel method can achieve a WCC of type (4.9) in $O(n^3\epsilon^{-2})$ for general convex functions. In light of well-known results in deterministic convex optimization, the WCCs summarized in Table 4.1 may be surprising. In particular, for any c-strongly convex function $f \in \mathcal{LC}^1$, the best known WCC results are in $O(\epsilon^{-2})$. We place particular emphasis on this result because it illustrates a gap between derivative-free and derivative-based optimization that is not just a factor of n. In this particular one-point bandit feedback setting, there do not seem to exist methods that achieve the optimal[9] $O(\epsilon^{-1})$ convergence rate attainable by gradient-based methods for smooth strongly convex stochastic optimization. Hu, Prashanth, György and Szepesvári (2016) partially address this issue concerning one-point bandit feedback, which they refer to as 'uncontrolled noise'. These observations motivated the study of two-point (multi-point) bandit feedback, which we will discuss in the next section, Section 4.2.2.

We further remark that every WCC in Table 4.1 has a polynomial dependence on the dimension n, raising natural questions about the applicability of these methods in high-dimensional settings. Wang, Du, Balakrishnan and Singh (2018) consider a mirror descent method employing a special gradient-free oracle computed via a compressed sensing technique. They prove a WCC of type (4.9) in $O(\log(d)^{3/2}n_z^2\epsilon^{-3})$, under additional assumptions on derivative sparsity, most importantly, that for all $\boldsymbol{x} \in \boldsymbol{\Omega}$, $\|\nabla f(\boldsymbol{x})\|_0 \leq n_z$ for some n_z. Thus, provided $n_z \ll n$, the polynomial dependence on n becomes a logarithmic dependence on n, at the expense of a WCC with a strictly worse dependence on ϵ than ϵ^{-2}. In Section 5.2, we discuss methods that similarly exploit known sparsity of objective function derivatives.

[9] Optimal here is meant in a minimax information-theoretic sense (Agarwal, Wainwright, Bartlett and Ravikumar 2009).

In an extension of Algorithm 6, Chen (2015, Chapter 3) dynamically updates the difference parameter by exploiting knowledge of the changing variance of $\boldsymbol{\xi}$.

4.2.2. Two-point (multi-point) bandit feedback

We now focus on the stochastic paradigm of two-point (or multi-point) bandit feedback. In this setting of bandit feedback, we do not encounter the same gaps in WCC results between derivative-free and derivative-based optimization that exist for one-point bandit feedback.

The majority of methods analysed in the two-point bandit feedback setting are essentially random-search methods of the form Algorithm 6. The gradient-free oracles from (3.1) in the two-point setting takes one of the two forms

$$\begin{aligned} \boldsymbol{g}_{\mu_k}(\boldsymbol{x};\boldsymbol{u};\boldsymbol{\xi}) &= \delta_{\mathrm{f}}(\tilde{f}(\cdot,\boldsymbol{\xi});\boldsymbol{x};\boldsymbol{u};\mu_k)\boldsymbol{B}\boldsymbol{u}, \quad \text{or} \\ \hat{\boldsymbol{g}}_{\mu_k}(\boldsymbol{x};\boldsymbol{u};\boldsymbol{\xi}) &= \delta_{\mathrm{c}}(\tilde{f}(\cdot,\boldsymbol{\xi});\boldsymbol{x};\boldsymbol{u};\mu_k)\boldsymbol{B}\boldsymbol{u}. \end{aligned} \tag{4.12}$$

The key observation in (4.12) is that $\boldsymbol{\xi}$ denotes a single realization used in the computation of both function values in the definitions of the oracles (see (2.28) and (2.29)). This assumption of 'controllable realizations' separates two-point bandit feedback from the more pessimistic one-point bandit feedback. This property of being able to recall a single realization $\boldsymbol{\xi}$ for two (or more) evaluations of $\tilde{f}$ is precisely why this setting of bandit feedback is called 'two-point' (or 'multi-point'). This property will also be exploited in Section 6.1.

The early work of Agarwal, Dekel and Xiao (2010) directly addresses the discussed gap in WCC results and demonstrates that a random-search method resembling Algorithm 6, but applied in the two-point (or multi-point) setting as opposed to the one-point setting, attains a WCC of type (4.9) in $O(\epsilon^{-1})$; this is a complexity result matching the optimal rate (in terms of ϵ-dependence) shown by Agarwal *et al.* (2009). See Table 4.2 for a summary of best known WCC results of type (4.9) for two-point bandit feedback methods.

Nesterov and Spokoiny (2017) provide a WCC result for Algorithm 6 using the stochastic gradient-free oracles (4.12), but strictly better WCCs have since been established. We also note that, in contrast with the gradient-free oracles of (3.1) in Section 3.1.2, the difference parameter μ is written as μ_k in (4.12), indicating that a selection for μ_k must be made in the kth iteration. In the works that we discuss here, μ_k in (4.12) is either chosen as a constant sufficiently small or else $\mu_k \to 0$ at a rate typically of the order of $1/k$. We also remark that many of the results discussed in this section trivially hold for deterministic convex problems, and can be seen as an extension of results concerning methods of the form of Algorithm 6.

Algorithm 7: Mirror-descent method with two-point gradient estimate

```
Choose initial point $\boldsymbol{x}_0$, sequence of step sizes $\{\alpha_k\}$, sequence of
  difference parameters $\{\mu_k\}$, and distribution of $\boldsymbol{\xi}$
for $k = 1, 2, \ldots, T-1$ do
    Sample $\boldsymbol{u}_k$ uniformly from the unit sphere $\mathcal{B}(\mathbf{0}; 1)$
    Sample a realization $\boldsymbol{\xi}_k$
    $\boldsymbol{g}_k \leftarrow \boldsymbol{g}_{\mu_k}(\boldsymbol{x}_k; \boldsymbol{u}_k; \boldsymbol{\xi}_k)$ using an oracle from (4.12)
    $\boldsymbol{x}_{k+1} \leftarrow \arg\min_{\boldsymbol{y}\in\boldsymbol{\Omega}} \boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{y} + \frac{1}{\alpha_k} D_\psi(\boldsymbol{y}, \boldsymbol{x}_k)$
```

Provided $f \in \mathcal{LC}^0$, the best known WCCs of type (4.9) (with variations in constants) can be found in Duchi, Jordan, Wainwright and Wibisono (2015), Gasnikov *et al.* (2017) and Shamir (2017). These works consider variants of mirror-descent methods with approximate gradients given by estimators of the form (4.12); see Algorithm 7 for a description of a basic mirror-descent method. Algorithm 7 depends on the concept of a *Bregman divergence* D_ψ, used in line 6 of Algorithm 7 to define a proximal-point subproblem. The Bregman divergence is defined by a function $\psi : \boldsymbol{\Omega} \to \mathbb{R}$, which is assumed to be 1-strongly convex with respect to the norm $\|\cdot\|_p$. To summarize many findings in this area (Duchi *et al.* 2015, Gasnikov *et al.* 2017, Shamir 2017), if $\|\cdot\|_q$ is the dual norm to $\|\cdot\|_p$ (*i.e.* $p^{-1} + q^{-1} = 1$) where $p \in \{1, 2\}$ and R_p denotes the radius of the feasible set in the $\|\cdot\|_p$-norm, then a bound on WCC of type (4.9) in $O(n^{2/q}R_p^2\epsilon^{-2})$ can be established for a method like Algorithm 7 in the two-point feedback setting where $f \in \mathcal{LC}^0$. These WCCs for methods like Algorithm 7 are responsible for the popularity of mirror-descent methods in machine learning. For many machine learning problems, solutions are typically sparse, and so, in some sense, $R_1 \leq R_2$. Thus, using a function ψ that is 1-strongly convex with respect to the $\|\cdot\|_1$-norm (*e.g.* simply letting $\psi = \|\cdot\|_1$) may be preferable to $p = 2$, in both theory and practice.

Duchi *et al.* (2015) also provide an information-theoretic lower bound on convergence rates for any method in the (non-strongly) convex, $f \in \mathcal{LC}^0$, two-point feedback setting. This bound matches the best known WCCs up to constants, demonstrating that these results are tight. This lower bound is still of the order of ϵ^{-2}, matching the result of Agarwal *et al.* (2009) in ϵ-dependence in the case where $f \in \mathcal{LC}^0$. It is also remarkable that this result is only a factor of $\sqrt{n}$ worse than the bounds provided by Agarwal *et al.* (2009) for the derivative-based case, as opposed to the factor of n that one may expect.

Additionally assuming $f(\boldsymbol{x})$ is strongly convex (but still assuming $f \in \mathcal{LC}^0$), Agarwal *et al.* (2010) prove, for a method like Algorithm 6, a WCC

Table 4.2. Best known WCCs of type (4.9) for two-point bandit feedback. N_ϵ is given only in terms of n and ϵ. See text for the definition of p, q. R_p denotes the size of the feasible set in the $\|\cdot\|_p$-norm. If f is c_p-strongly convex, then f is strongly convex with respect to the $\|\cdot\|_p$-norm with constant c_p. σ is the standard deviation on the gradient estimator $\boldsymbol{g}_\mu(\boldsymbol{x};\boldsymbol{u};\boldsymbol{\xi})$ (*i.e.* $\mathbb{E}_{\boldsymbol{\xi}}[\|\boldsymbol{g}_\mu(\boldsymbol{x};\boldsymbol{u};\boldsymbol{\xi})-\nabla f(\boldsymbol{x})\|^2] \le \sigma^2$). The Lipschitz constant of the gradient L_{g} is defined by the $\|\cdot\|_2$-norm. $\star$ denotes the additional assumption that $\mathbb{E}_{\boldsymbol{\xi}}[\|\boldsymbol{g}_\mu(\boldsymbol{x};\boldsymbol{u};\boldsymbol{\xi})\|] < \infty$. Method types include random search (RS), mirror descent (MD) and accelerated mirror descent (AMD).

Assumption on f	N_ϵ	Method type (citation)
convex	$n^{2/q}R_p\epsilon^{-2}$	MD (Duchi *et al.* 2015, Gasnikov *et al.* 2017, Shamir 2017)
c_p-strongly convex	$n^{2/q}c_p^{-1}\epsilon^{-1}$	MD (Gasnikov *et al.* 2017)
convex, smooth	$\max\left\{\dfrac{nL_{\mathrm{g}}R_2}{\epsilon}, \dfrac{n\sigma^2}{\epsilon^2}\right\}$	RS (Ghadimi and Lan 2013)
	$\max\left\{\dfrac{n^{2/q}L_{\mathrm{g}}R_p^2}{\epsilon}, \dfrac{n^{2/q}\sigma^2R_p^2}{\epsilon^2}\right\}$	MD (Dvurechensky, Gasnikov and Gorbunov 2018)
	$\max\left\{n^{1/2+1/q}\sqrt{\dfrac{L_{\mathrm{g}}R_p^2}{\epsilon}}, \dfrac{n^{2/q}\sigma^2R_p^2}{\epsilon^2}\right\}$	AMD (Dvurechensky *et al.* 2018)
convex, smooth, $\star$	$n^{2/q}R_p\epsilon^{-2}$	MD (Duchi *et al.* 2015)
c_p-strongly conv., smooth, $\star$	$n^{2/q}c_p^{-1}\epsilon^{-1}$	MD (Gasnikov *et al.* 2017)

in $O(n^2\epsilon^{-1})$. Using a method like Algorithm 7, Gasnikov *et al.* (2017) improve the dependence on n in this WCC to $O(n^{2/q}c_p^{-1}\epsilon^{-1})$, provided f is c_p-strongly convex with respect to $\|\cdot\|_p$.

We now address the case where f is convex and $f \in \mathcal{LC}^1$. Given an assumption that $\|\boldsymbol{g}_\mu(\boldsymbol{x};\boldsymbol{u};\boldsymbol{\xi})\|$ is uniformly bounded, Agarwal *et al.* (2010) demonstrate a WCC of type (4.9) in $O(n^2\epsilon^{-1})$. Dropping this somewhat restrictive assumption on the gradient-free oracle and assuming instead that the oracle used in a method like Algorithm 6 has bounded variance (*i.e.* the oracle satisfies $\mathbb{E}_{\boldsymbol{\xi}}[\|\boldsymbol{g}_\mu(\boldsymbol{x};\boldsymbol{u};\boldsymbol{\xi}) - \nabla f(\boldsymbol{x})\|^2] \le \sigma^2$), Ghadimi and Lan (2013) prove a WCC of a type similar to (4.9) (we avoid a discussion of randomized stopping) in $O(\max\{nL_{\mathrm{g}}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|\epsilon^{-1}, n\sigma^2 L_g\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|\epsilon^{-2}\})$. We mention that Nesterov and Spokoiny (2017) hinted at a similar WCC result, but with a strictly worse dependence on n, and different assumptions on $\boldsymbol{\xi}$.

5. Methods for structured objectives

The methods discussed in Sections 2 and 3 assume relatively little about the structure of the objective function f beyond some differentiability required for analysis. Section 4 considered the case where f is convex, which resulted, for example, in improved worst-case complexity results. In this section, we consider a variety of assumptions about additional known structure in f (including non-linear least squares, sparse, composite and minimax-based functional forms) and methods designed to exploit this additional structure. Although problems in this section could be solved by the general-purpose methods discussed in Sections 2 and 3, practical gains should be expected by exploiting the additional structure.

5.1. *Non-linear least squares*

A frequently encountered objective in many applications of computational science, engineering and industry is

$$f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{F}(\boldsymbol{x})\|_2^2 = \frac{1}{2}\sum_{i=1}^{p} F_i(\boldsymbol{x})^2. \tag{5.1}$$

For example, data-fitting problems are commonly cast as (5.1); given data y_i collected at design sites θ_i, one may need to estimate the parameters $\boldsymbol{x}$ of a non-linear model or simulation output that best fit the data. In this scenario, F_i is represented by $F_i(\boldsymbol{x}) = w_i(S_i(\boldsymbol{\theta};\boldsymbol{x}) - y_i)$, which is a weighted residual between the simulation output S_i and target data y_i. In this way, objectives of the form (5.1) (and their correlated residual generalizations) encapsulate both the solution of non-linear equations and statistical estimation problems.

The methods of Zhang, Conn and Scheinberg (2010), Zhang and Conn (2012) and Wild (2017) use the techniques of Section 2.2 to construct models of the individual F_i (thereby obtaining a model of the Jacobian $\nabla \boldsymbol{F}$) in (5.1). These models are then used to generate search directions in a trust-region framework resembling the Levenberg–Marquardt method (Levenberg 1944, Marquardt 1963, Moré 1978). The analysis by Zhang *et al.* (2010) and Zhang and Conn (2012) demonstrates – under certain assumptions, such as $f(\boldsymbol{x}_*) = 0$ at an optimal solution $\boldsymbol{x}_*$ (*i.e.* that the associated data-fitting problem has zero residual) – that the resulting methods achieve the same local quadratic convergence rate does as the Levenberg–Marquardt method. POUNDERS is a trust-region-based method for minimizing objectives of the form (5.1) that uses a full Newton approach for each residual F_i (Wild 2017, Dener *et al.* 2018). Another model-based method (implemented in DFO-LS (Cartis, Fiala, Marteau and Roberts 2018)) for minimizing functions of the form (5.1) – more closely resembling a Gauss–Newton method – is analysed by Cartis and Roberts (2017). Their method is shown to converge to stationary points of (5.1) even when $f(\boldsymbol{x}_*) > 0$, at the expense of slightly weaker theoretical guarantees on the convergence rate.

Kelley (2003) proposes a hybridization of a Gauss–Newton method with implicit filtering (Algorithm 4 from Section 2.3.2) that estimates the Jacobian of $\boldsymbol{F}$ by building linear models of each component F_i using central differences (2.29) with an algorithmically updated difference parameter. This hybrid method is shown to demonstrate superlinear convergence for zero-residual (*i.e.* $f(\boldsymbol{x}_*) = 0$) problems.

Earlier methods also used the vector F in order to more efficiently address objectives of the form (5.1). Spendley (1969) develops a simplex-based algorithm that employs quadratic approximations obtained by interpolating the vector F on the current simplex. Peckham (1970) proposes an iterative process that refines models of each component of F using between $n+1$ and $n+3+n/3$ points. Ralston and Jennrich (1978) also develop derivative-free Gauss–Newton methods and highlight their performance relative to methods that do not exploit the structure in (5.1). Brown and Dennis, Jr (1971) consider a variant of the Levenberg–Marquardt method that approximates gradients using appropriately selected difference parameters.

Li and Fukushima (2000) analyse the convergence of a derivative-free line-search method when assuming that the square of the Jacobian (*i.e.* $\nabla \boldsymbol{F}(\boldsymbol{x})^{\mathrm{T}} \nabla \boldsymbol{F}(\boldsymbol{x})$) of (5.1) is positive-definite everywhere. Grippo and Sciandrone (2007) and La Cruz, Martínez and Raydan (2006) augment a non-monotone line-search method to incorporate information about F in the case where $p = n$ in (5.1). Li and Li (2011) develop a line-search method that exploits a monotonicity property assumed about $\boldsymbol{F}$. La Cruz (2014) and Morini, Porcelli and Toint (2018) address (5.1) in the case where simple convex constraints are present.

5.2. *Sparse objective derivatives*

In some applications, it is known that

$$\nabla^2 f(\boldsymbol{x})_{ij} = \nabla^2 f(\boldsymbol{x})_{ji} = 0 \quad \text{for all } (i,j) \in S, \tag{5.2}$$

for all $\boldsymbol{x} \in \boldsymbol{\Omega}$, where the index set S defines the sparsity pattern of the Hessian. Similarly, one can consider *partially separable* objectives of the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{p} F_i(\boldsymbol{x}) = \sum_{i=1}^{p} F_i(\{\boldsymbol{x}_j\}_{j\in S_i}), \tag{5.3}$$

where each F_i depends only on some subset of indices $S_i \subset \{1,2,\dots,n\}$. The extreme cases of (5.3) are totally separable functions, where $p = n$ and $S_i = \{i\}$ for $i \in \{1,\dots,n\}$. In this special case, (DET) reduces to the minimization of n univariate functions.

Given the knowledge encoded in (5.2) and (5.3), derivative-free optimization methods need not consider interactions between certain components of $\boldsymbol{x}$ because they are known to be exactly zero. In the context of the model-based methods of Section 2.2, particularly when using quadratic models, using this knowledge amounts to dropping monomials in $\boldsymbol{\phi}(\boldsymbol{x})$ in (2.12) corresponding to the non-interacting (i,j) pairs from (5.2). Intuitively, such an action reduces the degrees of freedom in (2.14) when building models, necessitating fewer function evaluations in the right-hand side of (2.14).

Colson and Toint (2005) propose a trust-region method for functions of the form (5.3) that builds and maintains separate fully linear models for the individual F_i in an effort to use fewer objective function evaluations. Similarly, Colson and Toint (2001) propose a method for the case when $\nabla^2 f$ has a band or block structure that exploits knowledge when building models of the objective; the work of Colson and Toint (2002) extends this work to general sparse objective Hessians. Bagirov and Ugon (2006) develop an algorithm that exploits the fact that efficient discrete gradient estimates can be obtained for f having the form (5.3).

In the context of pattern-search methods (discussed in Section 2.1.2), Price and Toint (2006) exploit knowledge of f having the form (5.3) to choose a particular stencil of search directions when forming a positive spanning set. In particular, the stencil is chosen as $\{\boldsymbol{e}_1,\dots,\boldsymbol{e}_n,\boldsymbol{e}_{n+1},\dots,\boldsymbol{e}_{n+p}\}$, where $\{\boldsymbol{e}_1,\dots,\boldsymbol{e}_n\}$ are the elementary basis vectors and

$$\boldsymbol{e}_{n+i} = \sum_{j\in S_i} -\boldsymbol{e}_j,$$

for $i \in \{1,\dots,p\}$. Frimannslund and Steihaug (2010) also develop a DDS method for (5.3), with the search directions determined based on a smoothed quadratic formed from previously evaluated points.

Bandeira, Scheinberg and Vicente (2012) also assume that $\nabla^2 f(\boldsymbol{x})$ is sparse but do not assume knowledge of the sparsity structure (*i.e.* S in (5.2) is not known). They develop a quadratic model-based trust-region method where the models are selected by a minimum 1-norm solution to an underdetermined interpolation system (2.14). Under certain assumptions on f, if n_z is the number of non-zeros in the (unknown) sparsity pattern for $\nabla^2 f$, Bandeira *et al.* (2012) prove that $\boldsymbol{Y}$ in (2.14) must contain only $O((n_z+n)\log(n_z+n)\log(n))$ (as opposed to $O(n^2)$) randomly generated points in order to ensure that the constructed interpolation models are fully quadratic models of f with high probability. This work motivated the analysis of randomized trust-region methods discussed in Section 3.3 because the random underdetermined interpolation models of Bandeira *et al.* (2012) satisfy the assumptions made in Bandeira *et al.* (2014).

In Section 4.2.1, we noted the work of Wang *et al.* (2018), who used assumptions of gradient and Hessian sparsity (in particular, $\|\nabla f(\boldsymbol{x})\|_0 \leq n_z$) to improve the reduce the dependence on n in a WCC of type (4.9) from polynomial to logarithmic. Note that, similar to Bandeira *et al.* (2012), this is an assumption on knowing a universal bound (*i.e.* for all $\boldsymbol{x} \in \boldsymbol{\Omega}$) on the cardinality $\|\nabla f(\boldsymbol{x})\|_0$ rather than the actual non-zero components. Under a similar sparsity assumption, Balasubramanian and Ghadimi (2018) consider the two-point bandit feedback setting discussed in Section 4.2.2 and show that a truncated[10] version of the method proposed in Ghadimi and Lan (2013) has a WCC of type (4.9) in $O(n_z \log(n)^2/\epsilon^2)$. Like the result of Wang *et al.* (2018), this WCC result also exhibits a logarithmic dependence on n, provided $n_z \ll n$. As we will discuss in Section 6.4, Ghadimi and Lan (2013) analyse a method resembling Algorithm 6 to be applied to non-convex f in (STOCH). Balasubramanian and Ghadimi (2018) prove that the unaltered method of Ghadimi and Lan (2013) applied to problems in this sparse setting achieves a WCC of type

$$\mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|] \leq \epsilon, \tag{5.4}$$

in $O(n_z^2 \log(n)^2 \epsilon^{-4})$; this WCC result once again eliminates a polynomial dependence on n that would otherwise exist in a non-sparse setting. This WCC result, however, maintains the same ϵ-dependence as the method of Ghadimi and Lan (2013) in the non-sparse setting; in this sense, the method of Ghadimi and Lan (2013) is ‘automatically’ tuned for the sparse setting.

5.3. *Composite non-smooth optimization*

Sometimes, the objective f in (DET) is known to be non-smooth. Often, one has knowledge about the form of non-smoothness present in the objective,

[10] That is, all but the n_z largest values in $\boldsymbol{x}_{k+1}$ are set to 0 in line 5 of Algorithm 6.

and we discuss methods that exploit specific forms of non-smoothness in this section. For methods that do not access any structural information when optimizing non-smooth functions f, see Sections 2.1.2 and 2.3.

We define composite non-smooth functions as those of the form

$$f(\boldsymbol{x}) = h(\boldsymbol{F}(\boldsymbol{x})), \tag{5.5}$$

where $h : \mathbb{R}^p \to \mathbb{R}$ is a non-smooth function (in contrast to smooth h such as the sum of squares in (5.1)), and $\boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^p$ is continuously differentiable. In some of the works we cite, the definition of a composite non-smooth objective may include an additional smooth function g so that the objective function has the form $f(\boldsymbol{x}) + g(\boldsymbol{x})$, but we omit a discussion of this for the sake of focusing on the non-smooth aspect in (5.5).

5.3.1. Convex h

When h in (5.5) is convex (note that f may still be non-convex due to non-convexity in $\boldsymbol{F}$), one thrust of research extends the techniques in derivative-based composite non-smooth optimization; see the works of Yuan (1985) and Fletcher (1987, Chapter 14). For example, Yuan (1985) use derivatives to construct convex first-order approximations of f near $\boldsymbol{x}$,

$$\ell(\boldsymbol{x} + \boldsymbol{s}) = h(\boldsymbol{F}(\boldsymbol{x}) + \nabla\boldsymbol{F}(\boldsymbol{x})\boldsymbol{s}), \tag{5.6}$$

where $\nabla\boldsymbol{F}$ denotes the Jacobian of $\boldsymbol{F}$; see Hare (2017) for properties of such approximations in the derivative-free setting. By replacing $\nabla\boldsymbol{F}$ in (5.6) with the matrix $\boldsymbol{M}(\boldsymbol{x}_k)$ containing the gradients of a fully linear approximation to $\boldsymbol{F}$ at $\boldsymbol{x}_k$, Grapiglia, Yuan and Yuan (2016) and Garmanjani *et al.* (2016) independently analyse a model-based trust-region method similar to Algorithm 3 from Section 2.2.4 that uses the non-smooth trust-region subproblem

$$\underset{\boldsymbol{s}:\|\boldsymbol{s}\|\le\Delta_k}{\text{minimize}}\, \ell(\boldsymbol{x}_k + \boldsymbol{s}) = h(\boldsymbol{F}(\boldsymbol{x}_k) + \boldsymbol{M}(\boldsymbol{x}_k)\boldsymbol{s}). \tag{5.7}$$

Note that only $\boldsymbol{F}$ is assumed to be a black-box; these methods exploit the fact that h is convex with a known form in order to appropriately solve (5.6). Both Grapiglia *et al.* (2016) and Garmanjani *et al.* (2016) use the stationarity measure of Yuan (1985) in their analysis,

$$\Psi(\boldsymbol{x}) = \ell(\boldsymbol{x}) - \min_{\|\boldsymbol{s}\|\le 1} \ell(\boldsymbol{x} + \boldsymbol{s}), \tag{5.8}$$

for which it is known that $\Psi(\boldsymbol{x}_*) = 0$ if and only if $\boldsymbol{x}_*$ is a critical point of f in the sense that $\ell(\boldsymbol{x}_*) \le \ell(\boldsymbol{x}_* + \boldsymbol{s})$ for all $\boldsymbol{s} \in \mathbb{R}^n$. Worst-case complexity results that bound the effort required to attain $\|\Psi(\boldsymbol{x}_k)\| \le \epsilon$ are included in Table A.1.

Methods using the local approximation (5.6) require convexity of h in order for Ψ in (5.8) to be interpreted as a stationarity measure. From a practical perspective, the form of h directly affects the difficulty of solving the trust-region subproblem. Grapiglia *et al.* (2016) demonstrate this approach on a collection of problems of the form

$$f(\boldsymbol{x}) = \max_{i=1,\dots,p} F_i(\boldsymbol{x}), \tag{5.9}$$

where each F_i is assumed smooth. Garmanjani *et al.* (2016) test their method on a collection of problems of the form

$$f(\boldsymbol{x}) = \|\boldsymbol{F}(\boldsymbol{x})\|_1 = \sum_{i=1}^{p} |F_i(\boldsymbol{x})|. \tag{5.10}$$

For objectives of the form (5.9) or (5.10), the associated trust-region subproblems (5.7) can be cast as linear programs when the ∞-norm defines the trust region. (An early example of such an approach appears in Madsen (1975), where linear approximations to each F_i in (5.9) are constructed.) Although more general convex h could fit into this framework, one must be wary of the difficulty of the resulting subproblems.

Direct-search methods have also been adapted for composite non-smooth functions of specific forms. In these variants, knowledge of h in (5.5) informs the selection of search directions in a manner similar to that described in Section 5.2. Ma and Zhang (2009) and Bogani, Gasparo and Papini (2009) consider the cases of f of the form (5.9) and (5.10), respectively.

Objectives of the form (5.9) are also addressed by Hare, Planiden and Sagastizábal (2019), who develop an algorithm that decomposes such problems into orthogonal subspaces associated with directions of non-smoothness and directions of smoothness. The resulting derivative-free VU-algorithm employs model-based estimates of gradients to form and update this decomposition (Hare 2014). Liuzzi, Lucidi and Sciandrone (2006) address finite minimax problems by converting the original problem into a smooth problem using an exponential penalty function. Their DDS method adjusts the penalty parameter via a rule that depends on the current step size in order to guarantee convergence to a Clarke stationary point.

Approximate gradient-sampling methods are developed and analysed by Hare and Macklem (2013) and Hare and Nutini (2013) for the finite minimax problem (5.9). These methods effectively exploit the subdifferential structure of $h(\boldsymbol{y}) = \max_{i=1,\dots,p} y_i$ and employ derivative-free approximations of each $\nabla F_i(\boldsymbol{x})$. Larson, Menickelly and Wild (2016) propose a variant of gradient sampling, called *manifold sampling*, for objectives of the form (5.10). Unlike (approximate) gradient sampling, manifold sampling does not depend on a random sample of points to estimate the ϵ-subdifferential.

```
Algorithm 8: Smoothing method
Set initial smoothing parameter $\mu_1 > 0$, terminal smoothing parameter
  $\mu_* < \mu_1$, and decrease parameter $\gamma \in (0, 1)$
Choose initial point $\boldsymbol{x}_0$ and smooth optimization method $\mathfrak{M}$
$k \leftarrow 1$
while $\mu_k < \mu_*$ do
    Apply $\mathfrak{M}$ to $f_{\mu_k}$, supplying $\boldsymbol{x}_{k-1}$ as an initial point to $\mathfrak{M}$, until a
      termination criterion is satisfied and $\boldsymbol{x}_k$ is returned
    $\mu_{k+1} \leftarrow \gamma\mu_k$
    $k \leftarrow k + 1$
```

5.3.2. Non-convex h

When h is non-convex, minimization of (5.5) is considerably more challenging than when h is convex. Few methods exist that exploit the structure of non-convex h. One of the many challenges is that the model in (5.6) may no longer be an underestimator of h. Khan *et al.* (2018) propose a manifold sampling method for piecewise linear h; in contrast to the previously discussed methods, this method does not require that h be convex. Other methods applicable for non-convex h employ *smoothing functions*.

As mentioned in Section 2.1.2, the worst-case complexity of DDS methods applied to non-smooth (Lipschitz-continuous) objective functions is difficult to analyse. The reason that DDS methods generate an asymptotically dense set of polling directions is to ensure that no descent directions exist. An exception to this generality, however, is functions for which an appropriate smoothing function exists. Given a locally Lipschitz-continuous f, we say that $f_\mu : \mathbb{R}^n \to \mathbb{R}$ is a smoothing function for f provided that for any $\mu \in (0, \infty)$, f_μ is continuously differentiable and that

$$\lim_{\boldsymbol{z} \to \boldsymbol{x}, \mu \to 0^+} f_\mu(\boldsymbol{z}) = f(\boldsymbol{x}),$$

for all $\boldsymbol{x} \in \mathbb{R}^n$.

Thus, if a smoothing function f_μ exists for f, it is natural to iteratively apply a method for smooth unconstrained optimization to obtain approximate solutions $\boldsymbol{x}_k$ to $\min_{\boldsymbol{x}} f_{\mu_k}(\boldsymbol{x})$ while decreasing μ_k. We roughly prescribe such a smoothing method in Algorithm 8.

Garmanjani and Vicente (2012) consider the DDS framework analysed by Vicente (2013) as the method $\mathfrak{M}$ in Algorithm 8. They terminate $\mathfrak{M}$ when the step-size parameter α of Algorithm 2 is sufficiently small, where the notion of sufficiently small scales with μ_k in Algorithm 8. Garmanjani and Vicente (2012) prove a first-order stationarity result of the form

$$\liminf_{k \to \infty} \|\nabla f_{\mu_k}(\boldsymbol{x}_k)\| = 0. \tag{5.11}$$

Under certain assumptions (for instance, that h satisfies some regularity conditions at $\boldsymbol{F}(\boldsymbol{x}_*)$) this first-order stationarity result is equivalent to $0 \in \partial f(\boldsymbol{x}_*)$; that is, $\boldsymbol{x}_*$ is Clarke stationary.

Garmanjani and Vicente (2012) consider the decrease to be sufficient in line 4 of Algorithm 1 if $f(\boldsymbol{p}_i) < f(\boldsymbol{x}) - c_1\alpha^{3/2}$ for some $c_1 > 0$. If, furthermore, $\mathfrak{M}$ terminates in each iteration of Algorithm 8 when $\alpha < c_2\mu_k^2$ for some $c_2 > 0$, then an upper bound on the number of function evaluations needed to obtain

$$\|\nabla f_{\mu_*}(\boldsymbol{x}_k)\| \leq \epsilon, \tag{5.12}$$

for $\epsilon \in (0,1)$ and $\mu_* \in O(n^{-1/2}\epsilon)$, is in $O(\epsilon^{-3})$; see Table A.1. We note that while the sequence of smoothing parameters μ_k induces a type of limiting behaviour of the gradients (as seen in (5.12)) returned by the method $\mathfrak{M}$ used in Algorithm 8, this still does not necessarily recover elements of the Clarke subdifferential of f. The smoothing functions f_{μ_k} must satisfy an additional *gradient consistency property* in order for Algorithm 8 to produce a sequence of points $\boldsymbol{x}_k$ converging to Clarke stationary points (Rockafellar and Wets 2009, Theorem 9.67).

Garmanjani *et al.* (2016) consider the use of a model-based trust-region method $\mathfrak{M}$ in Algorithm 8. The authors demonstrate the first-order convergence result (5.11); they also prove the same WCC as is proved by Garmanjani and Vicente (2012).

5.4. Bilevel and general minimax problems

Bilevel optimization addresses problems where a *lower-level* objective is embedded within an *upper-level* problem. Bilevel problems take the form

$$\begin{aligned} \underset{\boldsymbol{x}\in\boldsymbol{\Omega}}{\text{minimize}} \quad & f^u(\boldsymbol{x}, \boldsymbol{x}^l) \\ \text{subject to} \quad & \boldsymbol{x}^l \in \underset{\boldsymbol{z}\in\boldsymbol{\Omega}^l}{\arg\min}\{f^l(\boldsymbol{x}, \boldsymbol{z})\}, \end{aligned} \tag{5.13}$$

where $f^u : \boldsymbol{\Omega} \subseteq \mathbb{R}^n \to \mathbb{R}$ and $f^l : \boldsymbol{\Omega}^l \subseteq \mathbb{R}^{n^l} \to \mathbb{R}$. Conn and Vicente (2012) propose a model-based trust-region method for solving (5.14) in the absence of derivative information. They show how to obtain approximations of the upper-level objective by solving the lower-level problem to sufficient accuracy. Mersha and Dempe (2011) and Zhang and Lin (2014) develop DDS-based algorithms for (5.13) under particular assumptions (*e.g.* strict convexity of the lower-level problem).

A special case of (5.13) is when $f^l = -f^u$, which results in the minimax problem (DET), where the objective is given by a maximization:

$$f(\boldsymbol{x}) = \max_{\boldsymbol{x}^l\in\boldsymbol{\Omega}^l} f^l(\boldsymbol{x}, \boldsymbol{x}^l). \tag{5.14}$$

In contrast to the finite minimax problem (5.9), the objective in (5.14) involves a potentially infinite set $\boldsymbol{\Omega}^l$.

Bertsimas, Nohadani and Teo (2010) and Bertsimas and Nohadani (2010) consider (5.14) when exact gradients of f^l may not be available. The authors assume that approximate gradients of f^l are available and propose methods with convergence analysis restricted to functions f in (5.14) that are convex in $\boldsymbol{x}$. Ciccazzo *et al.* (2015) and Latorre, Habal, Graeb and Lucidi (2019) develop derivative-free methods that employ approximate solutions of the inner problem in (5.14). Menickelly and Wild (2019) also consider (5.14) and develop a derivative-free method of outer approximations for more general f. Their analysis shows that the resulting limit points are Clarke stationary for f.

6. Methods for stochastic optimization

We now turn our attention to methods for solving the stochastic optimization problem (STOCH). In Section 4.2, we considered the case where $f(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[\tilde{f}(\boldsymbol{x};\boldsymbol{\xi})]$ is convex. In this section, we lift the assumption of convexity to consider a more general class of stochastic functions $\tilde{f}$.

In general, the analysis of methods for stochastic optimization requires assumptions on the random variable $\boldsymbol{\xi}$. In this section, we use the convention that $\boldsymbol{\xi} \sim \Xi$ denotes that the random variable $\boldsymbol{\xi}$ is from a distribution Ξ and that $\boldsymbol{\xi} \in \Xi$ refers to a random variable in the support of this distribution. Frequently, realizations $\boldsymbol{\xi} \sim \Xi$ are assumed to be independent and identically distributed (i.i.d.). Throughout this section, we assume that $\mathbb{E}_{\boldsymbol{\xi}}[\tilde{f}(\boldsymbol{x};\boldsymbol{\xi})]$ exists for each $\boldsymbol{x} \in \boldsymbol{\Omega}$ and $f(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[\tilde{f}(\boldsymbol{x};\boldsymbol{\xi})]$; that is, the objective of (STOCH) is well-defined. Another common assumption in the stochastic optimization literature is that some bound on the variance of $\tilde{f}(\boldsymbol{x};\boldsymbol{\xi})$ is assumed, that is,

$$\mathbb{E}_{\boldsymbol{\xi}}[(\tilde{f}(\boldsymbol{x};\boldsymbol{\xi}) - f(\boldsymbol{x}))^2] < \sigma^2 < \infty \quad \text{for all } \boldsymbol{x} \in \boldsymbol{\Omega}. \tag{6.1}$$

If, for a given $\boldsymbol{x}$, $\nabla_{\boldsymbol{x}}\tilde{f}(\boldsymbol{x};\boldsymbol{\xi})$ exists for each $\boldsymbol{\xi} \in \Xi$, then under certain regularity conditions it follows that $\nabla f(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[\nabla_{\boldsymbol{x}}\tilde{f}(\boldsymbol{x};\boldsymbol{\xi})]$; one such regularity condition is that

$$\tilde{f}(\cdot;\boldsymbol{\xi}) \text{ is } L_{\tilde{f}(\cdot;\boldsymbol{\xi})}\text{-Lipschitz-continuous and } \mathbb{E}_{\boldsymbol{\xi}}\big[L_{\tilde{f}(\cdot;\boldsymbol{\xi})}\big] < \infty.$$

We note that when first-order information is available, the assumption (6.1) is often replaced by an assumption on the variance of the expected gradient norm; see *e.g.* Bottou, Curtis and Nocedal (2018, Assumption 4.3). In this setting, a key class of methods for (STOCH) are *stochastic approximation* (SA) methods; see the paper proposing SA methods by Robbins and Monro (1951) and a survey of modern SA methods (often also referred to as ‘stochastic gradient’ methods when first-order information is available) by Bottou *et al.* (2018). Here we focus on situations where no objective derivative information is available; that is, stochastic gradient methods are not directly applicable. That said, some of the work we discuss attempts to approximate stochastic gradients, which are then used in an SA framework. As discussed in Section 1, we will not address global optimization methods, such as Bayesian optimization.[11]

Section 6.1 discusses stochastic approximation methods, and Section 6.2 presents direct-search methods for stochastic optimization. In Section 6.3 we highlight modifications to derivative-free model-based methods to address (STOCH), and in Section 6.4 we discuss bandit methods for (nonconvex) stochastic optimization.

6.1. Stochastic and sample-average approximation

One of the first analysed approaches for solving (STOCH) is the method of Kiefer and Wolfowitz (1952), inspired by the SA method of Robbins and Monro (1951). We state the basic Kiefer–Wolfowitz framework in Algorithm 9. Since Kiefer and Wolfowitz (1952) consider only univariate problems, Algorithm 9 is in fact the multivariate extension first of Blum (1954*b*). In Algorithm 9, $\nabla f(\boldsymbol{x}_k)$ is approximated by observing realizations of $\tilde{f}$ using central differences. That is, $\nabla f(\boldsymbol{x}_k)$ is approximated by

$$\boldsymbol{g}^{\mathrm{K}}(\boldsymbol{x}_k;\mu_k;\boldsymbol{\xi}_k) = \begin{bmatrix} \dfrac{\tilde{f}(\boldsymbol{x}_k+\mu_k\boldsymbol{e}_1;\boldsymbol{\xi}_1^+) - \tilde{f}(\boldsymbol{x}_k-\mu_k\boldsymbol{e}_1;\boldsymbol{\xi}_1^-)}{2\mu_k} \\ \vdots \\ \dfrac{\tilde{f}(\boldsymbol{x}_k+\mu_k\boldsymbol{e}_n;\boldsymbol{\xi}_n^+) - \tilde{f}(\boldsymbol{x}_k-\mu_k\boldsymbol{e}_n;\boldsymbol{\xi}_n^-)}{2\mu_k} \end{bmatrix}, \tag{6.2}$$

where $\mu_k > 0$ is a difference parameter, $\boldsymbol{e}_i$ is the ith elementary basis vector, and $2n$ realizations $\boldsymbol{\xi} \sim \Xi$ are employed. The next point $\boldsymbol{x}_{k+1}$ is then set to be $\boldsymbol{x}_k - \alpha_k \boldsymbol{g}^{\mathrm{K}}(\boldsymbol{x}_k;\mu_k;\boldsymbol{\xi}_k)$, where $\alpha_k > 0$ is a step-size parameter. As in Section 3, we note that $\boldsymbol{x}_{k+1}$ is a random variable that depends on the filtration generated by the method before $\boldsymbol{x}_{k+1}$ is realized; this will be the case throughout this section. In the SA literature, the sequences $\{\alpha_k\}$ and $\{\mu_k\}$ are often referred to as *gain* sequences.

Because evaluation of the function f requires computing an expectation (and in contrast to the primarily monotone algorithms in Section 2), stochastic optimization methods generally do not monotonically decrease f. This is exemplified by Algorithm 9, which updates $\boldsymbol{x}_{k+1}$ without considering the value of $\tilde{f}(\boldsymbol{x}_{k+1};\boldsymbol{\xi})$ for *any* realization of $\boldsymbol{\xi}$.

[11] We recommend Shahriari *et al.* (2016) and Frazier (2018) to readers interested in recent surveys of Bayesian optimization.

Algorithm 9: Kiefer–Wolfowitz method

```
Choose initial point $\boldsymbol{x}_0$, sequence of step sizes $\{\alpha_k\}$ and sequence of
  difference parameters $\{\mu_k\}$
for $k = 0, 1, 2, \ldots$ do
    Generate $\boldsymbol{\xi}_k = (\boldsymbol{\xi}_1^+, \boldsymbol{\xi}_1^-, \ldots, \boldsymbol{\xi}_n^+, \boldsymbol{\xi}_n^-)$
    Compute gradient estimate $\boldsymbol{g}_k^{\mathrm{K}}(\boldsymbol{x}_k; \mu_k; \boldsymbol{\xi}_k)$ via (6.2)
    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k - \alpha_k \boldsymbol{g}_k^{\mathrm{K}}(\boldsymbol{x}_k; \mu_k; \boldsymbol{\xi}_k)$
```

Historically, Algorithm 9 has been analysed by a community more concerned with stochastic processes than with optimization. Hence, convergence results differ from those commonly found in the optimization literature. For example, many results in the SA literature consider a continuation of the dynamics of Algorithm 9 applied to the deterministic f as an ordinary differential equation (ODE) in terms of $\boldsymbol{x}(t) : \mathbb{R} \to \mathbb{R}^n$. That is, they consider

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = -\nabla f(\boldsymbol{x}), \quad \boldsymbol{x} = \boldsymbol{x}(t).$$

and define the set of fixed points of the ODE, $\boldsymbol{S} = \{\boldsymbol{x} : \nabla f(\boldsymbol{x}) = 0\}$. Many convergence results then demonstrate that the continuation $\boldsymbol{x}(t)$ satisfies $\boldsymbol{x}(t) \to \boldsymbol{S}$ with probability one as the continuous iteration counter $t \to \infty$; see Kushner and Yin (2003) for a complete treatment of such ODE results.

In order to prove that the sequence of points $\boldsymbol{x}_k$ generated by Algorithm 9 converges almost surely (*i.e.* with probability one), conditions must be placed on the objective function, step sizes and difference parameters. In the SA literature there is no single consistent set of conditions, but there are nearly always conditions on the sequence of step sizes $\{\alpha_k\}$ requiring $\alpha_k \to 0$ and $\sum_k \alpha_k = \infty$. Intuitively, this divergence condition ensures that any point in the domain $\boldsymbol{\Omega}$ can be reached, independent of the history of iterations. As one example of convergence conditions, Bhatnagar, Prasad and Prashanth (2013) prove almost sure convergence of Algorithm 9 under the following assumptions (simplified for presentation).

(1) The sequences of step sizes and difference parameters satisfy $\alpha_k > 0$, $\mu_k > 0$, $\alpha_k \to 0$, $\mu_k \to 0$, $\sum_k \alpha_k = \infty$ and $\sum_k \alpha_k^2 \mu_k^{-2} < \infty$.

(2) The realizations $\boldsymbol{\xi} \sim \boldsymbol{\Xi}$ are i.i.d. and the distribution $\boldsymbol{\Xi}$ has a finite second moment.

(3) The function f is in $\mathcal{LC}^1$.

(4) $\sup_k\{\|\boldsymbol{x}_k\|\} < \infty$ with probability one.

Similar assumptions on algorithms of the form Algorithm 9 appear throughout the SA literature (Blum 1954*a*, Derman 1956, Sacks 1958, Fabian 1971, Kushner and Huang 1979, Ruppert 1991, Spall 2005). Convergence of Algorithm 9 under similar assumptions to those above, but with the modification that μ_k is fixed in every iteration to a sufficiently small constant (that scales inversely with $L_{\rm g}$), is additionally demonstrated by Bhatnagar *et al.* (2013).

In terms of WCCs, the convergence rates that have been historically derived for Algorithm 9 are also non-standard for optimization. In particular, results concerning convergence rates are typically shown as a convergence in distribution (Durrett 2010, Chapter 3.2): given a fixed $\boldsymbol{x}_* \in \boldsymbol{S}$,

$$\frac{1}{k^\gamma}(\boldsymbol{x}_k - \boldsymbol{x}_*) \to \mathcal{N}(\boldsymbol{0}, \boldsymbol{B}), \tag{6.3}$$

where $\gamma > 0$ and $\boldsymbol{B}$ is a covariance matrix, the entries of which depend on algorithmic parameters and $\nabla^2 f(\boldsymbol{x}_*)$ (provided it exists). With few assumptions on $\boldsymbol{\xi}$, it has been shown that (6.3) holds with $\gamma = 1/3$ (Spall 1992, L'Ecuyer and Yin 1998). Observe that these convergence rates are distinct from WCC results like those in (2.2).

Later, the use of *common random numbers* (CRNs) was considered. In contrast to (6.2), which employs a realization $\boldsymbol{\xi}_k = (\boldsymbol{\xi}_1^+, \boldsymbol{\xi}_1^-, \dots, \boldsymbol{\xi}_n^+, \boldsymbol{\xi}_n^-)$, a gradient estimator in the CRN regime uses a single realization $\boldsymbol{\xi}_k$ and has the form

$$\boldsymbol{g}^{\rm K}(\boldsymbol{x}_k; \mu_k; \boldsymbol{\xi}_k) = \begin{bmatrix} \delta_{\rm c}(\tilde{f}(\cdot; \boldsymbol{\xi}_k); \boldsymbol{x}_k; \boldsymbol{e}_1; \mu_k) \\ \vdots \\ \delta_{\rm c}(\tilde{f}(\cdot; \boldsymbol{\xi}_k); \boldsymbol{x}_k; \boldsymbol{e}_n; \mu_k) \end{bmatrix}, \tag{6.4}$$

where $\delta_{\rm c}(\cdot)$ is defined in (2.29)

The difference between (6.2) and (6.4) is analogous to the difference between one-point and two-point bandit feedback in the context of bandit problems (see Section 4.2). In the CRN regime, we can recall a single realization $\boldsymbol{\xi}_k$ to compute a finite-difference approximation in each coordinate direction. By using (6.4) as the gradient estimator in Algorithm 9, the rate (6.3) holds with $\gamma = 1/2$ (L'Ecuyer and Yin 1998, Kleinman, Spall and Naiman 1999). Thus, as in the analysis of bandit methods, the use of CRNs allows for strictly better convergence rate results.

Dai (2016*a*, 2016*b*) studies the complexity of Algorithm 9, as well as a method that uses the estimator (6.2) in Algorithm 7, under varying assumptions on $\boldsymbol{\Xi}$. Dai considers a gradient estimate of the form (6.4) with $\delta_{\rm f}(\tilde{f}(\cdot; \boldsymbol{\xi}_k); \boldsymbol{x}_k; \boldsymbol{e}_i; \mu_k)$ replacing each central difference; recall the definition of $\delta_{\rm f}(\cdot)$ in (2.28). Dai demonstrates that the best rate of the form (6.3) achievable by Algorithm 9 with forward differences has $\gamma = 1/3$, even when common random numbers are used. However, a rate of the form (6.3) with $\gamma = 1/2$ can be achieved using forward differences in Algorithm 7; Dai

(2016*a*, 2016*b*) draws parallels between this result and the WCC of Duchi *et al.* (2015), discussed in Section 4.2.2.

We remark that the gradient estimate (6.2) used in Algorithm 9 requires $2n$ evaluations of $\tilde{f}$ per iteration. Although replacing $\delta_{\mathrm{c}}(\tilde{f}(\cdot;\boldsymbol{\xi}_k);\boldsymbol{x}_k;\boldsymbol{e}_i;\mu_k)$ with $\delta_{\mathrm{f}}(\tilde{f}(\cdot;\boldsymbol{\xi}_k);\boldsymbol{x}_k;\boldsymbol{e}_i;\mu_k)$ could reduce this cost to $n+1$ evaluations of $\tilde{f}$ per iteration, it is still desirable to reduce this per-iteration cost from $O(n)$ to $O(1)$ evaluations. The SPSA method of Spall (1992) achieves this goal by using the gradient estimator

$$\boldsymbol{g}^{\mathrm{S}}(\boldsymbol{x}_k;\mu_k;\boldsymbol{\xi}_k;\boldsymbol{u}_k) = \delta_{\mathrm{c}}(\tilde{f}(\cdot,\boldsymbol{\xi}_k);\boldsymbol{x}_k;\boldsymbol{u}_k;\mu_k)\begin{bmatrix}\dfrac{1}{[\boldsymbol{u}_k]_1}\\ \vdots \\ \dfrac{1}{[\boldsymbol{u}_k]_n}\end{bmatrix}, \tag{6.5}$$

where $\boldsymbol{u}_k \in \mathbb{R}^n$ is randomly generated from some distribution in each iteration.

The construction of (6.5) requires evaluations of $\tilde{f}(\cdot;\boldsymbol{\xi}_k)$ at exactly two points. Algorithm 9 is then modified by replacing the gradient estimator $\boldsymbol{g}_k^{\mathrm{K}}(\boldsymbol{x}_k;\mu_k;\boldsymbol{\xi}_k)$ with $\boldsymbol{g}_k^{\mathrm{S}}(\boldsymbol{x}_k;\mu_k;\boldsymbol{\xi}_k;\boldsymbol{u}_k)$. Informally, the conditions on the distribution governing $\boldsymbol{u}_k$ originally proposed by Spall (1992) cause each entry of $\boldsymbol{u}_k$ to be bounded away from 0 with high probability (intuitively, to avoid taking huge steps). A simple example distribution satisfying these properties is to let each entry of $\boldsymbol{u}_k$ independently follow a Bernoulli distribution with support $\{1,-1\}$, both events occurring with probability $1/2$. Under appropriate assumptions resembling those for Algorithm 9, the sequence $\{\boldsymbol{x}_k\}$ generated by SPSA can be shown to converge in the same sense as Algorithm 9. Convergence rates of the form (6.3) matching those obtained for Algorithm 9 have also been established (Gerencsér 1997, Kleinman *et al.* 1999).

The performance of SA methods is highly sensitive to the chosen sequence of step sizes $\{\alpha_k\}$ (Hutchison and Spall 2013). This mirrors the situation in gradient-based SA methods where the tuning of algorithmic parameters is an active area of research (Diaz, Fokoue-Nkoutche, Nannicini and Samulowitz 2017, Ilievski, Akhtar, Feng and Shoemaker 2017, Balaprakash *et al.* 2018).

The SA methods above consider only a single evaluation of the stochastic function $\tilde{f}$ at any point. Other methods more accurately estimate $f(\boldsymbol{x}_k)$ by querying $\tilde{f}(\boldsymbol{x}_k;\boldsymbol{\xi})$ for multiple, different realizations ('samples') of $\boldsymbol{\xi}$. These methods belong to the framework of sample average approximation, wherein the original problem (STOCH) is replaced with a (sequence of) deterministic *sample-path problem(s)*:

$$\operatorname*{minimize}_{\boldsymbol{x}\in\boldsymbol{\Omega}} \frac{1}{p}\sum_{i=1}^{p} \tilde{f}(\boldsymbol{x};\boldsymbol{\xi}_i). \tag{6.6}$$

Retrospective approximation methods (Chen and Schmeiser 2001) vary the number of samples, p, in a predetermined sequence $\{p_0, p_1, \ldots\}$; the accuracy to which each instance of (6.6) subproblem is solved can also vary as a sample average approximation method progresses. Naturally, the performance of such a method depends critically on the sequence of sample sizes and accuracies used at each iteration; Pasupathy (2010) characterizes a class of sequences of predetermined sample sizes and accuracies for which derivative-free retrospective approximation methods can be shown to converge for smooth objectives.

Other approaches dynamically adjust the number of samples p from iteration to the next. For example, the method of Pasupathy, Glynn, Ghosh and Hashemi (2018) adjusts the number of samples p_k to balance the contributions from deterministic and stochastic errors in iteration k. The stochastic error at $\boldsymbol{x}_k$ is then

$$\left| f(\boldsymbol{x}_k) - \frac{1}{p_k} \sum_{i=1}^{p_k} \tilde{f}(\boldsymbol{x}_k; \boldsymbol{\xi}_{k,i}) \right|.$$

The deterministic error is the difference between the objective f and a specified approximation; for example, the deterministic error at $\boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k)$ using a first-order Taylor approximation is

$$|f(\boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k)) - (f(\boldsymbol{x}_k) - \alpha_k \|\nabla f(\boldsymbol{x}_k)\|^2)|.$$

Pasupathy *et al.* (2018) establish convergence rates for a variant of Algorithm 9 drawing independent samples $\{\boldsymbol{\xi}_{k,1}, \ldots, \boldsymbol{\xi}_{k,p_k}\}$ in each iteration.

6.2. *Direct-search methods for stochastic optimization*

Unsurprisingly, researchers have modified methods for deterministic objectives in order to produce methods appropriate for stochastic optimization. For example, in the paper inspiring Nelder and Mead (1965), Spendley *et al.* (1962) propose re-evaluating the point corresponding to the best simplex vertex if it hasn't changed in $n + 1$ iterations, saying that if the vertex is best 'only by reason of errors of observation, it is unlikely that the repeat observation will [be the best observed point], and the point will be eliminated in due course'. Barton and Ivey, Jr (1996) propose modifications to the Nelder–Mead method in order to avoid premature termination due to repeated shrinking. To alleviate this problem, they suggest reducing the amount the simplex is shrunk, re-evaluating the best point after each shrink operation, and re-evaluating each reflected point before performing a contraction. Chang (2012) proposes a Nelder–Mead variant that samples candidate points and all other points in the simplex an increasing number of times; this method ultimately ensures that stochasticity in the function evaluations will not affect the correct ranking of simplex vertices.

Sriver, Chrissis and Abramson (2009) augment a GPS method with a ranking and selection procedure and dynamically determine the number of samples performed for each polling point. The ranking and selection procedure allows the method to also address cases where $\boldsymbol{x}$ contains discrete variables. For the case of additive unbiased, Gaussian noise (*i.e.* $\tilde{f}(\boldsymbol{x};\boldsymbol{\xi}) = f(\boldsymbol{x}) + \sigma\boldsymbol{\xi}$ with $\boldsymbol{\xi}$ from a standard normal distribution and $\sigma > 0$ finite), they prove that the resulting method converges almost surely to a stationary point of f. For problems involving more general distributions, Kim and Zhang (2010) consider a DDS method that employs the sample mean

$$\frac{1}{p_k}\sum_{i=1}^{p_k} \tilde{f}(\boldsymbol{x};\boldsymbol{\xi}_{k,i}), \tag{6.7}$$

with a dynamically increasing sample size p_k. They establish a consistency result and appeal to the convergence properties of DDS methods. Sankaran, Audet and Marsden (2010) propose a surrogate-assisted method for stochastic optimization inspired by stochastic collocation techniques (see *e.g.* Gunzburger, Webster and Zhang 2014). Convergence for the method is established by appealing to the GPS and MADS mechanisms underlying the method.

Chen and Kelley (2016) consider an implicit-filtering method in which values of f are observable only through the sample average (6.7). Chen and Kelley (2016) demonstrate that the sequence of points generated by the method converges (*i.e.* $\{\nabla f(\boldsymbol{x}_k)\}$ admits a subsequence that converges to zero) with probability one if the sample size p_k increases to infinity. Algorithmically, p_k is adjusted to scale with the square of the inverse of the stencil step size (Δ_k in Algorithm 4).

Chen, Kelley, Xu and Zhang (2018*b*) consider the bound-constrained minimization of a composite non-smooth function of the form (5.5), where h is Lipschitz-continuous (but non-smooth) and $\boldsymbol{F}$ is continuously differentiable. However, they assume that values of $\boldsymbol{F}$ are observable only through sample averages and that a smoothing function h_μ of h (as discussed in Section 5.3.2) is available. They show that with probability one, the sequence of points from a smoothed implicit-filtering method converges to a first-order stationary point, where the stationarity measure is appropriate for non-smooth optimization.

6.3. Model-based methods for stochastic optimization

Analysis of the model-based trust-region methods in Section 2.2 generally depends on the construction of fully linear models of a deterministic function f; see (2.9). In particular, methods of the form of Algorithm 3 typically require that a model m_k satisfy

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{x}_k + \boldsymbol{s})| \leq \kappa_{\mathrm{ef}}\Delta_k^2 \quad \text{for all } \boldsymbol{s} \in \mathcal{B}(\mathbf{0};\Delta_k).$$

A natural model-based trust-region approach to stochastic optimization is to build a model m_k of the function f by fitting the model to observed values of the stochastic function $\tilde{f}$. Intuitively, if such models satisfy (2.9), then an extension of the analysis described in Section 2.2.4 should also apply to the minimization of f in (STOCH). The methods described here formalize the approximation properties of such models (which are stochastic because of their dependence on $\boldsymbol{\xi}$) and employ the models in a trust-region framework. For example, by employing an estimator $\bar{f}_p$ of f at each interpolation point $\boldsymbol{x}$ used in model construction, we can replace each function value $f(\boldsymbol{x})$ with $\bar{f}_p(\boldsymbol{x})$ in the interpolation system (2.14). One example of such an estimator $\bar{f}_p$ is the sample average (6.7).

Early work in applying derivative-free trust-region methods for stochastic optimization includes that of Deng and Ferris (2006), which modifies the UOBYQA method of Powell (2002). The kth iteration of the method of Deng and Ferris (2006) uses Bayesian techniques to dynamically update a budget of p_k new $\tilde{f}$ evaluations. This budget is then apportioned among the current set of interpolation points $\boldsymbol{y} \in \boldsymbol{Y}$ in order to reduce the variance in each value of $\bar{f}_{p_k}(\boldsymbol{y})$, with the authors using the sample mean for the estimator $\bar{f}_{p_k}$. Deng and Ferris (2009) show that, given assumptions on the sequence of evaluated $\boldsymbol{\xi}$ (*i.e.* the sample path), every limit point $\boldsymbol{x}_*$ produced by this method is stationary with probability 1.

Another method in this vein, STRONG, was proposed by Chang, Hong and Wan (2013) and combines response surface methodology (Box and Draper 1987) with a trust-region mechanism. In the analysis of STRONG, it is assumed that model gradients $\nabla m_k(\boldsymbol{x}_k)$ almost surely equal the true gradients $\nabla f(\boldsymbol{x}_k)$ as $k \to \infty$, which is algorithmically encouraged by monotonically increasing the sample size p_k in an inner loop. QNSTOP by Castle (2012) presents a similar approach using response surface models in a trust-region framework, but its convergence analysis and assumptions mirror those of stochastic approximation methods.

Both Larson and Billups (2016) and Chen, Menickelly and Scheinberg (2018*a*) build on the idea of probabilistically fully linear models in (3.5), which essentially says that the condition (2.9) needs to hold on a given iteration only with some probability (Bandeira *et al.* 2014). In contrast to the usage of such models in randomized methods for deterministic objectives (the subject of Section 3.3), in stochastic optimization the filtration in (3.5) also includes the realizations of the stochastic evaluations of $\tilde{f}$. This probabilistic notion of uniform local model quality is powerful. For example, although the connection is not made by Regier, Jordan and McAuliffe (2017), this notion of model quality implies probabilistic descent properties such as those required by Regier *et al.* (2017). This implication is an example of a setting in which stochastic gradient estimators can be replaced by gradients of probabilistically fully linear models.

One way to satisfy (3.5) is to build a regression model using randomly sampled points. For example, Menickelly (2017, Theorem 4.2.6) shows that evaluating $\tilde{f}$ on a sufficiently large set of points uniformly sampled from $\mathcal{B}(\boldsymbol{x}_k; \Delta_k)$ can be used to construct a probabilistically fully linear regression model.

Larson and Billups (2016) prove convergence of a probabilistic variant of Algorithm 3 in the sense that, for any $\epsilon > 0$,

$$\lim_{k\to\infty} \mathbb{P}[\|\nabla f(\boldsymbol{x}_k)\| > \epsilon] = 0.$$

Under similar assumptions, Chen *et al.* (2018*a*) prove almost sure convergence to a stationary point, that is,

$$\lim_{k\to\infty} \|\nabla f(\boldsymbol{x}_k)\| = 0 \quad \text{with probability one.} \tag{6.8}$$

Blanchet, Cartis, Menickelly and Scheinberg (2019) provide a WCC result for the variant of Algorithm 3 presented by Chen *et al.* (2018*a*). Blanchet *et al.* (2019) extend the analysis of Cartis and Scheinberg (2018) to study the stopping time of the stochastic process generated by the method of Chen *et al.* (2018*a*). In contrast to previous WCC results discussed in this survey, which bound the number of function evaluations N_ϵ needed to attain some form of *expected ϵ-optimality* (*e.g.* (3.2) or (5.4)), Blanchet *et al.* (2019) prove that the *expected number of iterations*, $\mathbb{E}[T_\epsilon]$, needed to achieve (2.2) is in $O(\epsilon^{-2})$. Paquette and Scheinberg (2018) apply similar analysis to a derivative-free stochastic line-search method, where they demonstrate that for non-convex f, $\mathbb{E}[T_\epsilon] \in O(\epsilon^{-2})$, while for convex and strongly convex f, $\mathbb{E}[T_\epsilon] \in O(\epsilon^{-1})$ and $\mathbb{E}[T_\epsilon] \in O(\log(\epsilon^{-1}))$, respectively. Since the number of function evaluations per iteration of the derivative-free methods of Blanchet *et al.* (2019) and Paquette and Scheinberg (2018) is highly variable across iterations, the total work (in terms of function evaluations) is not readily apparent from such WCC results.

Larson and Billups (2016) and Chen *et al.* (2018*a*) demonstrate that sampling $\tilde{f}$ on $\mathcal{B}(\boldsymbol{x}_k; \Delta_k)$ of the order of Δ_k^{-4} times will ensure that (2.9) holds (*i.e.* one can obtain a fully linear model) with high probability. Shashaani, Hunter and Pasupathy (2016) and Shashaani, Hashemi and Pasupathy (2018) take a related but distinct approach. As opposed to requiring that models be probabilistically fully linear, their derivative-free trust-region method performs adaptive Monte Carlo sampling both at current points $\boldsymbol{x}_k$ and interpolation points; the number of samples p_k is chosen to balance a measure of statistical error with the optimality gap at $\boldsymbol{x}_k$. Shashaani *et al.* (2018) prove that their method achieves almost sure convergence of the form (6.8).

A model-based trust-region method for constrained stochastic optimization, SNOWPAC, is developed by Augustin and Marzouk (2017). Their method addresses the stochasticity by employing Gaussian process-based models of robustness measures such as expectation and conditional value at risk. The approach used is an extension of the constrained deterministic method NOWPAC of Augustin and Marzouk (2014), which we discuss in Section 7.

6.4. Bandit feedback methods

While much of the literature on bandit methods for stochastic optimization focuses on convex objectives f (as discussed in Section 4.2), here we discuss treatment of non-convex objectives f. We recall our notation and discussion from Section 4.2, in particular the notion of regret minimization shown in (4.7).

In the absence of convexity, regret bounds do not translate into bounds on optimization error as easily as in (4.8). Some works address the case where each $\tilde{f}(\cdot;\boldsymbol{\xi}_k)$ in (4.7) is Lipschitz-continuous and employ a partitioning of the feasible region $\boldsymbol{\Omega}$ (Kleinberg, Slivkins and Upfal 2008, Bubeck, Stoltz and Yu 2011*b*, Bubeck, Munos, Stoltz and Szepesvári 2011*a*, Valko, Carpentier and Munos 2013, Zhang, Yang, Jin and Zhou 2015). These methods employ global optimization strategies that we do not discuss further here.

In another line of work, Ghadimi and Lan (2013) consider the application of an algorithm like Algorithm 6 with the choice of gradient estimator $\boldsymbol{g}_\mu(\boldsymbol{x};\boldsymbol{u};\boldsymbol{\xi})$ from (4.12). Under an assumption of bounded variance of the estimator (*i.e.* $\mathbb{E}_{\boldsymbol{\xi}}[\|\boldsymbol{g}_\mu(\boldsymbol{x};\boldsymbol{u};\boldsymbol{\xi}) - \nabla f(\boldsymbol{x})\|^2] \leq \sigma^2$), Ghadimi and Lan (2013) prove a WCC result similar to the one they obtained in the convex case; see Section 4.2.2. They show that an upper bound on the (randomized) number of iterations needed to attain

$$\mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|^2] \leq \epsilon \tag{6.9}$$

is in $O(\max\{nL_gR_{\boldsymbol{x}}\epsilon^{-1}, nL_gR_{\boldsymbol{x}}\sigma^2\epsilon^{-2}\})$. Notice that the stationarity condition given in (6.9) involves a square on the gradient norm, making it distinct from a result like (3.2) or (5.4). Thus, assuming σ^2 is sufficiently large, the result of Ghadimi and Lan (2013) translates to a WCC of type (5.4) in $O(n^2\epsilon^{-4})$.

Balasubramanian and Ghadimi (2018, 2019) propose a method that uses two-point bandit feedback (*i.e.* a gradient estimator from (4.12)) within a derivative-free conditional gradient method (Ghadimi 2019). The gradient estimator is used to define a linear model, which is minimized over $\boldsymbol{\Omega}$ to produce a trial step. If $\boldsymbol{\Omega}$ is bounded, they show a WCC of type (5.4) that again grows like ϵ^{-4}.

By replacing gradients with estimators of the form (4.12) in the stochastic variance-reduced gradient framework of machine learning (Reddi *et al.* 2016), Liu *et al.* (2018) prove a WCC of type (6.9) in $O(n\epsilon^{-1} + b^{-1})$, where b is the size of a minibatch drawn with replacement in each iteration. Gu, Huo and

Huang (2016) prove a similar WCC result in an asynchronous parallel computing environment for a distinct method using minibatches for variance reduction.

7. Methods for constrained optimization

In this section, we discuss derivative-free methods for problems where the feasible region $\mathbf{\Omega}$ is a proper subset of $\mathbb{R}^n$. In the derivative-free setting, such constrained optimization problems can take many forms since an additional distinction is associated with the derivative-free nature of objective and constraint functions. For example, and in contrast to the preceding sections, a derivative-free constrained optimization problem may involve an objective function f for which a gradient is made available to the optimization method. The problem is still derivative-free if there is a constraint function defining the feasible region $\mathbf{\Omega}$ for which a (sub)gradient is not available to the optimization method.

As is common in many application domains where derivative-free methods are applied, the feasible region $\mathbf{\Omega}$ may also involve discrete choices. In particular, these choices can include categorical variables that are either ordinal (*e.g.* letter grades in {A, B, C, D, F}) or non-ordinal (*e.g.* compiler type in {flang, gfortran, ifort}). Although ordinal categorical variables can be mapped to a subset of the reals, the same cannot be done for non-ordinal variables. Therefore, we generalize the formulations of (DET) and (STOCH) to the problem

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{y}}{\text{minimize}} \quad & f(\boldsymbol{x},\boldsymbol{y}) \\ \text{subject to} \quad & \boldsymbol{x} \in \mathbf{\Omega} \subset \mathbb{R}^n \\ & \boldsymbol{y} \in \mathbf{N}, \end{aligned} \tag{CON}$$

where $\boldsymbol{y}$ represents a vector of non-ordinal variables and $\mathbf{N}$ is a finite set of feasible values. Here we assume that discrete-valued ordinal variables are included in $\boldsymbol{x}$. Furthermore, most of the methods we discuss do not explicitly treat non-ordinal variables $\boldsymbol{y}$; hence, except where indicated, we will drop the use of $\boldsymbol{y}$.

Similar to Section 5, here we distinguish methods based on the assumptions made about the problem structure. We organize these assumptions based on the black-box optimization constraint taxonomy of Le Digabel and Wild (2015), which characterizes the type of constraint functions that occur in a particular specification of a derivative-free optimization problem. When constraints are explicitly stated (*i.e.* ‘known’ to the method), this taxonomy takes the form of the tree in Figure 7.1.

The first distinction in Figure 7.1 is whether a constraint is algebraically available to the optimization method or whether it depends on a black-box

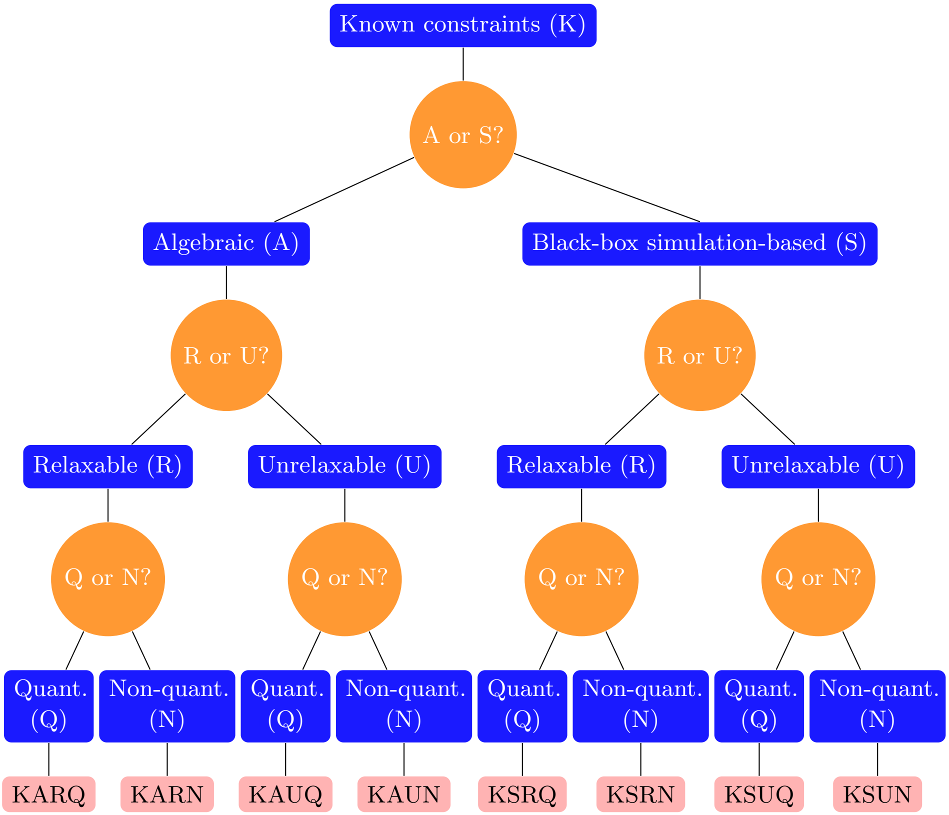

Figure 7.1. Tree-based taxonomy of known (*i.e.* non-hidden) constraints from Le Digabel and Wild (2015).

simulation. In the context of derivative-free optimization, we will assume that it is these latter constraint functions for which a (sub)gradient is not made available to the optimization method. Algebraic constraints are those for which a functional form or simple projection operator is provided to the optimization method. Section 7.1 discusses methods that exclusively handle algebraic constraints. Examples of such algebraic constraints have been discussed earlier in this paper (*e.g.* Sections 3.1 and 4), wherein it is assumed that satisfaction of the constraints (*e.g.* through a simple projection) is trivial relative to evaluation of the objective. This imbalance between the ease of the constraint and objective functions is also the subject of recent WCC analysis (Cartis, Gould and Toint 2018).

Section 7.2 discusses methods that target situations where one or more constraints do not have available derivatives.

The next distinction in Figure 7.1 is whether a constraint can be relaxed or whether the constraint must be satisfied in order to obtain meaningful information for the objective f and/or other constraint functions. Unrelaxable constraints are a relatively common occurrence in derivative-free optimization. In contrast to classic optimization, constraints are sometimes introduced solely to prevent errors in the evaluation of, for example, a simulation-based objective function. Methods for addressing relaxable algebraic constraints are discussed in Section 7.1.1, and unrelaxable algebraic constraints are the focus of Section 7.1.2.

Hidden constraints are not represented in Figure 7.1. Hidden constraints are constraints that are not explicitly stated in a problem specification. Violating these constraints is detected only when attempting to evaluate the objective or constraint functions; for example, a simulation may fail to return output, thus leaving one of these functions undefined. Some derivative-free methods directly account for the possibility that such failures may be present despite not being explicitly stated. Hidden constraints have been addressed in works including those of Avriel and Wilde (1967), Choi and Kelley (2000), Choi *et al.* (2000), Carter *et al.* (2001), Conn, Scheinberg and Toint (2001), Huyer and Neumaier (2008), Lee, Gramacy, Linkletter and Gray (2011), Chen and Kelley (2016), Porcelli and Toint (2017) and Müller and Day (2019).

7.1. Algebraic constraints

When all constraints are algebraically available, we can characterize the ordinal feasible region by a collection of inequality constraints:

$$\boldsymbol{\Omega} = \{\boldsymbol{x} \in \mathbb{R}^n : c_i(\boldsymbol{x}) \leq 0, \text{ for all } i \in I\}, \tag{7.1}$$

where each $c_i : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ and the set I is finite for all of the methods discussed. Problems with semi-infinite constraints can be addressed by using structured approaches as in Section 5.4. In this setting, we define the constraint function $\boldsymbol{c} : \mathbb{R}^n \to (\mathbb{R} \cup \{\infty\})^{|I|}$, where the ith entry of the vector $\boldsymbol{c}(\boldsymbol{x})$ is given by $c_i(\boldsymbol{x})$. Equality constraints can be represented in (7.1) by including both $c_i(\boldsymbol{x})$ and $-c_i(\boldsymbol{x})$; however, this practice should be avoided since it can hamper both theoretical and empirical performance.

7.1.1. Relaxable algebraic constraints

Relaxable algebraic constraints are the constraints that are typically treated in derivative-based non-linear optimization. We will organize our discussion into three primary types of methods: penalty approaches, filter approaches, and approaches with subproblems that employ models of the constraint functions.

Penalty approaches. Given constraints defined by (7.1), it is natural in the setting of relaxable constraints to quantify the violation of the ith constraint via the value of $\max\{0, c_i(\boldsymbol{x})\}$. In fact, given a penalty parameter $\rho > 0$, a common approach in relaxable constrained optimization is to replace the minimization of $f(\boldsymbol{x})$ with the minimization of a merit function such as

$$f(\boldsymbol{x}) + \frac{\rho}{2}\sum_{i\in I}\max\{0, c_i(\boldsymbol{x})\}. \tag{7.2}$$

The merit function in (7.2) is typically called an *exact penalty function*, because for a sufficiently large (but finite) value of $\rho > 0$, every local minimum $\boldsymbol{x}_*$ of (CON) is also a local minimum of the merit function in (7.2). We note that each summand $\max\{0, c_i(\boldsymbol{x})\}$ is generally non-smooth; the summand is still convex provided $c_i(\boldsymbol{x})$ is convex. Through the mapping

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} f(\boldsymbol{x}) \\ \boldsymbol{c}(\boldsymbol{x}) \end{bmatrix},$$

functions of the form (7.2) can be seen as cases of the composite non-smooth function (5.5) and are hence amenable to the methods discussed in Section 5.3. In contrast to this non-smooth approach, a more popular merit function historically has been the *quadratic penalty function*,

$$f(\boldsymbol{x}) + \rho\sum_{i\in I}\max\{0, c_i(\boldsymbol{x})\}^2. \tag{7.3}$$

However, the merit function in (7.3) lacks the same exactness guarantees that come with (7.2); even as ρ grows arbitrarily large, local minima of (CON) need not correspond in any way with minima of (7.3).

Another popular means of maintaining the smoothness (and convexity, when applicable) of (7.3) but regaining the exactness of (7.2) is to consider Lagrangian-based merit functions. Associating multipliers λ_i with each of the constraints in (7.1), the Lagrangian of (CON) is

$$L(\boldsymbol{x};\boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i\in I}\lambda_i\, c_i(\boldsymbol{x}). \tag{7.4}$$

Combining (7.4) with (7.3) yields the *augmented Lagrangian* merit function

$$L_A(\boldsymbol{x};\boldsymbol{\lambda};\rho) = f(\boldsymbol{x}) + \sum_{i\in I}\lambda_i\, c_i(\boldsymbol{x}) + \frac{\rho}{2}\sum_{i\in I}\max\{0, c_i(\boldsymbol{x})\}^2 \tag{7.5}$$

with the desired properties; that is, for non-negative $\boldsymbol{\lambda}$ and ρ, $L_A(\boldsymbol{x};\boldsymbol{\lambda};\rho)$ is smooth (convex) provided that $\boldsymbol{c}$ is.

In all varieties of these methods, which we broadly refer to as penalty approaches, the parameter ρ is dynamically updated between iterations. Methods typically increase ρ in order to promote feasibility; penalty methods tend to approach solutions from outside of $\boldsymbol{\Omega}$ and hence typically assume that the penalized constraints are relaxable. For a review of general penalty approaches, see Fletcher (1987, Chapter 12).

Lewis and Torczon (2002) adapt the augmented Lagrangian approach of Conn, Gould and Toint (1991) in one of the first proofs that DDS methods can be globally convergent for non-linear optimization. They utilize pattern search (see the discussion in Section 2.1.2) to approximately minimize the augmented Lagrangian function (7.5) in each iteration of their method. That is, each iteration of their method solves a subproblem

$$\underset{\boldsymbol{x}}{\text{minimize}}\{L_A(\boldsymbol{x};\boldsymbol{\lambda};\rho) : \boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u}\}. \tag{7.6}$$

Lewis and Torczon (2002) prove global convergence of their method to first-order Karush–Kuhn–Tucker (KKT) points. We note that the algebraic availability of bound constraints is explicitly used in (7.6). Other constraints could be algebraic or simulation-based because the method used to approximately solve (7.6) does not require availability of the derivative $\nabla_{\boldsymbol{x}} L_A(\boldsymbol{x};\boldsymbol{\lambda};\rho)$. The approach of Lewis and Torczon (2002) is expanded by Lewis and Torczon (2010), who demonstrate the benefits of treating linear constraints (including bound constraints) outside of the augmented Lagrangian merit function. That is, they consider subproblems of the form

$$\underset{\boldsymbol{x}}{\text{minimize}}\{L_A(\boldsymbol{x};\boldsymbol{\lambda};\rho) : \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}\}. \tag{7.7}$$

Bueno, Friedlander, Martínez and Sobral (2013) propose an inexact restoration method for problems (CON) where $\boldsymbol{\Omega}$ is given by equality constraints. The inexact restoration method alternates between improving feasibility (measured through the constraint violation $\|\boldsymbol{c}(\boldsymbol{x})\|_2$ in this equality-constrained case) and then approximately minimizing a $\|\cdot\|_2$-based exact penalty function before dynamically adjusting the penalty parameter ρ. Because of the separation of the feasibility and optimality phases of the inexact restoration method, the feasibility phase requires no evaluations of f. This feasibility phase is easier when constraint functions are available algebraically because (sub)derivative-based methods can be employed. Bueno *et al.* (2013) prove global convergence to first-order KKT points of this method under appropriate assumptions.

Amaioua, Audet, Conn and Le Digabel (2018) study the performance of a search step in MADS when solving (CON). One of their approaches uses the exact penalty (7.2), a second approach uses the augmented Lagrangian (7.5) and a third combines these two.

Audet, Le Digabel and Peyrega (2015) show that the convergence properties of MADS extend to problems with linear equality constraints. They explicitly address these algebraic constraints by reformulating the original problem into a new problem without equality constraints (and possibly fewer variables); other constraints are treated as will be discussed in Section 7.2.

Filter approaches. Whereas a penalty approach combines an objective function f and a measure of constraint violation into a single merit function to be minimized approximately, a *filter method* can be understood as a biobjective method minimizing the objective and the constraint violation simultaneously. For this general discussion, we will refer to the measure of constraint violation as $h(\boldsymbol{x})$. For example, in (7.2),

$$h(\boldsymbol{x}) = \sum_{i \in I} \max\{0, c_i(\boldsymbol{x})\}.$$

From the perspective of biobjective optimization, a *filter* can be understood as a subset of non-dominated points in the (f, h) space. A two-dimensional point $(f(\boldsymbol{x}_l), h(\boldsymbol{x}_l))$ is non-dominated, in the finite set of points $\{\boldsymbol{x}_j : j \in J\}$ evaluated by a method, provided there is no $j \in J \setminus \{l\}$ with

$$f(\boldsymbol{x}_j) \le f(\boldsymbol{x}_l) \quad \text{and} \quad h(\boldsymbol{x}_j) \le h(\boldsymbol{x}_l).$$

Unlike biobjective optimization, however, filter methods adaptively vary the subset of non-dominated points considered in order to identify feasible points (*i.e.* points where h vanishes). Different filter methods employ different mechanisms for managing the filter and generating new points.

Brekelmans, Driessen, Hamers and den Hertog (2005) employ a filter for handling relaxable algebraic constraints. Their model-based method attempts to have model-improving points satisfy the constraints. Ferreira, Karas, Sachine and Sobral (2017) extend the inexact-restoration method of Bueno *et al.* (2013) by replacing the penalty formulation with a filter mechanism and again prove global convergence to first-order KKT points.

Approaches with subproblems using modelled constraints. Another means of constraint handling is to construct local models m^{c_i} of each constraint c_i in (7.1). Given a local model m^f of the objective function f, such methods generally employ a sequence of subproblems of the form

$$\underset{\boldsymbol{s}}{\operatorname{minimize}}\{m^f(\boldsymbol{s}) : c_i(\boldsymbol{x}+\boldsymbol{s}) \le 0, \text{ for all } i \in I\}. \tag{7.8}$$

As an example approach, sequential quadratic programming (SQP) methods are popular derivative-based methods that employ a quadratic model of the objective function and linear models of the constraint functions. Several derivative-free approaches of this form exist, which we detail in this section. We mention that many of these approaches will generally impose an additional trust-region constraint (*i.e.* $\|\boldsymbol{s}\| \le \Delta$) on (7.8). As in Section 2.2.4, this trust-region constraint often has the additional role of monitoring the quality of the model m^f. Furthermore, such a trust-region constraint ensures that whenever $\boldsymbol{s} = \boldsymbol{0}$ is feasible for (7.8), the feasible region of (7.8) is compact.

Conn, Scheinberg and Toint (1998) consider an adaptation of a model-based trust-region method to constrained problems with differentiable algebraic constraints treated via the trust-region subproblem (7.8). They target problems where they deem the algebraic constraints to be ‘easy’, meaning that the resulting trust-region subproblem is not too difficult to solve. This method is implemented in the solver DFO (Conn *et al.* 2001).

The CONDOR method of Vanden Berghen (2004) and Vanden Berghen and Bersini (2005) extends the unconstrained UOBYQA method of Powell (2002) to address algebraic constraints. The trust-region subproblem considered takes the form

$$\underset{\boldsymbol{x}}{\operatorname{minimize}}\{m^f(\boldsymbol{x}) : c_i(\boldsymbol{s}) \leq 0, \text{ for all } i \in I'; \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}; \|\boldsymbol{s}\| \leq \Delta\}, \tag{7.9}$$

where m^f is a quadratic model and $I' \subseteq I$ captures the non-linear constraints in (7.1). In solving (7.9), the linear constraints are enforced explicitly and the non-linear constraints are addressed via an SQP approach. As will be discussed in Section 7.1.2, this corresponds to the linear constraints being treated as unrelaxable.

The LINCOA model-based method of Powell (2015) addresses linear inequality constraints. The LINCOA trust-region subproblem, which can be seen as (7.9) with $I' = \emptyset$, enforces the linear constraints via an active set approach. The active set decreases the degrees of freedom in the variables by restricting $\boldsymbol{x}$ to an affine subspace. Numerically efficient conjugate gradient and Krylov methods are proposed for working in the resulting subspace. Although considerable care is taken to have most points satisfy the linear constraints $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$, these constraints are ultimately treated as relaxable, since the method does not enforce these constraints when attempting to improve the quality of the model m^f.

Conejo *et al.* (2013) propose a trust-region algorithm when $\boldsymbol{\Omega}$ is closed and convex. They assume that it is easy to compute the projection onto $\boldsymbol{\Omega}$, which facilitates enforcement of the constraints via the trust-region subproblem (7.8). This approach is extended to include more general forms of $\boldsymbol{\Omega}$ by Conejo, Karas and Pedroso (2015). As with LINCOA, although subproblem solutions are feasible, the constraints are treated as relaxable since they may be violated in the course of improving the model m^f.

Martínez and Sobral (2012) propose a feasibility restoration method intended for problems with inequality constraints where the feasible region is ‘thin’: for example, if $\boldsymbol{\Omega}$ is defined by both $c_i(\boldsymbol{x}) \leq 0$ and $-c_i(\boldsymbol{x}) \leq 0$ for some i. Each iteration contains two steps: one that seeks to minimize the objective and one that seeks to decrease infeasibility using many evaluation of the constraint functions (without evaluating the objective). Similar to the progressive-barrier method discussed in Section 7.2, the method by Martínez and Sobral (2012) dynamically updates a tolerable level of infeasibility.

7.1.2. Unrelaxable algebraic constraints

We now address the case when all of the constraints are available algebraically but an unrelaxable constraint also exists. In this setting, such unrelaxable constraints are typically necessary to ensure meaningful output of a black-box objective function. Consequently, methods must always maintain feasibility (or at least establish feasibility and then maintain it) with respect to the unrelaxable constraints.

An early example of a method for unrelaxable constraints is the ‘complex’ method of Box (1965). This extension of the simplex method of Spendley *et al.* (1962) treats unrelaxable bound constraints by modifying the simplex operations to project into the interior of any potentially violated bound constraint. May (1974, 1979) extends the unconstrained derivative-free method of Mifflin (1975) to address unrelaxable linear constraints. The method of May (1979) uses finite-difference estimates, but care is taken to ensure that the perturbed points never violate the constraints.

As seen in Section 7.1.1, several approaches treat non-linear algebraic constraints via a merit function and enforce unrelaxable linear constraints via a constrained subproblem. These include the works of Lewis and Torczon (2002) for bound constraints in (7.6), Lewis and Torczon (2010) for inequality constraints in (7.7), and Vanden Berghen (2004) for inequality constraints in (7.9). Another merit function relevant for unrelaxable constraints is the extended-value merit function

$$h(\boldsymbol{x}) = f(\boldsymbol{x}) + \infty\, \delta_{\boldsymbol{\Omega}^C}(\boldsymbol{x}), \tag{7.10}$$

where $\delta_{\boldsymbol{\Omega}^C}$ is the indicator function of $\boldsymbol{\Omega}^C$. Such an *extreme-barrier* approach (see *e.g.* the discussion by Lewis and Torczon 1999) is particularly relevant for simulation-based constraints. Hence, with the exception of explicit treatment of unrelaxable algebraic constraints, we postpone significant discussion of extreme-barrier methods until Section 7.2.

DDS methods for unrelaxable algebraic constraints. Within DDS methods, an intuitive approach to handling unrelaxable constraints is to limit poll directions $\boldsymbol{D}_k$ so that $\boldsymbol{x}_k + \boldsymbol{d}_k$ is feasible with respect to the unrelaxable constraints. Lewis and Torczon (1999) and Lucidi and Sciandrone (2002*a*), respectively, develop pattern-search and coordinate-search methods for unrelaxable bound-constrained problems. By modifying the polling directions Lewis and Torczon (2000) show that pattern-search methods are also convergent in the presence of unrelaxable linear constraints. Chandramouli and Narayanan (2019) address unrelaxable bound constraints within a DDS method that employs a model-based method in the search step in addition to a bound-constrained line-search step. Kolda, Lewis and Torczon (2006) develop and analyse a new condition, related to the tangent cone of nearby active constraints, on the sets of directions used within a generating set search method when solving linearly constrained problems. The condition ensures that evaluated points are guaranteed to satisfy the linear constraints. Lucidi, Sciandrone and Tseng (2002) propose feasible descent methods that sample the objective over a finite set of search directions. Each iteration considers a set of ϵ-active constraints (*i.e.* those constraints for which $c_i(\boldsymbol{x}_k) \geq -\epsilon$) for general algebraic inequality constraints. Poll steps are projected in order to ensure they are feasible with respect to these ϵ-active constraints. The analysis of Lucidi *et al.* (2002) extends that of Lewis and Torczon (2000) and establishes convergence to a first-order KKT point under standard assumptions.

As introduced in Section 7.1.1, Audet *et al.* (2015) reformulate optimization problems with unrelaxable linear equality constraints in the context of MADS.

Gratton, Royer, Vicente and Zhang (2019*b*) extend the randomized DDS method of Gratton *et al.* (2015) to linearly constrained problems; candidate points are accepted only if they are feasible. Gratton *et al.* (2019*b*) establish probabilistic convergence and complexity results using a stationary measure appropriate for linearly constrained problems.

Model-based methods for unrelaxable algebraic constraints. Model-based methods are more challenging to design in the presence of unrelaxable constraints because enforcing guarantees of model quality such as those in (2.9) can be difficult. For a fixed value of $\boldsymbol{\kappa}$ in (2.9), it may be impossible to obtain a $\boldsymbol{\kappa}$-fully linear model using only feasible points. As an example, consider two linear constraints for which the angle between the constraints is too small to allow for $\boldsymbol{\kappa}$-fully linear model construction; avoiding interpolation points drawn from such thin regions motivated development of the wedge-based method of Marazzi and Nocedal (2002) from Section 2.2.4.

Powell (2009) proposes BOBYQA, a model-based trust-region method for bound-constrained optimization without derivatives that extends the unconstrained method NEWUOA in Powell (2006). BOBYQA ensures that all points at which f is evaluated satisfy the bound constraints. Arouxét, Echebest and Pilotta (2011) modify BOBYQA to use an active-set strategy in solving the bound-constrained trust-region subproblems; an $\|\cdot\|_\infty$-trust region is employed so that these subproblems correspond to minimization of a quadratic over a compact, bound-constrained domain. Wild (2008*a*, Section 6.3) develops an RBF-model-based method for unrelaxable bound constraints by enforcing the bounds during both model improvement and $\|\cdot\|_\infty$-trust-region subproblems. Gumma, Hashim and Ali (2014) extend the NEWUOA method to address linearly constrained problems. The linear constraints are enforced both when solving the trust-region subproblem and when seeking to improve the geometry of the interpolation points. Gratton, Toint and Tröltzsch (2011) propose a model-based method for unrelaxable bound-constrained optimization, which restricts the construction of fully linear models to subspaces defined by nearly active constraints. Working in such a reduced space means that the machinery for unconstrained models in Section 2.2.4 again applies.

Methods for problems with unrelaxable discrete constraints. Constraints that certain variables take discrete values are often unrelaxable in derivative-free optimization. For example, a black-box simulation may be unable to assign meaningful output when input variables take non-integer values. That such integer constraints are unrelaxable presents challenges distinct from those typically arising in mixed-integer non-linear optimization (Belotti *et al.* 2013).

Naturally, researchers have modified derivative-free methods for continuous optimization to address integer constraints. Audet and Dennis, Jr (2000) and Abramson, Audet, Chrissis and Walston (2009*a*), respectively, propose integer-constrained pattern-search and MADS methods to ensure that evaluated points respect integer constraints. Abramson, Audet and Dennis, Jr (2007) develop a pattern-search method that employs a filter that handles general inequality constraints and ensures that integer-constrained variables are always integer.

Porcelli and Toint (2017) propose the 'brute-force optimizer' BFO, a DDS method for mixed-variable problems (including those with ordinal categorical variables) that aligns the poll points to respect the discrete constraints. A recursive call of the method reduces the number of discrete variables by fixing a subset of these variables.

Liuzzi, Lucidi and Rinaldi (2011) solve mixed-integer bound-constrained problems by using both a local discrete search (to address integer variables) and a line search (for continuous variables). This approach is extended by Liuzzi, Lucidi and Rinaldi (2015) to also address mixed-integer problems with general constraints using the SQP approach from Liuzzi, Lucidi and Sciandrone (2010). Liuzzi, Lucidi and Rinaldi (2018) solve constrained integer optimization problems by performing non-monotone line searches along feasible *primitive directions* $\boldsymbol{D}$ in a neighbourhood of the current point $\boldsymbol{x}_k$. Feasible primitive directions are those $\boldsymbol{d} \in \mathbb{Z}^n \cap \boldsymbol{\Omega}$ satisfying $\mathrm{GCD}(\boldsymbol{d}_1, \dots, \boldsymbol{d}_{|\boldsymbol{D}|}) = 1$, that is, directions in a bounded neighbourhood that are not integer multiples of one another.

The method by Rashid, Ambani and Cetinkaya (2012) for mixed-integer problems builds multiquadric RBF models. Candidate points are produced by using gradient-based mixed-integer optimization techniques; the authors' relaxation-based approach employs a 'proxy model' that coincides with function values from points satisfying the unrelaxable integer constraints. The methods of Müller, Shoemaker and Piché (2013*a*, 2013*b*) and Müller (2016) similarly employ a global RBF model over the integer lattice, with various strategies for generating trial points based on this model. Newby and Ali (2015) build on BOBYQA to address bound-constrained mixed-integer problems. They outline an approach for building interpolation models of objectives using only points that are feasible. Their trust-region subproblems consist of minimizing a quadratic objective subject to bound and integer constraints.

Many of the discussed methods have been shown to converge to points that are mesh-isolated local solutions; see Newby and Ali (2015) for discussion of such 'local minimizers'. When an objective is convex, one can do better. Larson, Leyffer, Palkar and Wild (2019) propose a method for certifying a global minimum of a convex objective f subject to unrelaxable integer constraints. They form a piecewise linear underestimator by interpolating f through subsets of $n+1$ affinely independent points. The resulting underestimator is then used to generate new candidate points until global optimality has been certified.

7.2. Simulation-based constraints

As opposed to the preceding section, methods in this section are not limited to constraints that have closed-form solutions but also address constraints that depend on the output from some calculated function. Many methods address such simulation-based constraints by using approaches similar to those used for algebraic constraints.

Filter approaches. Filter methods for simulation-based constraints, as with algebraic constraints, seek to simultaneously decrease the objective and constraint violation. For example, Audet and Dennis, Jr (2004) develop a pattern-search method for general constrained optimization that accepts steps that improve either the objective or some measure of violation of simulation-based constraints. Their hybrid approach applies an extreme barrier to points that violate linear or bound constraints. Audet (2004) provides examples where the method by Audet and Dennis, Jr (2004) does not converge to stationary points.

Pourmohamad (2016) models objective and constraint functions using Gaussian process models in a filter-based method. Because these models are stochastic, point acceptability is determined by criteria such as probability of filter acceptability or expected area of dominated region (in the filter space).

Echebest, Schuverdt and Vignau (2015) develop a derivative-free method in the inexact feasibility restoration filter method framework of Gonzaga, Karas and Vanti (2004). Echebest *et al.* (2015) employ fully linear models of the objective and constraint functions and show that the resulting limit points are first-order KKT points.

Penalty approaches. The original MADS method (Audet and Dennis, Jr 2006) converts constrained problems to unconstrained problems by using the extreme-barrier approach mentioned above; that is, the merit function (7.10) effectively assigns a value of infinity to points that violate any constraint. A similar approach to general constraints is used by the 'complex' method of Box (1965); the simplex (complex) is updated to maintain the feasibility of the vertices of the simplex. As a consequence of its generality, the extreme-barrier approach is applicable for algebraic constraints, simulation-based constraints and even hidden constraints. Furthermore, because (7.10) is independent of the degree of both constraint satisfaction and constraint violation, the extreme barrier is able to address non-quantifiable constraints.

In contrast, the progressive-barrier method by Audet and Dennis, Jr (2009) employs a quadratic constraint penalty similar to (7.3) for relaxable simulation-based constraints $\{c_i : i \in I_r\}$ and an extreme-barrier penalty for unrelaxable simulation-based constraints $\{c_i : i \in I_u\}$. Their progressive-barrier method maintains a non-increasing threshold value ϵ_k that quantifies the allowable relaxable constraint violation in each iteration. Their approach effectively uses the merit function

$$h_k(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \boldsymbol{\Omega}_u \text{ and } \sum_{i\in I_r} \max\{0, c_i(\boldsymbol{x})\}^2 < \varepsilon_k, \\ \infty & \text{otherwise,} \end{cases} \tag{7.11}$$

where $\boldsymbol{\Omega}_u = \{\boldsymbol{x} : c_i(\boldsymbol{x}) \le 0,\ \forall i \in I_u\}$ denotes the feasible domain with respect to the unrelaxable constraints. The progressive-barrier method maintains a set of feasible and infeasible incumbent points and seeks to decrease the threshold ϵ_k to 0 based on the value of (7.11) at infeasible incumbent points. Trial steps are accepted as incumbents based on criteria resembling, but distinct from, those used by filter methods. Convergence to Clarke stationary points is obtained for particular sequences of the incumbent points. The NOMAD (Le Digabel 2011) implementation of MADS allows users to choose to address inequality constraints handled via extreme-barrier, progressive-barrier or filter approaches.

Also within the DDS framework, Gratton and Vicente (2014) use an extreme-barrier approach to handle unrelaxable constraints and an exact penalty function to handle the relaxable constraints. That is, step acceptability is based on satisfaction of the unrelaxable constraints as well as sufficient decrease in the merit function (7.2), with the set I containing only those constraints that are relaxable. As the algorithm progresses, relaxable constraints are transferred to the set of constraints treated by the extreme barrier; this approach is similar to that underlying the progressive-barrier approach.

Liuzzi and Lucidi (2009) and Liuzzi *et al.* (2010) consider line-search methods that apply a penalty to simulation-based constraints; Liuzzi and Lucidi (2009) employ an exact penalty function (a smoothed version of

$\| \cdot \|_\infty$), whereas Liuzzi *et al.* (2010) employ a sequence of quadratic penalty functions of the form (7.3). Fasano, Liuzzi, Lucidi and Rinaldi (2014) propose a similar line-search approach to address constraint and objective functions that are not differentiable.

Primarily concerned with equality constraints, Sampaio and Toint (2015, 2016) propose a derivative-free variant of trust-funnel methods, a class of methods proposed by Gould and Toint (2010) that avoid the use of both merit functions and filters.

Diniz-Ehrhardt, Martínez and Pedroso (2011) propose a method that models objective and constraint functions in an augmented Lagrangian framework. Similarly, Picheny, Gramacy, Wild and Le Digabel (2016) use an augmented Lagrangian framework, wherein the merit function in (7.5) uses Gaussian process models of the objective and constraint functions in place of the actual objective and constraint functions.

Approaches with subproblems using modelled constraints. In early work, Glass and Cooper (1965) develop a coordinate-search method that also uses linear models of the objective and constraint functions. On each iteration, after the coordinate directions are polled, the models are used in a linear program to generate new points; points are accepted only if they are feasible. Extending this idea, Powell (1994) develops the constrained optimization by linear approximation (COBYLA) method, which builds linear interpolation models of the objective and constraint functions on a common set of $n+1$ affinely independent points. Care is taken to maintain the non-degeneracy of this simplex. The method can handle both inequality and equality constraints, with candidate points obtained from a linearly constrained subproblem and then accepted based on a merit function of the form (7.2).

Bűrmen, Olenšek and Tuma (2015) propose a variant of MADS with a specialized model-based search step

$$\underset{\boldsymbol{x}}{\text{minimize}}\{m^f(\boldsymbol{x}) : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}, \tag{7.12}$$

where m^f is a strongly convex quadratic model of f and $(\boldsymbol{A}, \boldsymbol{b})$ are determined from linear regression models of the constraint functions. Both the search and poll steps are accepted only if they are feasible; this corresponds to the method effectively treating the constraints with an extreme-barrier approach.

Gramacy and Le Digabel (2015) extend the MADS framework by using treed Gaussian processes to model both the objective and simulation-based constraint functions. The resulting models are used both within the search step and to order the poll points (within an opportunistic polling paradigm) using a filter-based approach.

A number of methods work with restrictions of the domain $\boldsymbol{\Omega}$ in order to promote feasibility (typically with respect to the simulation-based constraints) of the generated points. Such strategies are often motivated by a desire to avoid the situation where feasibility is established only asymptotically. An example of such a restricted domain is the set

$$\boldsymbol{\Omega}_{\text{res}}(\boldsymbol{\epsilon}) = \{\boldsymbol{x} \in \mathbb{R}^n : c_i(\boldsymbol{x}) \leq 0 \ \forall i \in I_a,\ m^{c_i}(\boldsymbol{x}) + \epsilon_i(\boldsymbol{x}) \leq 0 \ \forall i \in I_s\}, \quad (7.13)$$

where algebraic constraints (corresponding to $i \in I_a$) are explicitly enforced and a parameter (or function of $\boldsymbol{x}$) $\boldsymbol{\epsilon}$ controls the degree of restriction for the modelled simulation-based constraints (corresponding to $i \in I_s$).

The methods of Regis (2013) utilize interpolating radial basis function surrogates of the objective and constraint functions. Acceptance of infeasible points is allowed and is followed by a constraint restoration phase that minimizes a quadratic penalty based on the modelled constraint violation. When the current point is feasible, a subproblem is solved with a feasible set defined by (7.13) in addition to a constraint that lower-bounds the distance between the trial point and the current point. Each parameter ϵ_i is adjusted based on the feasibility of constraint $i \in I_s$ in recent iterations.

Augustin and Marzouk (2014) develop a trust-region method employing fully linear models of both constraint and objective functions. They introduce a *path augmentation* scheme intended to locally convexify the simulation-based constraints. Their trust-region subproblem at the current point $\boldsymbol{x}_k$ minimizes the model of the objective function subject to a trust-region constraint and the restricted feasible set (7.13), where $\epsilon_i(\boldsymbol{x}) = \epsilon_0 \|\boldsymbol{x} - \boldsymbol{x}_k\|^{2/(1+p)}$ and where $\epsilon_0 > 0$ and $p \in (0, 1)$ are fixed constants. Augustin and Marzouk (2014) establish convergence of their method from feasible starting points; that is, they show a first-order criticality measure asymptotically tends to 0. Augustin and Marzouk (2014) produce a code, NOWPAC, that employs minimum-Frobenius norm quadratic models of both the objective and constraint functions. This work is extended by Augustin and Marzouk (2017) to the stochastic optimization problem (STOCH).

Whereas Augustin and Marzouk (2014) consider a local convexification of inequality constraints through the addition of a convex function to the constraint models, Regis and Wild (2017) consider a similar model-based approach but define an envelope around models of nearly active constraints. In particular, at the current point $\boldsymbol{x}_k$, the restricted feasible set (7.13) uses the parameter

$$\epsilon_i(\boldsymbol{x}) = \begin{cases} 0 & \text{if } c_i(\boldsymbol{x}_k) > -\xi_i, \\ \xi_i & \text{if } c_i(\boldsymbol{x}_k) \leq -\xi_i, \end{cases}$$

where $\{\xi_i : i \in I_s\}$ is fixed. This form of $\boldsymbol{\epsilon}$ ensures that trust-region subproblems remain non-empty and avoids applying a restriction when the algorithm is sufficiently close to the level set $\{\boldsymbol{x} : c_i(\boldsymbol{x}) = 0\}$.

Tröltzsch (2016) considers an SQP method in the style of Omojokun (1989), which applies a two-step process that first seeks to improve a measure of constraint violation and then solves a subproblem restricted to the null space of modelled constraint gradients. Tröltzsch (2016) uses linear models of the constraint functions and quadratic models of the objective function, with these models replacing $\boldsymbol{c}$ and f in the augmented Lagrangian merit function in (7.5). Step acceptance uses a merit function (an exact penalty function).

Müller and Woodbury (2017) develop a method for addressing computationally inexpensive objectives while satisfying computationally expensive constraints. Their two-phase method first seeks feasibility by solving a multi-objective optimization problem (a problem class that is the subject of Section 8.4) in which the constraint violations are minimized simultaneously; the second phase seeks to reduce the objective subject to constraints derived from cubic RBF models of the constraint functions.

Bajaj, Iyer and Hasan (2018) propose a two-phase method. In the feasibility phase, a trust-region method is applied to a quadratic penalty function that employs models of the simulation-based constraints. The trust-region subproblem at iteration k takes the form

$$\underset{\boldsymbol{x}}{\text{minimize}}\left\{\sum_{i\in I_s}\max\{0, m^{c_i}(\boldsymbol{x})\}^2 : c_i(\boldsymbol{x})\leq 0\ \forall i\in I_a,\ \boldsymbol{x}\in\mathcal{B}(\boldsymbol{x}_k;\Delta_k)\right\} \tag{7.14}$$

and thus explicitly enforces the algebraic constraints ($i\in I_a$) and penalizes violation of the modelled simulation-based constraints ($i\in I_s$). In the optimality phase, a trust-region method is applied to a model of the objective function, and the modelled constraint violation is bounded by that achieved in the feasibility phase; that is, the trust-region subproblem is

$$\begin{aligned}
\underset{\boldsymbol{x}}{\text{minimize}}\quad & m^f(\boldsymbol{x})\\
\text{subject to}\quad & c_i(\boldsymbol{x})\leq 0 \quad \text{for all } i\in I_a\\
& m^{c_i}(\boldsymbol{x})\leq c_i(\boldsymbol{x}_{\text{pen}}) \quad \text{for all } i\in I_s\\
& \boldsymbol{x}\in\mathcal{B}(\boldsymbol{x}_k;\Delta_k),
\end{aligned}$$

where $\boldsymbol{x}_{\text{pen}}$ is the point returned from the feasibility phase.

Hare and Lewis (2005) present an approach for approximating the normal and tangent cones; their approach is quite general and applies to the case when the domain is defined by non-quantifiable black-box constraints. Davis and Hare (2013) consider a simplex-gradient-based approach for approximating normal cones when the black-box constraints are quantifiable. Naturally, such approximate cones could be used to determine if a method's candidate solution approximately satisfies a stationarity condition.

8. Other extensions and practical considerations

We conclude with a cursory look at extensions of the methods presented, especially highlighting active areas of development.

8.1. Methods allowing for concurrent function evaluations

A number of the methods presented in this survey readily allow for the concurrent evaluation of the objective function at multiple points $\boldsymbol{x} \in \mathbb{R}^n$. Performing function evaluations concurrently through the use of parallel computing resources should decrease the wall-clock time required by a given method. Depending on the method, there is a natural limit to the amount of concurrency that can be utilized efficiently. Below we summarize such methods and their limits for concurrency.

The simplex methods discussed in Section 2.1 benefit from performing n concurrent evaluations of the objective when a shrink operation is performed. Also, the points corresponding to the expansion and reflection operations could be evaluated in parallel. Non-opportunistic directional direct-search methods are especially amenable to parallelization (Dennis, Jr and Torczon 1991) because the $|\boldsymbol{D}_k|$ poll directions can be evaluated concurrently.

Model-based methods from Section 2.2 can use concurrent evaluations during model building when, for example, evaluating up to $\dim(\mathcal{P}^{d,n})$ additional points for use in (2.14). In another example, CONDOR (Vanden Berghen and Bersini 2005, Vanden Berghen 2004) utilizes concurrent evaluations of the objective to replace points far away from the current trust-region centre by maximizing the associated Lagrange polynomial. The thesis by Olsson (2014) considers three ways of using concurrent resources within a model-based algorithm: using multiple starting points, evaluating different models in order to better predict a point's value, and generating multiple points with each model (*e.g.* solving with the trust-region subproblem with different radii). A similar approach of generating multiple trial points concurrently is employed in the parallel direct-search, trust-region method of Hough and Meza (2002).

Finite-difference-based approaches (*e.g.* Section 2.3) allow for n concurrent evaluations with forward differences (2.28) or $2n$ concurrent evaluations with central differences (2.29). Implicit filtering also performs such a central-difference calculation that can utilize $2n$ concurrent evaluations (line 5 of Algorithm 4). Line-search methods can evaluate multiple points concurrently during their line-search procedure. The methods of García-Palomares and Rodríguez (2002) and García-Palomares, García-Urrea and Rodríguez-Hernández (2013) also consider using parallel resources to concurrently evaluate points in a neighbourhood of interest.

When using a set of independently generated points, pure random search exhibits perfect scaling as the level of available concurrency increases. Otherwise, the randomized methods for deterministic objectives from Section 3 can utilize concurrent evaluations in a manner similar to that of their deterministic counterparts. Nesterov random search can use n or $2n$ concurrent objective evaluations when computing an approximate gradient in (3.1). Randomized DDS methods can concurrently evaluate $|\boldsymbol{D}_k|$ poll points, and randomized trust-region methods can concurrently evaluate points needed for building and improving models.

In addition to the above approaches for using parallel resources, methods from Section 5 for structured problems can use concurrent evaluations to calculate parts of the objective. For example, methods for optimizing separable objectives such as (5.1) or (5.3) can evaluate the p component functions F_i concurrently.

The various gradient approximations used by methods in Section 6 are amenable to parallelization in the same manner as previously discussed, but with the additional possibility of also evaluating at multiple $\boldsymbol{\xi}$ values. SA methods can use $2n$ concurrent evaluations of $\tilde{f}$ in calculating (6.2) or (6.4) and SPSA can use two concurrent evaluations when calculating (6.5). Methods employing the sample mean estimator (6.7) can utilize p_k evaluations concurrently.

8.2. Multistart methods

A natural approach for addressing non-convex objectives for which it is not known whether multiple local minima exist is to start a local optimization method from different points in the domain in the hope of identifying different local minima. Such multistart approaches also allow for the use of methods that are specialized for optimizing problems with known structure.

Multistart methods allow for the use of concurrent objective evaluations if two or more local optimization runs are being performed at the same time. Multistart methods also allow one to utilize additional computational resources; this ability is especially useful when an objective evaluation does not become faster with additional resources or when the local optimization method is inherently sequential.

Boender, Rinnooy Kan, Timmer and Stougie (1982) derive confidence intervals on the objective value of a global minimizer when starting a local optimization method at uniformly drawn points. Their analysis gives rise to the multilevel single linkage (MLSL) method (Rinnooy Kan and Timmer 1987*a*, Rinnooy Kan and Timmer 1987*b*). Iteration k of the method draws N points uniformly over the domain and starts a local optimization method from sampled points that do not have any other point within a specific distance, depending on k and N, with a smaller objective value. With this rule, and under assumptions on the distance between minimizers in $\boldsymbol{\Omega}$ and properties of the local optimization method used, MLSL is shown to almost surely identify all local minima while starting the local optimization method from only finitely many points. Larson and Wild (2016, 2018) generalize MLSL by showing similar theoretical results when starting-point selection utilizes points both from the random sampling and from those generated by local optimization runs.

If a meaningful variance exists in the objective evaluation times, batched evaluation of points may result in an inefficient use of computational resources. Such concerns have motivated the development of a number of methods including the HOPSPACK framework (Plantenga 2009), which supports the sharing of information between different local optimization methods. Shoemaker and Regis (2003) also use information from multiple optimization methods to determine points at which to evaluate the objective function. Similarly, the SNOBFIT method by Huyer and Neumaier (2008) uses concurrent objective evaluations while combining local searches in a global framework. The software focuses on robustness in addressing many practical concerns including soft constraints, hidden constraints, and a problem domain that is modified by the user as the method progresses.

Instead of coordinating concurrent instances of a pattern-search method, Audet, Dennis, Jr and Le Digabel (2008*a*) propose an implementation of MADS that decomposes the domain into subspaces to be optimized over in parallel. Alarie *et al.* (2018) study different approaches for selecting subsets of variables to define subproblems in such an approach. Custódio and Madeira (2015) maintain concurrent instances of a pattern-search method, and merge those instances that become sufficiently close. Taddy, Lee, Gray and Griffin (2009) use a global treed-Gaussian process to guide a local pattern-search method to encourage the identification of better local minima.

8.3. Other global optimization methods

Guarantees of global optimality for general continuous functions rely on candidate points being generated densely in the domain (Törn and Žilinskas 1989, Theorem 1.3); such candidate points can be generated in either a deterministic or randomized fashion. When f is Lipschitz-continuous on $\boldsymbol{\Omega}$ and the Lipschitz constant L_{f} is available to the optimization method, one need not generate points densely in the domain. In particular, if $\hat{\boldsymbol{x}}$ is an approximate minimizer of f and $\boldsymbol{x}$ is a point satisfying $f(\boldsymbol{x}) > f(\hat{\boldsymbol{x}})$, no global minimizer can lie in – and therefore no point needs to be sampled from – $\mathcal{B}(\boldsymbol{x}; (f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}}))/L_{\mathrm{f}})$. Naturally, the benefit of exploiting this fact requires accurate knowledge of the Lipschitz constant. One can empirically observe a lower bound on L_{f}, but obtaining useful upper bounds on L_{f} may not be possible. Methods that exploit this Lipschitz knowledge may suffer a considerable performance decrease when overestimating L_{f} (Hansen, Jaumard and Lu 1991).

Motivated by situations where the Lipschitz constant of f is unavailable, Jones, Perttunen and Stuckman (1993) develop the DIRECT (DIviding RECTangles) method. DIRECT partitions a bound-constrained $\mathbf{\Omega}$ into $2n+1$ hyper-rectangles (hence the method's name) with an evaluated point at the centre of each. Each hyper-rectangle is scored via a combination of the length of its longest side and the function value at its centre. This scoring favours hyper-rectangles exhibiting both long sides and small function values; the best-scoring hyper-rectangles are further divided. (As such, DIRECT's performance can be significantly affected by adding a constant value to the objective (Finkel and Kelley 2006).) DIRECT generates centres that are dense in $\mathbf{\Omega}$ and will therefore identify the global minimizers of f over $\mathbf{\Omega}$, even when f is non-smooth (Jones *et al.* 1993, Finkel and Kelley 2004, Finkel and Kelley 2009). Several versions of DIRECT that perform concurrent function evaluations take significant care to ensure the sequence of points generated is the same as that produced by DIRECT (He, Verstak, Sosonkina and Watson 2009*a*, He, Verstak, Watson and Sosonkina 2007, He, Verstak, Watson and Sosonkina 2009*b*, He, Watson and Sosonkina 2009*c*). Similar hyper-rectangle partitioning strategies are used by the methods of Munos (2011). The multilevel coordinate-search (MCS) method by Huyer and Neumaier (1999) is inspired by DIRECT in many ways. MCS maintains a partitioning of the domain and subdivides hyper-rectangles based on their size and value. MCS uses the function values at boundary points, rather than the centre points, to determine the value of a hyper-rectangle; such boundary points can be shared by more than one hyper-rectangle. Huyer and Neumaier (2008) show that a version of MCS needs to consider only finitely many hyper-rectangles before identifying a global minimizer.

Many randomized approaches for generating points densely in a domain $\mathbf{\Omega}$ have been developed. These include Bayesian optimization methods and related variants (Mockus 1989, Jones, Schonlau and Welch 1998, Frazier 2018), some of which have established complexity rates (Bull 2011). Such randomized samplings of $\mathbf{\Omega}$ can be used to produce a global surrogate; similar to other model-based methods, this global model can be minimized to produce points where the objective should be evaluated. Although minimizing such a global surrogate may be difficult, such a subproblem may be easier than the original problem, which typically entails a computationally expensive objective function for which derivatives are unavailable. Vu, D'Ambrosio, Hamadi and Liberti (2016) provide a recent survey of such surrogate-based methods for global optimization.

8.4. Methods for multi-objective optimization

Multi-objective optimization problems are typically stated as

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & \boldsymbol{F}(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in \boldsymbol{\Omega} \subset \mathbb{R}^n, \end{array} \tag{MOO}$$

where $p > 1$ objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \dots, p$ define the vector-valued mapping $\boldsymbol{F}$ via $\boldsymbol{F}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \dots, f_p(\boldsymbol{x})]$. Given potentially conflicting objectives $f_1, \dots, f_p$, the problem (MOO) is well-defined only when given an ordering on the vector of objective values $\boldsymbol{F}(\boldsymbol{x})$. Given distinct points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n$, $\boldsymbol{x}_1$ *Pareto dominates* $\boldsymbol{x}_2$ provided

$$f_i(\boldsymbol{x}_1) \le f_i(\boldsymbol{x}_2) \quad \text{for all } i = 1, \dots, p \quad \text{and} \quad f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2) \quad \text{for some } j.$$

The set of all feasible points that are not Pareto-dominated by any other feasible point is referred to as the *Pareto(-optimal) set of* (MOO). An in-depth treatment of such problems is provided by Ehrgott (2005).

Ideally, a method designed for the solution of (MOO) should return an approximation of the Pareto set. If at least one objective $f_1, \dots, f_p$ is non-convex, however, approximating the Pareto set can be challenging. Consequently, methods for multi-objective optimization typically pursue *Pareto stationarity*, which is a form of local optimality characterized by a first-order stationarity condition. If $\boldsymbol{\Omega} = \mathbb{R}^n$, a point $\boldsymbol{x}_*$ is a Pareto stationary point of $\boldsymbol{F}$ provided that for each $\boldsymbol{d} \in \mathbb{R}^n$, there exists $j \in \{1, \dots, p\}$ such that $f_j'(\boldsymbol{x}_*; \boldsymbol{d}) \ge 0$. This notion of stationarity is an extension of the one given for single-objective optimization in (2.1).

Typical methods for (MOO) return a collection of points that are not known to be Pareto-dominated and thus serve as an approximation to the set of Pareto points. From a theoretical point of view, most methods endeavour only to demonstrate that all accumulation points are Pareto stationary, and rarely prove the existence of more than one such point. From a practical point of view, comparing the approximate Pareto sets returned by a method for multi-objective optimization is not straightforward. For discussions of some comparators used in multi-objective optimization, see Knowles and Corne (2002) and Audet *et al.* (2018*a*).

Various derivative-free methods discussed in this survey have been extended to address (MOO). The method of Audet, Savard and Zghal (2008*b*) solves biobjective optimization problems by iteratively combining the two objectives into a single objective (for instance, by considering a weighted sum of the two objectives); MADS is then applied to this single-objective problem. Audet, Savard and Zghal (2010) extend the method of Audet *et al.* (2008*b*) to multi-objective problems with more than two objectives. Audet *et al.* (2008*b*, 2010) demonstrate that all refining points of the sequence of candidate points produced by these methods are Pareto stationary.

Custódio, Madeira, Vaz and Vicente (2011) propose *direct-multisearch* methods, a multi-objective analogue of direct-search methods. Like direct-search methods, direct-multisearch methods involve both a search step and a poll step. Direct-multisearch methods maintain a list of non-dominated points; at the start of an iteration, one non-dominated point must be selected to serve as the centre for a poll step. Custódio *et al.* (2011) demonstrate that at any accumulation point $\boldsymbol{x}_*$ of the maintained sequence of non-dominated points from a direct-multisearch method, it holds that for any direction $\boldsymbol{d}$ that appears in a poll step infinitely often, $f_j'(\boldsymbol{x}_*, \boldsymbol{d}) \geq 0$ for at least one j. In other words, accumulation points of the method are Pareto stationary when restricted to these directions $\boldsymbol{d}$. Custódio and Madeira (2016) incorporate these direct-multisearch methods within a multistart framework in an effort to find multiple Pareto stationary points and thus to better approximate the Pareto set.

For stochastic biobjective problems, Kim and Ryu (2011) employ sample average approximation to estimate $\boldsymbol{F}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[\tilde{\boldsymbol{F}}(\boldsymbol{x}; \boldsymbol{\xi})]$ and propose a model-based trust-region method. Ryu and Kim (2014) adapt the approach of Kim and Ryu (2011) to the deterministic biobjective setting. At the start of each iteration, these methods construct fully linear models of both objectives around a (currently) non-dominated point. These methods solve three trust-region subproblems – one for each of the two objectives, and a third that weights the two objectives as in Audet *et al.* (2008*b*) – and accept all non-dominated trial points. If both objectives are in $\mathcal{LC}^1$, Ryu and Kim (2014) prove that one of the three objectives satisfies a lim-inf convergence result of the form (2.5), implying the existence of a Pareto-stationary accumulation point.

Liuzzi, Lucidi and Rinaldi (2016) propose a method for constrained multi-objective non-smooth optimization that separately handles each objective and constraint via an exact penalty (see (7.2)) in order to determine whether a point is non-dominated. Given the non-sequential nature of how non-dominated points are selected, Liuzzi *et al.* (2016) identify and link the subsequences implied by a lim-inf convergence result. They show that limit points of these linked sequences are Pareto stationary provided the search directions used in each linked sequence are asymptotically dense in the unit sphere.

Cocchi, Liuzzi, Papini and Sciandrone (2018) extend implicit filtering to the multi-objective case. They approximate each objective gradient separately using implicit-filtering techniques; they combine these approximate gradients in a disciplined way to generate search directions. Cocchi *et al.* (2018) demonstrate that their method generates at least one accumulation point and that every such accumulation point is Pareto stationary.

8.5. *Methods for multifidelity optimization*

Multifidelity optimization concerns the minimization of a high-fidelity objective function $f = f_0$ in situations where a lower-fidelity version f_ϵ (for $\epsilon > 0$) also exists. Evaluations of the lower-fidelity function f_ϵ are less computationally expensive than are evaluations of f_0; hence, a goal in multifidelity optimization is to exploit the existence of the lower-fidelity f_ϵ in order to perform as few evaluations of f_0 as possible. An example of such a setting occurs when there exist multiple grid resolutions defining discretizations for the numerical solution of partial differential equations that defines f_0 and f_ϵ.

Polak and Wetter (2006) develop a pattern-search method that exploits the existence of multiple levels of fidelity. The method begins at the coarsest available level and then monotonically refines the level of fidelity (*i.e.* decreases ϵ) after a sufficient number of consecutive unsuccessful iterations occur.

A method that both decreases ϵ and increases ϵ (akin to the V- and W-cycles of multigrid methods (Xu and Zikatanov 2017)), is the multilevel method of Frandi and Papini (2013). The method follows the MG/Opt framework of Nash (2000) and instantiates runs of a coordinate-search method at specified fidelity and solution accuracy levels. Another multigrid-inspired method is developed by Liu, Zeng and Yang (2015), wherein a hierarchy of DIRECT runs are performed at varying fidelity and budget levels.

Model-based methods have also been extended to the multifidelity setting. For example, March and Willcox (2012) employ a fully linear RBF model to interpolate the error between two different fidelity levels. Their method then employs this model within a trust-region framework, but uses f_0 to determine whether to accept a given step.

Another model-based approach for multifidelity optimization is co-kriging; see, for example, Xiong, Qian and Wu (2013) and Le Gratiet and Cannamela (2015). In such approaches, a statistical surrogate (typically a Gaussian process model) is constructed for each fidelity level with the aim of modelling the relationships among the fidelity levels in areas of the domain relevant to optimization.

Derivative-free methods for multifidelity, multi-objective and concurrent/parallel optimization remain an especially open avenue of future research.

Acknowledgements

We are grateful to referees and colleagues whose comments greatly improved the manuscript; these include Mark Abramson, Charles Audet, Ana Luísa Custódio, Sébastien Le Digabel, Warren Hare, Paul Hovland, Jorge Moré, Raghu Pasupathy, Margherita Porcelli, Francesco Rinaldi, Clément Royer, Katya Scheinberg, Sara Shashaani, Aekaansh Verma and Zaikun Zhang. We are especially indebted to Gail Pieper and Glennis Starling for their invaluable editing. This material is based upon work supported by the applied mathematics and SciDAC activities of the Office of Advanced Scientific Computing Research, Office of Science, US Department of Energy, under Contract DE-AC02-06CH11357.

Appendix: Collection of WCC results

Table A.1 contains select WCC bounds for methods appearing in the literature. Given $\epsilon > 0$, all WCC bounds in this appendix are given in terms of N_ϵ, an upper bound on the number of *function evaluations* of a method to guarantee that the specified condition is met. We present results in this form because function evaluation complexity of derivative-free methods is often of greater interest than is iteration complexity. We present N_ϵ in terms of four parameters:

- the accuracy ϵ;
- the dimension n;
- the Lipschitz constant of the function L_{f}, the Lipschitz constant of the function gradient L_{g} or the Lipschitz constant of the function Hessian L_{H} (provided these constants are well-defined); and
- a measure of how far the starting point $\boldsymbol{x}_0$ is from a stationary point $\boldsymbol{x}_*$. In this appendix, this measure is either $f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*)$,

$$R_{\mathrm{level}} = \sup_{\boldsymbol{x}\in\mathbb{R}^n} \{\|\boldsymbol{x} - \boldsymbol{x}_*\| : f(\boldsymbol{x}) \le f(\boldsymbol{x}_0)\} \tag{8.1}$$

or

$$R_{\boldsymbol{x}} \ge \|\boldsymbol{x}_0 - \boldsymbol{x}_*\|. \tag{8.2}$$

We present additional constants in N_ϵ when particularly informative.

Naturally, each method in Table A.1 has additional algorithmic parameters that influence algorithmic behaviour. We have omitted the dependence of each method's WCC on the selection of algorithmic parameters to allow for an easier comparison of methods.

We recall that, with the exception of the methods from Nesterov and Spokoiny (2017) and Konečný and Richtárik (2014), the methods referenced in Table A.1 do not require knowledge of the value of the relevant Lipschitz constants.

Table A.1. Known WCC bounds on the number of function evaluations needed to achieve a given stationarity measure.

Rate type	Method type (citation)$^{\text{NOTES}}$	N_ϵ
$f \in \mathcal{LC}^1$		
$\\|\nabla f(\boldsymbol{x}_k)\\| \le \epsilon$	DDS (Konečný and Richtárik 2014)	$\dfrac{n^2 L_{\text{g}}(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^2}$
	TR (Garmanjani *et al.* 2016)	$\dfrac{n^2 L_{\text{g}}^2(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^2}$
	ARC-DFO (Cartis *et al.* 2012)$^{\text{A}}$	$\dfrac{n^2 \max\{L_{\text{H}}, L_{\text{g}}\}^{3/2}(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^{3/2}}$
$\mathbb{E}_{\boldsymbol{U}_{k-1}}[\\|\nabla f(\hat{\boldsymbol{x}}_k)\\|] \le \epsilon$	RS (Nesterov and Spokoiny 2017)$^{\text{B}}$	$\dfrac{n L_{\text{g}}(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^2}$
$\\|\nabla f(\boldsymbol{x}_k)\\| \le \epsilon$ w.p. $1 - p_1$	DDS (Gratton *et al.* 2015)$^{\text{C}}$	$\dfrac{mn L_{\text{g}}^2(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^2}$
$\\|\nabla f(\boldsymbol{x}_k)\\| \le \epsilon$ w.p. $1 - p_2$	TR (Gratton *et al.* 2018)$^{\text{C,D}}$	$\dfrac{m \max\{\kappa_{\text{ef}}, \kappa_{\text{eg}}\}^2(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^2}$
$f \in \mathcal{LC}^2$		
$\max\{\\|\nabla f(\boldsymbol{x}_k)\\|, -\lambda_k\} \le \epsilon$	DDS (Gratton *et al.* 2016)	$\dfrac{n^5 \max\{L_{\text{H}}, L_{\text{g}}\}^3(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^3}$
	TR (Gratton *et al.* 2019*a*)	$\dfrac{n^5 \max\{L_{\text{H}}^3, L_{\text{g}}^2\}(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^3}$
$\max\{\\|\nabla f(\boldsymbol{x}_k)\\|, -\lambda_k\} \le \epsilon$ w.p. $1 - p_3$	TR (Gratton *et al.* 2018)$^{\text{C,D}}$	$\dfrac{m \max\{\kappa_{\text{eg}}, \kappa_{\text{eH}}\}^3(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^3}$

Table A.1 continued.

Rate type	Method type (citation)$^{\text{NOTES}}$	N_ϵ
$f \in \mathcal{LC}^1$, f is λ-strongly convex		
$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_*) \leq \epsilon$	DDS (Konečný and Richtárik 2014)	$\dfrac{n^2 L_{\rm g}}{\lambda} \log\left(\dfrac{1}{\epsilon}\right)$
$\mathbb{E}_{\boldsymbol{U}_{k-1}}[f(\hat{\boldsymbol{x}}_k)] - f(\boldsymbol{x}_*) \leq \epsilon$	RS (Nesterov and Spokoiny 2017)$^{\rm B}$	$\dfrac{n L_{\rm g}}{\lambda} \log\left(\dfrac{L_{\rm g} {R_{\boldsymbol{x}}}^2}{\epsilon}\right)$
$f \in \mathcal{LC}^1$, f is convex		
$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_*) \leq \epsilon$	DDS (Konečný and Richtárik 2014)$^{\rm E}$	$\dfrac{n^2 L_{\rm g} R_{\rm level}}{\epsilon}$
$\mathbb{E}_{\boldsymbol{U}_{k-1}}[f(\hat{\boldsymbol{x}}_k)] - f(\boldsymbol{x}_*) \leq \epsilon$	RS (Nesterov and Spokoiny 2017)$^{\rm B}$	$\dfrac{n L_{\rm g} {R_{\boldsymbol{x}}}^2}{\epsilon}$
$f \in \mathcal{LC}^0$, f is convex		
$\mathbb{E}_{\boldsymbol{U}_{k-1}}[f(\hat{\boldsymbol{x}}_k)] - f(\boldsymbol{x}_*) \leq \epsilon$	RS (Nesterov and Spokoiny 2017)$^{\rm B}$	$\dfrac{n^2 L_{\rm f}^2 {R_{\boldsymbol{x}}}^2}{\epsilon^2}$
$f \in \mathcal{LC}^0$		
$\mathbb{E}_{\boldsymbol{U}_{k-1}}[\Vert \nabla f_{\bar{\mu}}(\hat{\boldsymbol{x}}_k) \Vert] \leq \epsilon, \bar{\mu} = \dfrac{\epsilon}{L_{\rm f}\sqrt{n}}$	RS (Nesterov and Spokoiny 2017)$^{\rm B}$	$\dfrac{n^3 L_{\rm f}^5 (f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^3}$

Table A.1 continued.

Rate type	Method type (citation)[NOTES]	N_ϵ
$f = h \circ \boldsymbol{F}$, *convex* $h \in \mathcal{LC}^0$, $\boldsymbol{F} \in \mathcal{LC}^1$		
$\Psi(\boldsymbol{x}_k) \leq \epsilon$	TR (Garmanjani *et al.* 2016)[F]	$\dfrac{pn^2 L_{\mathrm{g}}(\boldsymbol{F})^2 L_{\mathrm{f}}(h)^2 (f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^2}$
A smoothed $f_\mu(\boldsymbol{x})$ *for* f		
$\\|\nabla f_{\mu_k}(\boldsymbol{x}_k)\\| \leq \epsilon$ where $\mu_k \in O\left(\dfrac{\epsilon}{\sqrt{n}}\right)$	DDS (Garmanjani and Vicente 2012)[G]	$\dfrac{n^{5/2}\left[-\log(\epsilon) + \log(n)\right](f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^3}$
	TR (Garmanjani *et al.* 2016)[G]	$\dfrac{n^{5/2}\left[\lvert\log(\epsilon)\rvert + \log(n)\right](f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))}{\epsilon^3}$

A We omit an additional $|\log(\epsilon)|$ dependence.

B $\hat{\boldsymbol{x}}_k = \arg\min_{j=1,\dots,k} f(\boldsymbol{x}_j)$.

C m is the number of function evaluations performed in each iteration, independent of n.

D Gratton *et al.* (2018) prove results for an arbitrary model-building scheme that assumes the ability to yield p-probabilistically $\boldsymbol{\kappa}_Q$-fully quadratic models (where $\boldsymbol{\kappa}_Q = (\kappa_{\mathrm{ef}}, \kappa_{\mathrm{eg}}, \kappa_{\mathrm{eH}})$) when $f \in \mathcal{LC}^2$ and p-probabilistically $\boldsymbol{\kappa}_L$-fully linear models (where $\boldsymbol{\kappa}_L = (\kappa_{\mathrm{ef}}, \kappa_{\mathrm{eg}})$) when $f \in \mathcal{LC}^1$. The construction of probabilistically fully quadratic models or probabilistically fully linear models when $m \ll (n+1)(n+2)/2$ remains an open question. Note that when $p = 1$, it is known that by using $m \in O(n^2)$ points, one can guarantee $\boldsymbol{\kappa}_Q$-fully quadratic models with $\kappa_{\mathrm{ef}}, \kappa_{\mathrm{eg}}, \kappa_{\mathrm{eH}} \in O(nL_{\mathrm{H}})$ (Conn *et al.* 2008*a*, Theorem 3). In this case, the result of Gratton *et al.* (2018) yields a rate weaker than that obtained by Gratton *et al.* (2016) by a factor of L_{g}. Similarly, when $p = 1$, it is known that by using $m \in O(n)$ points, one can guarantee $\boldsymbol{\kappa}_L$-fully linear models with $\kappa_{\mathrm{ef}}, \kappa_{\mathrm{eg}} \in O(n^{1/2} L_{\mathrm{g}})$ (Conn *et al.* 2008*a*, Theorem 2). In this case, the result of Gratton *et al.* (2018) yields a rate comparable to that obtained by Garmanjani *et al.* (2016).

E Vicente (2013) derives the same bound but with L_{g}^2 instead of L_{g}; however, the method of Vicente (2013) does not require the value L_{g}.

F $L_{\mathrm{g}}(\boldsymbol{F})$ is the Lipschitz constant of the Jacobian $J(F)$, $L_{\mathrm{f}}(h)$ is the Lipschitz constant of h, and p is the dimension of the domain of h. A bound for a similar method with an additional $|\log(\epsilon)|$ dependence appears in Grapiglia *et al.* (2016).

G Lipschitz constants do not appear because they are 'cancelled' by choosing the rate at which smoothing parameter $\mu_k \to 0$.

In Table A.1, we employ the constants

$$p_1 = \exp\biggl(-\frac{nL_{\rm g}^2}{\epsilon^2}(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))\biggr),$$
$$p_2 = \exp\biggl(-\frac{\max\{\kappa_{\rm ef}, \kappa_{\rm eg}\}^2}{\epsilon^2}(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))\biggr),$$
$$p_3 = \exp\biggl(-\frac{\max\{\kappa_{\rm eg}, \kappa_{\rm eH}\}^3}{\epsilon^3}(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*))\biggr).$$

REFERENCES[12]

M. A. Abramson (2005), ‘Second-order behavior of pattern search’, *SIAM J. Optim.* **16**, 515–530.

M. A. Abramson and C. Audet (2006), ‘Convergence of mesh adaptive direct search to second-order stationary points’, *SIAM J. Optim.* **17**, 606–619.

M. A. Abramson, C. Audet and J. E. Dennis, Jr (2004), ‘Generalized pattern searches with derivative information’, *Math. Program.* **100**, 3–25.

M. A. Abramson, C. Audet, J. Chrissis and J. Walston (2009*a*), ‘Mesh adaptive direct search algorithms for mixed variable optimization’, *Optim. Lett.* **3**, 35–47.

M. A. Abramson, C. Audet, J. E. Dennis, Jr and S. Le Digabel (2009*b*), ‘OrthoMADS: A deterministic MADS instance with orthogonal directions’, *SIAM J. Optim.* **20**, 948–966.

M. A. Abramson, L. Frimannslund and T. Steihaug (2013), ‘A subclass of generating set search with convergence to second-order stationary points’, *Optim. Methods Software* **29**, 900–918.

M. Abramson, C. Audet and J. E. Dennis, Jr (2007), ‘Filter pattern search algorithms for mixed variable constrained optimization problems’, *Pacific J. Optim.* **3**, 477–500.

A. Agarwal, O. Dekel and L. Xiao (2010), Optimal algorithms for online convex optimization with multi-point bandit feedback. In *23rd Conference on Learning Theory (COLT 2010)* (A. T. Kalai and M. Mohri, eds), pp. 28–40.

A. Agarwal, D. P. Foster, D. J. Hsu, S. M. Kakade and A. Rakhlin (2011), Stochastic convex optimization with bandit feedback. In *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor *et al.*, eds), Curran Associates, pp. 1035–1043.

A. Agarwal, M. J. Wainwright, P. L. Bartlett and P. K. Ravikumar (2009), Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems 22* (Y. Bengio *et al.*, eds), Curran Associates, pp. 1–9.

[12] The URLs cited in this work were correct at the time of going to press, but the publisher and the authors make no undertaking that the citations remain live or are accurate or appropriate.

R. Agrawal (1995), ‘Sample mean based index policies with $O(\log n)$ regret for the multi-armed bandit problem’, *Adv. Appl. Probab.* **27**, 1054–1078.

S. Alarie, N. Amaioua, C. Audet, S. Le Digabel and L.-A. Leclaire (2018), Selection of variables in parallel space decomposition for the mesh adaptive direct search algorithm. Technical report, Cahier du GERAD G-2018-38, GERAD.

N. Alexandrov, J. E. Dennis, Jr, R. M. Lewis and V. Torczon (1998), ‘A trust region framework for managing the use of approximation models in optimization’, *Struct. Optim.* **15**, 16–23.

N. Amaioua, C. Audet, A. R. Conn and S. Le Digabel (2018), ‘Efficient solution of quadratically constrained quadratic subproblems within the mesh adaptive direct search algorithm’, *European J. Oper. Res.* **268**, 13–24.

S. Amaran, N. V. Sahinidis, B. Sharda and S. J. Bury (2015), ‘Simulation optimization: A review of algorithms and applications’, *Ann. Oper. Res.* **240**, 351–380.

H. L. Anderson (1986), ‘Scientific uses of the MANIAC’, *J. Statist. Phys.* **43**, 731–748.

M. B. Arouxét, N. Echebest and E. A. Pilotta (2011), ‘Active-set strategy in Powell’s method for optimization without derivatives’, *Comput. Appl. Math.* **30**, 171–196.

C. Audet (2004), ‘Convergence results for generalized pattern search algorithms are tight’, *Optim. Engng* **5**, 101–122.

C. Audet (2014), A survey on direct search methods for blackbox optimization and their applications. In *Mathematics Without Boundaries: Surveys in Interdisciplinary Research* (P. M. Pardalos and T. M. Rassias, eds), Springer, pp. 31–56.

C. Audet and J. E. Dennis, Jr (2000), ‘Pattern search algorithms for mixed variable programming’, *SIAM J. Optim.* **11**, 573–594.

C. Audet and J. E. Dennis, Jr (2002), ‘Analysis of generalized pattern searches’, *SIAM J. Optim.* **13**, 889–903.

C. Audet and J. E. Dennis, Jr (2004), ‘A pattern search filter method for nonlinear programming without derivatives’, *SIAM J. Optim.* **14**, 980–1010.

C. Audet and J. E. Dennis, Jr (2006), ‘Mesh adaptive direct search algorithms for constrained optimization’, *SIAM J. Optim.* **17**, 188–217.

C. Audet and J. E. Dennis, Jr (2009), ‘A progressive barrier for derivative-free nonlinear programming’, *SIAM J. Optim.* **20**, 445–472.

C. Audet and W. L. Hare (2017), *Derivative-Free and Blackbox Optimization*, Springer.

C. Audet, J. Bigeon, D. Cartier, S. Le Digabel and L. Salomon (2018*a*), Performance indicators in multiobjective optimization. Technical report, Cahier du GERAD G-2018-90, GERAD.

C. Audet, J. E. Dennis, Jr and S. Le Digabel (2008*a*), ‘Parallel space decomposition of the mesh adaptive direct search algorithm’, *SIAM J. Optim.* **19**, 1150–1170.

C. Audet, A. Ianni, S. Le Digabel and C. Tribes (2014), ‘Reducing the number of function evaluations in mesh adaptive direct search algorithms’, *SIAM J. Optim.* **24**, 621–642.

C. Audet, A. Ihaddadene, S. Le Digabel and C. Tribes (2018*b*), ‘Robust optimization of noisy blackbox problems using the mesh adaptive direct search algorithm’, *Optim. Lett.* **12**, 675–689.

C. Audet, S. Le Digabel and M. Peyrega (2015), ‘Linear equalities in blackbox optimization’, *Comput. Optim. Appl.* **61**, 1–23.

C. Audet, G. Savard and W. Zghal (2008*b*), ‘Multiobjective optimization through a series of single-objective formulations’, *SIAM J. Optim.* **19**, 188–210.

C. Audet, G. Savard and W. Zghal (2010), ‘A mesh adaptive direct search algorithm for multiobjective optimization’, *European J. Oper. Res.* **204**, 545–556.

P. Auer (2002), ‘Using confidence bounds for exploitation-exploration trade-offs’, *J. Mach. Learn. Res.* **3**, 397–422.

P. Auer, N. Cesa-Bianchi and P. Fischer (2002), ‘Finite-time analysis of the multiarmed bandit problem’, *Machine Learning* **47**, 235–256.

P. Auer, N. Cesa-Bianchi, Y. Freund and R. E. Schapire (2003), ‘The nonstochastic multiarmed bandit problem’, *SIAM J. Comput.* **32**, 48–77.

A. Auger, N. Hansen, J. Perez Zerpa, R. Ros and M. Schoenauer (2009), Experimental comparisons of derivative free optimization algorithms. In *Experimental Algorithms (SEA 2009)* (J. Vahrenhold, ed.), Vol. 5526 of Lecture Notes in Computer Science, Springer, pp. 3–15.

F. Augustin and Y. M. Marzouk (2014), NOWPAC: A provably convergent derivative-free nonlinear optimizer with path-augmented constraints. arXiv:1403.1931

F. Augustin and Y. M. Marzouk (2017), A trust-region method for derivative-free nonlinear constrained stochastic optimization. arXiv:1703.04156

M. Avriel and D. J. Wilde (1967), ‘Optimal condenser design by geometric programming’, *Ind. Engng Chem. Process Des. Dev.* **6**, 256–263.

F. Bach and V. Perchet (2016), Highly-smooth zero-th order online optimization. In *29th Annual Conference on Learning Theory (COLT 2016)* (V. Feldman *et al.*, eds). *Proc. Mach. Learn. Res.* **49**, 257–283.

A. M. Bagirov and J. Ugon (2006), ‘Piecewise partially separable functions and a derivative-free algorithm for large scale nonsmooth optimization’, *J. Global Optim.* **35**, 163–195.

A. M. Bagirov, B. Karasözen and M. Sezer (2007), ‘Discrete gradient method: Derivative-free method for nonsmooth optimization’, *J. Optim. Theory Appl.* **137**, 317–334.

I. Bajaj, S. S. Iyer and M. M. F. Hasan (2018), ‘A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point’, *Comput. Chem. Engng* **116**, 306–321.

P. Balaprakash, M. Salim, T. D. Uram, V. Vishwanath and S. M. Wild (2018), DeepHyper: Asynchronous hyperparameter search for deep neural networks. In *25th IEEE International Conference on High Performance Computing, Data, and Analytics (HiPC18)*. doi:10.1007/s10589-019-00063-3

K. Balasubramanian and S. Ghadimi (2018), Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. In *Advances in Neural Information Processing Systems 31* (S. Bengio *et al.*, eds), Curran Associates, pp. 3459–3468.

K. Balasubramanian and S. Ghadimi (2019), Zeroth-order nonconvex stochastic optimization: Handling constraints, high-dimensionality, and saddle-points. arXiv:1809.06474

A. S. Bandeira, K. Scheinberg and L. N. Vicente (2012), 'Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization', *Math. Program.* **134**, 223–257.

A. S. Bandeira, K. Scheinberg and L. N. Vicente (2014), 'Convergence of trust-region methods based on probabilistic models', *SIAM J. Optim.* **24**, 1238–1264.

N. V. Banichuk, V. M. Petrov and F. L. Chernous'ko (1966), 'The solution of variational and boundary value problems by the method of local variations', *USSR Comput. Math. Math. Phys.* **6**, 1–21.

R. R. Barton and J. S. Ivey, Jr (1996), 'Nelder–Mead simplex modifications for simulation optimization', *Manag. Sci.* **42**, 954–973.

J. Barzilai and J. M. Borwein (1988), 'Two-point step size gradient methods', *IMA J. Numer. Anal.* **8**, 141–148.

H. H. Bauschke, W. L. Hare and W. M. Moursi (2014), 'A derivative-free comirror algorithm for convex optimization', *Optim. Methods Software* **30**, 706–726.

A. G. Baydin, B. A. Pearlmutter, A. A. Radul and J. M. Siskind (2018), 'Automatic differentiation in machine learning: A survey', *J. Mach. Learn. Res.* **18** (153), 1–43.

P. Belitz and T. Bewley (2013), 'New horizons in sphere-packing theory, II: Lattice-based derivative-free optimization via global surrogates', *J. Global Optim.* **56**, 61–91.

A. Belloni, T. Liang, H. Narayanan and A. Rakhlin (2015), Escaping the local minima via simulated annealing: Optimization of approximately convex functions. In *28th Conference on Learning Theory (COLT 2015). Proc. Mach. Learn. Res.* **40**, 240–265.

P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke and A. Mahajan (2013), Mixed-integer nonlinear optimization. In *Acta Numerica*, Vol. 22, Cambridge University Press, pp. 1–131.

A. S. Berahas, R. H. Byrd and J. Nocedal (2019), 'Derivative-free optimization of noisy functions via quasi-Newton methods', *SIAM J. Optim.*, to appear. arXiv:1803.10173

D. Bertsimas and O. Nohadani (2010), 'Robust optimization with simulated annealing', *J. Global Optim.* **48**, 323–334.

D. Bertsimas, O. Nohadani and K. M. Teo (2010), 'Robust optimization for unconstrained simulation-based problems', *Oper. Res.* **58**, 161–178.

S. Bhatnagar, H. Prasad and L. A. Prashanth (2013), Kiefer–Wolfowitz algorithm. In *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods*, Vol. 434 of Lecture Notes in Control and Information Sciences, Springer, pp. 31–39.

A. Bibi, E. H. Bergou, O. Sener, B. Ghanem and P. Richtárik (2019), A stochastic derivative-free optimization method with importance sampling. arXiv:1902.01272

S. C. Billups, J. Larson and P. Graf (2013), 'Derivative-free optimization of expensive functions with computational error using weighted regression', *SIAM J. Optim.* **55**, 27–53.

M. Björkman and K. Holmström (2000), 'Global optimization of costly nonconvex functions using radial basis functions', *Optim. Engng* **1**, 373–397.

J. Blanchet, C. Cartis, M. Menickelly and K. Scheinberg (2019), 'Convergence rate analysis of a stochastic trust region method via submartingales', *INFORMS J. Optim.*, to appear. arXiv:1609.07428

J. R. Blum (1954*a*), 'Approximation methods which converge with probability one', *Ann. Math. Statist.* **25**, 382–386.

J. R. Blum (1954*b*), 'Multidimensional stochastic approximation methods', *Ann. Math. Statist.* **25**, 737–744.

C. G. E. Boender, A. H. G. Rinnooy Kan, G. T. Timmer and L. Stougie (1982), 'A stochastic method for global optimization', *Math. Program.* **22**, 125–140.

C. Bogani, M. G. Gasparo and A. Papini (2009), 'Generalized pattern search methods for a class of nonsmooth optimization problems with structure', *J. Comput. Appl. Math.* **229**, 283–293.

D. M. Bortz and C. T. Kelley (1998), The simplex gradient and noisy optimization problems. In *Computational Methods for Optimal Design and Control* (J. Borggaard *et al.*, eds), Vol. 24 of Progress in Systems and Control Theory, Birkhäuser, pp. 77–90.

L. Bottou, F. E. Curtis and J. Nocedal (2018), 'Optimization methods for large-scale machine learning', *SIAM Review* **60**, 223–311.

G. E. P. Box and N. R. Draper (1987), *Empirical Model Building and Response Surfaces*, Wiley.

M. J. Box (1965), 'A new method of constrained optimization and a comparison with other methods', *Comput. J.* **8**, 42–52.

M. J. Box (1966), 'A comparison of several current optimization methods, and the use of transformations in constrained problems', *Comput. J.* **9**, 67–77.

R. Brekelmans, L. Driessen, H. Hamers and D. den Hertog (2005), 'Constrained optimization involving expensive function evaluations: A sequential approach', *European J. Oper. Res.* **160**, 121–138.

R. P. Brent (1973), *Algorithms for Minimization Without Derivatives*, Prentice Hall.

K. M. Brown and J. E. Dennis, Jr (1971), 'Derivative free analogues of the Levenberg–Marquardt and Gauss algorithms for nonlinear least squares approximation', *Numer. Math.* **18**, 289–297.

S. Bubeck and N. Cesa-Bianchi (2012), 'Regret analysis of stochastic and nonstochastic multi-armed bandit problems', *Found. Trends Mach. Learn.* **5**, 1–122.

S. Bubeck, Y. T. Lee and R. Eldan (2017), Kernel-based methods for bandit convex optimization. In *49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017)*, ACM, pp. 72–85.

S. Bubeck, R. Munos, G. Stoltz and C. Szepesvári (2011*a*), '$\mathcal{X}$-armed bandits', *J. Mach. Learn. Res.* **12**, 1655–1695.

S. Bubeck, G. Stoltz and J. Y. Yu (2011*b*), Lipschitz bandits without the Lipschitz constant. In *Algorithmic Learning Theory (ALT 2011)* (J. Kivinen *et al.*, eds), Vol. 6925 of Lecture Notes in Computer Science, Springer, pp. 144–158.

L. F. Bueno, A. Friedlander, J. M. Martínez and F. N. C. Sobral (2013), 'Inexact restoration method for derivative-free optimization with smooth constraints', *SIAM J. Optim.* **23**, 1189–1213.

M. D. Buhmann (2000), Radial basis functions. In *Acta Numerica*, Vol. 9, Cambridge University Press, pp. 1–38.

A. D. Bull (2011), ‘Convergence rates of efficient global optimization algorithms’, *J. Mach. Learn. Res.* **12**, 2879–2904.

J. V. Burke, F. E. Curtis, A. S. Lewis, M. L. Overton and L. E. A. Simões (2018), Gradient sampling methods for nonsmooth optimization. arXiv:1804.11003

Á. Bűrmen, J. Olenšek and T. Tuma (2015), ‘Mesh adaptive direct search with second directional derivative-based Hessian update’, *Comput. Optim. Appl.* **62**, 693–715.

Á. Bűrmen, J. Puhan and T. Tuma (2006), ‘Grid restrained Nelder–Mead algorithm’, *Comput. Optim. Appl.* **34**, 359–375.

R. G. Carter, J. M. Gablonsky, A. Patrick, C. T. Kelley and O. J. Eslinger (2001), ‘Algorithms for noisy problems in gas transmission pipeline optimization’, *Optim. Engng* **2**, 139–157.

C. Cartis and L. Roberts (2017), A derivative-free Gauss–Newton method. arXiv:1710.11005

C. Cartis and K. Scheinberg (2018), ‘Global convergence rate analysis of unconstrained optimization methods based on probabilistic models’, *Math. Program.* **169**, 337–375.

C. Cartis and J. Fiala and B. Marteau and L. Roberts (2018), Improving the flexibility and robustness of model-based derivative-free optimization solvers. arXiv:1804.00154

C. Cartis, N. I. M. Gould and P. L. Toint (2012), ‘On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization’, *SIAM J. Optim.* **22**, 66–86.

C. Cartis, N. I. M. Gould and P. L. Toint (2018), Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints. arXiv:1811.01220

B. Castle (2012), Quasi-Newton methods for stochastic optimization and proximity-based methods for disparate information fusion. PhD thesis, Indiana University.

J. Céa (1971), *Optimisation: Théorie et Algorithmes*, Méthodes Mathématiques de l’Informatique, Dunod.

G. Chandramouli and V. Narayanan (2019), A scaled conjugate gradient based direct search algorithm for high dimensional box constrained derivative free optimization. arXiv:1901.05215

K.-H. Chang (2012), ‘Stochastic Nelder–Mead simplex method: A new globally convergent direct search method for simulation optimization’, *European J. Oper. Res.* **220**, 684–694.

K.-H. Chang, L. J. Hong and H. Wan (2013), ‘Stochastic trust-region response-surface method (STRONG): A new response-surface framework for simulation optimization’, *INFORMS J. Comput.* **25**, 230–243.

H. Chen and B. W. Schmeiser (2001), ‘Stochastic root finding via retrospective approximation’, *IIE Trans.* **33**, 259–275.

R. Chen (2015), Stochastic derivative-free optimization of noisy functions. PhD thesis, Lehigh University.

R. Chen, M. Menickelly and K. Scheinberg (2018*a*), 'Stochastic optimization using a trust-region method and random models', *Math. Program.* **169**, 447–487.

X. Chen and C. T. Kelley (2016), 'Optimization with hidden constraints and embedded Monte Carlo computations', *Optim. Engng* **17**, 157–175.

X. Chen, C. T. Kelley, F. Xu and Z. Zhang (2018*b*), 'A smoothing direct search method for Monte Carlo-based bound constrained composite nonsmooth optimization', *SIAM J. Sci. Comput.* **40**, A2174–A2199.

T. D. Choi and C. T. Kelley (2000), 'Superlinear convergence and implicit filtering', *SIAM J. Optim.* **10**, 1149–1162.

T. D. Choi, O. J. Eslinger, C. T. Kelley, J. W. David and M. Etheridge (2000), 'Optimization of automotive valve train components with implicit filtering', *Optim. Engng* **1**, 9–27.

A. Ciccazzo, V. Latorre, G. Liuzzi, S. Lucidi and F. Rinaldi (2015), 'Derivative-free robust optimization for circuit design', *J. Optim. Theory Appl.* **164**, 842–861.

G. Cocchi, G. Liuzzi, A. Papini and M. Sciandrone (2018), 'An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints', *Comput. Optim. Appl.* **69**, 267–296.

B. Colson and P. L. Toint (2001), 'Exploiting band structure in unconstrained optimization without derivatives', *Optim. Engng* **2**, 399–412.

B. Colson and P. L. Toint (2002), A derivative-free algorithm for sparse unconstrained optimization problems. In *Trends in Industrial and Applied Mathematics* (A. H. Siddiqi and M. Kočvara, eds), Vol. 72 of Applied Optimization, Springer, pp. 131–147.

B. Colson and P. L. Toint (2005), 'Optimizing partially separable functions without derivatives', *Optim. Methods Software* **20**, 493–508.

P. D. Conejo, E. W. Karas and L. G. Pedroso (2015), 'A trust-region derivative-free algorithm for constrained optimization', *Optim. Methods Software* **30**, 1126–1145.

P. D. Conejo, E. W. Karas, L. G. Pedroso, A. A. Ribeiro and M. Sachine (2013), 'Global convergence of trust-region algorithms for convex constrained minimization without derivatives', *Appl. Math. Comput.* **220**, 324–330.

A. R. Conn and S. Le Digabel (2013), 'Use of quadratic models with mesh-adaptive direct search for constrained black box optimization', *Optim. Methods Software* **28**, 139–158.

A. R. Conn and P. L. Toint (1996), An algorithm using quadratic interpolation for unconstrained derivative free optimization. In *Nonlinear Optimization and Applications* (G. Di Pillo and F. Giannessi, eds), Springer, pp. 27–47.

A. R. Conn and L. N. Vicente (2012), 'Bilevel derivative-free optimization and its application to robust optimization', *Optim. Methods Software* **27**, 561–577.

A. R. Conn, N. I. M. Gould and P. L. Toint (1991), 'A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds', *SIAM J. Numer. Anal.* **28**, 545–572.

A. R. Conn, N. I. M. Gould and P. L. Toint (2000), *Trust-Region Methods*, SIAM.

A. R. Conn, K. Scheinberg and P. L. Toint (1997*a*), On the convergence of derivative-free methods for unconstrained optimization. In *Approximation Theory and Optimization: Tributes to M. J. D. Powell* (A. Iserles and M. Buhmann, eds), Cambridge University Press, pp. 83–108.

A. R. Conn, K. Scheinberg and P. L. Toint (1997*b*), 'Recent progress in unconstrained nonlinear optimization without derivatives', *Math. Program.* **79**, 397–414.

A. R. Conn, K. Scheinberg and P. L. Toint (1998), A derivative free optimization algorithm in practice. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, American Institute of Aeronautics and Astronautics.

A. R. Conn, K. Scheinberg and P. L. Toint (2001), DFO (derivative free optimization software). https://projects.coin-or.org/Dfo

A. R. Conn, K. Scheinberg and L. N. Vicente (2008*a*), 'Geometry of interpolation sets in derivative free optimization', *Math. Program.* **111**, 141–172.

A. R. Conn, K. Scheinberg and L. N. Vicente (2008*b*), 'Geometry of sample sets in derivative free optimization: Polynomial regression and underdetermined interpolation', *IMA J. Numer. Anal.* **28**, 721–748.

A. R. Conn, K. Scheinberg and L. N. Vicente (2009*a*), 'Global convergence of general derivative-free trust-region algorithms to first and second order critical points', *SIAM J. Optim.* **20**, 387–415.

A. R. Conn, K. Scheinberg and L. N. Vicente (2009*b*), *Introduction to Derivative-Free Optimization*, SIAM.

I. D. Coope and C. J. Price (2000), 'Frame based methods for unconstrained optimization', *J. Optim. Theory Appl.* **107**, 261–274.

I. D. Coope and R. Tappenden (2019), 'Efficient calculation of regular simplex gradients', *Comput. Optim. Appl.* **72**, 561–588.

A. L. Custódio and J. F. A. Madeira (2015), 'GLODS: Global and local optimization using direct search', *J. Global Optim.* **62**, 1–28.

A. L. Custódio and J. F. A. Madeira (2016), MultiGLODS: Global and local multiobjective optimization using direct search. Technical report, Universidade Nova de Lisboa.

A. L. Custódio and L. N. Vicente (2005), SID-PSM: A pattern search method guided by simplex derivatives for use in derivative-free optimization. http://www.mat.uc.pt/sid-psm

A. L. Custódio and L. N. Vicente (2007), 'Using sampling and simplex derivatives in pattern search methods', *SIAM J. Optim.* **18**, 537–555.

A. L. Custódio, J. E. Dennis, Jr and L. N. Vicente (2008), 'Using simplex gradients of nonsmooth functions in direct search methods', *IMA J. Numer. Anal.* **28**, 770–784.

A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz and L. N. Vicente (2011), 'Direct multisearch for multiobjective optimization', *SIAM J. Optim.* **21**, 1109–1140.

A. L. Custódio, H. Rocha and L. N. Vicente (2009), 'Incorporating minimum Frobenius norm models in direct search', *Comput. Optim. Appl.* **46**, 265–278.

L. Dai (2016*a*), Convergence rates of finite difference stochastic approximation algorithms, I: General sampling. In *Sensing and Analysis Technologies for Biomedical and Cognitive Applications* (L. Dai *et al.*, eds), SPIE.

L. Dai (2016*b*), Convergence rates of finite difference stochastic approximation algorithms, II: Implementation via common random numbers. In *Sensing and Analysis Technologies for Biomedical and Cognitive Applications* (L. Dai *et al.*, eds), SPIE.

C. D'Ambrosio, G. Nannicini and G. Sartor (2017), 'MILP models for the selection of a small set of well-distributed points', *Oper. Res. Lett.* **45**, 46–52.

C. Davis (1954), 'Theory of positive linear dependence', *Amer. J. Math.* **76**, 733–746.

P. Davis (2005), Michael J. D. Powell: An oral history. In *History of Numerical Analysis and Scientific Computing Project*, SIAM.

C. Davis and W. L. Hare (2013), 'Exploiting known structures to approximate normal cones', *Math. Oper. Res.* **38**, 665–681.

R. De Leone, M. Gaudioso and L. Grippo (1984), 'Stopping criteria for linesearch methods without derivatives', *Math. Program.* **30**, 285–300.

O. Dekel, R. Eldan and T. Koren (2015), Bandit smooth convex optimization: Improving the bias-variance tradeoff. In *Advances in Neural Information Processing Systems 28* (C. Cortes *et al.*, eds), Curran Associates, pp. 2926–2934.

A. Dener, A. Denchfield, T. Munson, J. Sarich, S. M. Wild, S. Benson and L. Curfman McInnes (2018), TAO 3.10 users manual. Technical Memorandum ANL/MCS-TM-322, Argonne National Laboratory.

G. Deng and M. C. Ferris (2009), 'Variable-number sample-path optimization', *Math. Program.* **117**, 81–109.

G. Deng and M. C. Ferris (2006), Adaptation of the UOBYQA algorithm for noisy functions. In *Proceedings of the 2006 Winter Simulation Conference*, IEEE, pp. 312–319.

J. E. Dennis, Jr and V. Torczon (1991), 'Direct search methods on parallel machines', *SIAM J. Optim.* **1**, 448–474.

J. E. Dennis, Jr and V. Torczon (1997), Managing approximation models in optimization. In *Multidisciplinary Design Optimization: State-of-the-Art* (N. M. Alexandrov and M. Y. Hussaini, eds), SIAM, pp. 330–347.

J. E. Dennis, Jr and D. J. Woods (1987), Optimization on microcomputers: The Nelder–Mead simplex algorithm. In *New Computing Environments: Microcomputers in Large-Scale Computing* (A. Wouk, ed.), SIAM, pp. 116–122.

C. Derman (1956), 'An application of Chung's lemma to the Kiefer–Wolfowitz stochastic approximation procedure', *Ann. Math. Statist.* **27**, 532–536.

G. I. Diaz, A. Fokoue-Nkoutche, G. Nannicini and H. Samulowitz (2017), 'An effective algorithm for hyperparameter optimization of neural networks', *IBM J. Res. Develop.* **61**, 9:1–9:11.

M. A. Diniz-Ehrhardt, J. M. Martínez and L. G. Pedroso (2011), 'Derivative-free methods for nonlinear programming with general lower-level constraints', *Comput. Appl. Math.* **30**, 19–52.

M. A. Diniz-Ehrhardt, J. M. Martínez and M. Raydan (2008), 'A derivative-free nonmonotone line-search technique for unconstrained optimization', *J. Comput. Appl. Math.* **219**, 383–397.

M. Dodangeh and L. N. Vicente (2016), 'Worst case complexity of direct search under convexity', *Math. Program.* **155**, 307–332.

M. Dodangeh, L. N. Vicente and Z. Zhang (2016), 'On the optimal order of worst case complexity of direct search', *Optim. Lett.* **10**, 699–708.

E. D. Dolan, R. M. Lewis and V. Torczon (2003), 'On the local convergence of pattern search', *SIAM J. Optim.* **14**, 567–583.

J. C. Duchi, M. I. Jordan, M. J. Wainwright and A. Wibisono (2015), 'Optimal rates for zero-order convex optimization: The power of two function evaluations', *IEEE Trans. Inform. Theory* **61**, 2788–2806.

R. Durrett (2010), *Probability: Theory and Examples*, Cambridge University Press.

P. Dvurechensky, A. Gasnikov and E. Gorbunov (2018), An accelerated method for derivative-free smooth stochastic convex optimization. arXiv:1802.09022

N. Echebest, M. L. Schuverdt and R. P. Vignau (2015), 'An inexact restoration derivative-free filter method for nonlinear programming', *Comput. Appl. Math.* **36**, 693–718.

M. Ehrgott (2005), *Multicriteria Optimization*, second edition, Springer.

C. Elster and A. Neumaier (1995), 'A grid algorithm for bound constrained optimization of noisy functions', *IMA J. Numer. Anal.* **15**, 585–608.

V. Fabian (1971), Stochastic approximation. In *Optimizing Methods in Statistics*, Elsevier, pp. 439–470.

G. Fasano, G. Liuzzi, S. Lucidi and F. Rinaldi (2014), 'A linesearch-based derivative-free approach for nonsmooth constrained optimization', *SIAM J. Optim.* **24**, 959–992.

G. Fasano, J. L. Morales and J. Nocedal (2009), 'On the geometry phase in model-based algorithms for derivative-free optimization', *Optim. Methods Software* **24**, 145–154.

E. Fermi and N. Metropolis (1952), Numerical solution of a minimum problem. Technical report LA-1492, Los Alamos Scientific Laboratory of the University of California.

P. S. Ferreira, E. W. Karas, M. Sachine and F. N. C. Sobral (2017), 'Global convergence of a derivative-free inexact restoration filter algorithm for nonlinear programming', *Optimization* **66**, 271–292.

D. E. Finkel and C. T. Kelley (2004), Convergence analysis of the DIRECT algorithm. Technical report 04-28, Center for Research in Scientific Computation, North Carolina State University.
https://projects.ncsu.edu/crsc/reports/ftp/pdf/crsc-tr04-28.pdf

D. E. Finkel and C. T. Kelley (2006), 'Additive scaling and the DIRECT algorithm', *J. Global Optim.* **36**, 597–608.

D. E. Finkel and C. T. Kelley (2009), 'Convergence analysis of sampling methods for perturbed Lipschitz functions', *Pacific J. Optim.* **5**, 339–350.

A. Flaxman, A. Kalai and B. McMahan (2005), Online convex optimization in the bandit setting: Gradient descent without a gradient. In *16th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '05)*, ACM, pp. 385–394.

R. Fletcher (1965), 'Function minimization without evaluating derivatives: A review', *Comput. J.* **8**, 33–41.

R. Fletcher (1987), *Practical Methods of Optimization*, second edition, Wiley.

A. Forrester, A. Sobester and A. Keane (2008), *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley.

S. Forth, P. Hovland, E. Phipps, J. Utke and A. Walther, eds (2012), *Recent Advances in Algorithmic Differentiation*, Springer.

E. Frandi and A. Papini (2013), 'Coordinate search algorithms in multilevel optimization', *Optim. Methods Software* **29**, 1020–1041.

P. I. Frazier (2018), A tutorial on Bayesian optimization. arXiv:1807.02811

L. Frimannslund and T. Steihaug (2007), ‘A generating set search method using curvature information’, *Comput. Optim. Appl.* **38**, 105–121.

L. Frimannslund and T. Steihaug (2010), A new generating set search algorithm for partially separable functions. In *4th International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2010)*, IARIA, pp. 65–70.

L. Frimannslund and T. Steihaug (2011), ‘On a new method for derivative free optimization’, *Internat. J. Adv. Software* **4**, 244–255.

M. C. Fu, F. W. Glover and J. April (2005), Simulation optimization: A review, new developments, and applications. In *Proceedings of the 2005 Winter Simulation Conference*, IEEE.

F. Gao and L. Han (2012), ‘Implementing the Nelder–Mead simplex algorithm with adaptive parameters’, *Comput. Optim. Appl.* **51**, 259–277.

U. M. García-Palomares and J. F. Rodríguez (2002), ‘New sequential and parallel derivative-free algorithms for unconstrained minimization’, *SIAM J. Optim.* **13**, 79–96.

U. M. García-Palomares, I. J. García-Urrea and P. S. Rodríguez-Hernández (2013), ‘On sequential and parallel non-monotone derivative-free algorithms for box constrained optimization’, *Optim. Methods Software* **28**, 1233–1261.

R. Garmanjani and L. N. Vicente (2012), ‘Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization’, *IMA J. Numer. Anal.* **33**, 1008–1028.

R. Garmanjani, D. Jùdice and L. N. Vicente (2016), ‘Trust-region methods without using derivatives: Worst case complexity and the nonsmooth case’, *SIAM J. Optim.* **26**, 1987–2011.

A. V. Gasnikov, E. A. Krymova, A. A. Lagunovskaya, I. N. Usmanova and F. A. Fedorenko (2017), ‘Stochastic online optimization: Single-point and multi-point non-linear multi-armed bandits; Convex and strongly-convex case’, *Automat. Remote Control* **78**, 224–234.

L. Gerencsér (1997), Rate of convergence of moments of Spall’s SPSA method. In *Stochastic Differential and Difference Equations*, Vol. 23 of Progress in Systems and Control Theory, Birkhäuser, pp. 67–75.

S. Ghadimi (2019), ‘Conditional gradient type methods for composite nonlinear and stochastic optimization’, *Math. Program.* **173**, 431–464.

S. Ghadimi and G. Lan (2013), ‘Stochastic first- and zeroth-order methods for nonconvex stochastic programming’, *SIAM J. Optim.* **23**, 2341–2368.

P. E. Gill, W. Murray and M. H. Wright (1981), *Practical Optimization*, Academic Press.

P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright (1983), ‘Computing forward-difference intervals for numerical optimization’, *SIAM J. Sci. Statist. Comput.* **4**, 310–321.

P. Gilmore and C. T. Kelley (1995), ‘An implicit filtering algorithm for optimization of functions with many local minima’, *SIAM J. Optim.* **5**, 269–285.

H. Glass and L. Cooper (1965), ‘Sequential search: A method for solving constrained optimization problems’, *J. Assoc. Comput. Mach.* **12**, 71–82.

D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro and D. Sculley (2017), Google Vizier: A service for black-box optimization. In *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*, ACM, pp. 1487–1495.

C. C. Gonzaga, E. Karas and M. Vanti (2004), 'A globally convergent filter method for nonlinear programming', *SIAM J. Optim.* **14**, 646–669.

N. I. M. Gould and P. L. Toint (2010), 'Nonlinear programming without a penalty function or a filter', *Math. Program.* **122**, 155–196.

R. B. Gramacy and S. Le Digabel (2015), 'The mesh adaptive direct search algorithm with treed Gaussian process surrogates', *Pacific J. Optim.* **11**, 419–447.

G. N. Grapiglia, J. Yuan and Y.-x. Yuan (2016), 'A derivative-free trust-region algorithm for composite nonsmooth optimization', *Comput. Appl. Math.* **35**, 475–499.

S. Gratton and L. N. Vicente (2014), 'A merit function approach for direct search', *SIAM J. Optim.* **24**, 1980–1998.

S. Gratton, C. W. Royer and L. N. Vicente (2016), 'A second-order globally convergent direct-search method and its worst-case complexity', *Optimization* **65**, 1105–1128.

S. Gratton, C. W. Royer and L. N. Vicente (2019*a*), 'A decoupled first/second-order steps technique for nonconvex nonlinear unconstrained optimization with improved complexity bounds', *Math. Program.* doi:10.1007/s10107-018-1328-7

S. Gratton, C. W. Royer, L. N. Vicente and Z. Zhang (2015), 'Direct search based on probabilistic descent', *SIAM J. Optim.* **25**, 1515–1541.

S. Gratton, C. W. Royer, L. N. Vicente and Z. Zhang (2018), 'Complexity and global rates of trust-region methods based on probabilistic models', *IMA J. Numer. Anal.* **38**, 1579–1597.

S. Gratton, C. W. Royer, L. N. Vicente and Z. Zhang (2019*b*), 'Direct search based on probabilistic feasible descent for bound and linearly constrained problems', *Comput. Optim. Appl.* **72**, 525–559.

S. Gratton, P. L. Toint and A. Tröltzsch (2011), 'An active-set trust-region method for derivative-free nonlinear bound-constrained optimization', *Optim. Methods Software* **26**, 873–894.

G. A. Gray and T. G. Kolda (2006), 'Algorithm 856: APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization', *ACM Trans. Math. Software* **32**, 485–507.

A. Griewank (2003), A mathematical view of automatic differentiation. In *Acta Numerica*, Vol. 12, Cambridge University Press, pp. 321–398.

A. Griewank and A. Walther (2008), *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, SIAM.

A. Griewank, A. Walther, S. Fiege and T. Bosse (2016), 'On Lipschitz optimization based on gray-box piecewise linearization', *Math. Program.* **158**, 383–415.

L. Grippo and F. Rinaldi (2014), 'A class of derivative-free nonmonotone optimization algorithms employing coordinate rotations and gradient approximations', *Comput. Optim. Appl.* **60**, 1–33.

L. Grippo and M. Sciandrone (2007), 'Nonmonotone derivative-free methods for nonlinear equations', *Comput. Optim. Appl.* **37**, 297–328.

L. Grippo, F. Lampariello and S. Lucidi (1988), 'Global convergence and stabilization of unconstrained minimization methods without derivatives', *J. Optim. Theory Appl.* **56**, 385–406.

B. Gu, Z. Huo and H. Huang (2016), Zeroth-order asynchronous doubly stochastic algorithm with variance reduction. arXiv:1612.01425

E. A. E. Gumma, M. H. A. Hashim and M. M. Ali (2014), 'A derivative-free algorithm for linearly constrained optimization problems', *Comput. Optim. Appl.* **57**, 599–621.

M. D. Gunzburger, C. G. Webster and G. Zhang (2014), Stochastic finite element methods for partial differential equations with random input data. In *Acta Numerica*, Vol. 23, Cambridge University Press, pp. 521–650.

H.-M. Gutmann (2001), 'A radial basis function method for global optimization', *J. Global Optim.* **19**, 201–227.

L. Han and G. Liu (2004), 'On the convergence of the UOBYQA method', *J. Appl. Math. Comput.* **16**, 125–142.

P. Hansen, B. Jaumard and S.-H. Lu (1991), 'On the number of iterations of Piyavskii's global optimization algorithm', *Math. Oper. Res.* **16**, 334–350.

W. L. Hare (2014), 'Numerical analysis of $\mathcal{VU}$-decomposition, $\mathcal{U}$-gradient, and $\mathcal{U}$-Hessian approximations', *SIAM J. Optim.* **24**, 1890–1913.

W. L. Hare (2017), 'Compositions of convex functions and fully linear models', *Optim. Lett.* **11**, 1217–1227.

W. L. Hare and A. S. Lewis (2005), 'Estimating tangent and normal cones without calculus', *Math. Oper. Res.* **30**, 785–799.

W. L. Hare and Y. Lucet (2013), 'Derivative-free optimization via proximal point methods', *J. Optim. Theory Appl.* **160**, 204–220.

W. L. Hare and M. Macklem (2013), 'Derivative-free optimization methods for finite minimax problems', *Optim. Methods Software* **28**, 300–312.

W. L. Hare and J. Nutini (2013), 'A derivative-free approximate gradient sampling algorithm for finite minimax problems', *Comput. Optim. Appl.* **56**, 1–38.

W. L. Hare, C. Planiden and C. Sagastizábal (2019), A derivative-free $\mathcal{VU}$-algorithm for convex finite-max problems. arXiv:1903.11184

E. Hazan, T. Koren and K. Y. Levy (2014), Logistic regression: Tight bounds for stochastic and online optimization. In *27th Conference on Learning Theory (COLT 2014)* (M. F. Balcan *et al.*, eds). *Proc. Mach. Learn. Res.* **35**, 197–209.

J. He, A. Verstak, M. Sosonkina and L. T. Watson (2009*a*), 'Performance modeling and analysis of a massively parallel DIRECT, part 2', *Internat. J. High Performance Comput. Appl.* **23**, 29–41.

J. He, A. Verstak, L. T. Watson and M. Sosonkina (2007), 'Design and implementation of a massively parallel version of DIRECT', *Comput. Optim. Appl.* **40**, 217–245.

J. He, A. Verstak, L. T. Watson and M. Sosonkina (2009*b*), 'Performance modeling and analysis of a massively parallel DIRECT, part 1', *Internat. J. High Performance Comput. Appl.* **23**, 14–28.

J. He, L. T. Watson and M. Sosonkina (2009*c*), 'Algorithm 897: VTDIRECT95: Serial and parallel codes for the global optimization algorithm DIRECT', *ACM Trans. Math. Software* **36**, 1–24.

T. Homem-de-Mello and G. Bayraksan (2014), 'Monte Carlo sampling-based methods for stochastic optimization', *Surv. Oper. Res. Manag. Sci.* **19**, 56–85.

R. Hooke and T. A. Jeeves (1961), '"Direct search" solution of numerical and statistical problems', *J. Assoc. Comput. Mach.* **8**, 212–229.

P. D. Hough and J. C. Meza (2002), 'A class of trust-region methods for parallel optimization', *SIAM J. Optim.* **13**, 264–282.

P. D. Hough, T. G. Kolda and V. J. Torczon (2001), 'Asynchronous parallel pattern search for nonlinear optimization', *SIAM J. Sci. Comput.* **23**, 134–156.

X. Hu, L. A. Prashanth, A. György and C. Szepesvári (2016), (Bandit) convex optimization with biased noisy gradient oracles. In *19th International Conference on Artificial Intelligence and Statistics. Proc. Mach. Learn. Res.* **51**, 819–828.

D. W. Hutchison and J. C. Spall (2013), Stochastic approximation. In *Encyclopedia of Operations Research and Management Science*, Springer, pp. 1470–1476.

W. Huyer and A. Neumaier (1999), 'Global optimization by multilevel coordinate search', *J. Global Optim.* **14**, 331–355.

W. Huyer and A. Neumaier (2008), 'SNOBFIT: Stable noisy optimization by branch and fit', *ACM Trans. Math. Software* **35**, 9:1–9:25.

I. Ilievski, T. Akhtar, J. Feng and C. Shoemaker (2017), Efficient hyperparameter optimization for deep learning algorithms using deterministic RBF surrogates. In *31st AAAI Conference on Artificial Intelligence (AAAI '17)*, pp. 822–829.

K. G. Jamieson, R. D. Nowak and B. Recht (2012), Query complexity of derivative-free optimization. In *Advances in Neural Information Processing Systems 25* (F. Pereira *et al.*, eds), Curran Associates, pp. 2672–2680.

D. R. Jones, C. D. Perttunen and B. E. Stuckman (1993), 'Lipschitzian optimization without the Lipschitz constant', *J. Optim. Theory Appl.* **79**, 157–181.

D. R. Jones, M. Schonlau and W. J. Welch (1998), 'Efficient global optimization of expensive black-box functions', *J. Global Optim.* **13**, 455–492.

V. G. Karmanov (1974), 'Convergence estimates for iterative minimization methods', *USSR Comput. Math. Math. Phys.* **14**, 1–13.

C. T. Kelley (1999*a*), 'Detection and remediation of stagnation in the Nelder–Mead algorithm using a sufficient decrease condition', *SIAM J. Optim.* **10**, 43–55.

C. T. Kelley (1999*b*), *Iterative Methods for Optimization*, SIAM.

C. T. Kelley (2003), Implicit filtering and nonlinear least squares problems. In *System Modeling and Optimization XX* (E. W. Sachs and R. Tichatschke, eds), Vol. 130 of IFIP: The International Federation for Information Processing, Springer, pp. 71–90.

C. T. Kelley (2011), *Implicit Filtering*, SIAM.

K. A. Khan, J. Larson and S. M. Wild (2018), 'Manifold sampling for optimization of nonconvex functions that are piecewise linear compositions of smooth components', *SIAM J. Optim.* **28**, 3001–3024.

J. Kiefer and J. Wolfowitz (1952), 'Stochastic estimation of the maximum of a regression function', *Ann. Math. Statist.* **22**, 462–466.

S. Kim and J.-h. Ryu (2011), 'A trust-region algorithm for bi-objective stochastic optimization', *Procedia Comput. Sci.* **4**, 1422–1430.

S. Kim and D. Zhang (2010), Convergence properties of direct search methods for stochastic optimization. In *Proceedings of the 2010 Winter Simulation Conference*, IEEE.

S. Kim, R. Pasupathy and S. G. Henderson (2015), A guide to sample average approximation. In *Handbook of Simulation Optimization* (M. Fu, ed.), Vol. 216 of International Series in Operations Research & Management Science, Springer, pp. 207–243.

K. C. Kiwiel (2010), 'A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization', *SIAM J. Optim.* **20**, 1983–1994.

R. Kleinberg, A. Slivkins and E. Upfal (2008), Multi-armed bandits in metric spaces. In *40th Annual ACM Symposium on Theory of Computing (STOC 2008)*, ACM, pp. 681–690.

N. L. Kleinman, J. C. Spall and D. Q. Naiman (1999), 'Simulation-based optimization with stochastic approximation using common random numbers', *Manag. Sci.* **45**, 1570–1578.

J. Knowles and D. Corne (2002), On metrics for comparing nondominated sets. In *Proceedings of the 2002 Congress on Evolutionary Computation*, Vol. 1, IEEE, pp. 711–716.

T. G. Kolda, R. M. Lewis and V. Torczon (2006), 'Stationarity results for generating set search for linearly constrained optimization', *SIAM J. Optim.* **17**, 943–968.

T. G. Kolda, R. M. Lewis and V. J. Torczon (2003), 'Optimization by direct search: New perspectives on some classical and modern methods', *SIAM Review* **45**, 385–482.

J. Konečný and P. Richtárik (2014), Simple complexity analysis of simplified direct search. arXiv:1410.0390

H. Kushner and G. Yin (2003), *Stochastic Approximation and Recursive Algorithms and Applications*, Springer.

H. J. Kushner and H. Huang (1979), 'Rates of convergence for stochastic approximation type algorithms', *SIAM J. Control Optim.* **17**, 607–617.

W. La Cruz (2014), 'A projected derivative-free algorithm for nonlinear equations with convex constraints', *Optim. Methods Software* **29**, 24–41.

W. La Cruz, J. M. Martínez and M. Raydan (2006), 'Spectral residual method without gradient information for solving large-scale nonlinear systems of equations', *Math. Comput.* **75** (255), 1429–1449.

J. C. Lagarias, B. Poonen and M. H. Wright (2012), 'Convergence of the restricted Nelder–Mead algorithm in two dimensions', *SIAM J. Optim.* **22**, 501–532.

J. C. Lagarias, J. A. Reeds, M. H. Wright and P. E. Wright (1998), 'Convergence properties of the Nelder–Mead simplex algorithm in low dimensions', *SIAM J. Optim.* **9**, 112–147.

T. L. Lai and H. Robbins (1985), 'Asymptotically efficient adaptive allocation rules', *Adv. Appl. Math.* **6**, 4–22.

J. Larson and S. C. Billups (2016), 'Stochastic derivative-free optimization using a trust region framework', *Comput. Optim. Appl.* **64**, 619–645.

J. Larson and S. M. Wild (2016), ‘A batch, derivative-free algorithm for finding multiple local minima’, *Optim. Engng* **17**, 205–228.

J. Larson and S. M. Wild (2018), ‘Asynchronously parallel optimization solver for finding multiple minima’, *Math. Program. Comput.* **10**, 303–332.

J. Larson, S. Leyffer, P. Palkar and S. M. Wild (2019), A method for convex black-box integer global optimization. arXiv:1903.11366

J. Larson, M. Menickelly and S. M. Wild (2016), ‘Manifold sampling for ℓ_1 nonconvex optimization’, *SIAM J. Optim.* **26**, 2540–2563.

V. Latorre, H. Habal, H. Graeb and S. Lucidi (2019), ‘Derivative free methodologies for circuit worst case analysis’, *Optim. Lett.* doi:10.1007/s11590-018-1364-5

M. Lazar and F. Jarre (2016), ‘Calibration by optimization without using derivatives’, *Optim. Engng* **17**, 833–860.

S. Le Digabel (2011), ‘Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm’, *ACM Trans. Math. Software* **37**, 44:1–44:15.

S. Le Digabel and S. M. Wild (2015), A taxonomy of constraints in black-box simulation-based optimization. arXiv:1505.07881

L. Le Gratiet and C. Cannamela (2015), ‘Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes’, *Technometrics* **57**, 418–427.

P. L’Ecuyer and G. Yin (1998), ‘Budget-dependent convergence rate of stochastic approximation’, *SIAM J. Optim.* **8**, 217–247.

H. Lee, R. B. Gramacy, C. Linkletter and G. A. Gray (2011), ‘Optimization subject to hidden constraints via statistical emulation’, *Pacific J. Optim.* **7**, 467–478.

C. Lemarechal and R. Mifflin, eds (1978), *Nonsmooth Optimization: Proceedings of an IIASA Workshop*, Pergamon Press.

K. Levenberg (1944), ‘A method for the solution of certain non-linear problems in least squares’, *Quart. Appl. Math.* **2**, 164–168.

R. M. Lewis and V. Torczon (1999), ‘Pattern search algorithms for bound constrained minimization’, *SIAM J. Optim.* **9**, 1082–1099.

R. M. Lewis and V. Torczon (2000), ‘Pattern search methods for linearly constrained minimization’, *SIAM J. Optim.* **10**, 917–941.

R. M. Lewis and V. Torczon (2002), ‘A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds’, *SIAM J. Optim.* **12**, 1075–1089.

R. M. Lewis and V. Torczon (2010), A direct search approach to nonlinear programming problems using an augmented Lagrangian method with explicit treatment of the linear constraints. Technical report WM-CS-2010-01, Department of Computer Science, College of William and Mary.

R. M. Lewis, V. Torczon and M. W. Trosset (2000), ‘Direct search methods: Then and now’, *J. Comput. Appl. Math.* **124**, 191–207.

S. Leyffer (2015), ‘It’s to solve problems: An interview with Roger Fletcher’, *Optima* **99**, 1–6.

D.-H. Li and M. Fukushima (2000), ‘A derivative-free line search and global convergence of Broyden-like method for nonlinear equations’, *Optim. Methods Software* **13**, 181–201.

Q. Li and D.-H. Li (2011), ‘A class of derivative-free methods for large-scale nonlinear monotone equations’, *IMA J. Numer. Anal.* **31**, 1625–1635.

Q. Liu, J. Zeng and G. Yang (2015), ‘MrDIRECT: A multilevel robust DIRECT algorithm for global optimization problems’, *J. Global Optim.* **62**, 205–227.

S. Liu, B. Kailkhura, P.-Y. Chen, P. Ting, S. Chang and L. Amini (2018), Zeroth-order stochastic variance reduction for nonconvex optimization. In *Advances in Neural Information Processing Systems 31* (S. Bengio *et al.*, eds), Curran Associates, pp. 3731–3741.

G. Liuzzi and S. Lucidi (2009), ‘A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ℓ_∞ penalty function’, *SIAM J. Optim.* **20**, 1–29.

G. Liuzzi, S. Lucidi and F. Rinaldi (2011), ‘Derivative-free methods for bound constrained mixed-integer optimization’, *Comput. Optim. Appl.* **53**, 505–526.

G. Liuzzi, S. Lucidi and F. Rinaldi (2015), ‘Derivative-free methods for mixed-integer constrained optimization problems’, *J. Optim. Theory Appl.* **164**, 933–965.

G. Liuzzi, S. Lucidi and F. Rinaldi (2016), ‘A derivative-free approach to constrained multiobjective nonsmooth optimization’, *SIAM J. Optim.* **26**, 2744–2774.

G. Liuzzi, S. Lucidi and F. Rinaldi (2018), An algorithmic framework based on primitive directions and nonmonotone line searches for black box problems with integer variables. Report 6471, Optimization Online. http://www.optimization-online.org/DB_HTML/2018/02/6471.html

G. Liuzzi, S. Lucidi and M. Sciandrone (2006), ‘A derivative-free algorithm for linearly constrained finite minimax problems’, *SIAM J. Optim.* **16**, 1054–1075.

G. Liuzzi, S. Lucidi and M. Sciandrone (2010), ‘Sequential penalty derivative-free methods for nonlinear constrained optimization’, *SIAM J. Optim.* **20**, 2614–2635.

S. Lucidi and M. Sciandrone (2002*a*), ‘A derivative-free algorithm for bound constrained optimization’, *Comput. Optim. Appl.* **21**, 119–142.

S. Lucidi and M. Sciandrone (2002*b*), ‘On the global convergence of derivative-free methods for unconstrained optimization’, *SIAM J. Optim.* **13**, 97–116.

S. Lucidi, M. Sciandrone and P. Tseng (2002), ‘Objective-derivative-free methods for constrained optimization’, *Math. Program.* **92**, 37–59.

J. Ma and X. Zhang (2009), ‘Pattern search methods for finite minimax problems’, *J. Appl. Math. Comput.* **32**, 491–506.

K. Madsen (1975), Minimax solution of non-linear equations without calculating derivatives. In *Nondifferentiable Optimization* (M. L. Balinski and P. Wolfe), Vol. 3 of Mathematical Programming Studies, Springer, pp. 110–126.

A. Maggiar, A. Wächter, I. S. Dolinskaya and J. Staum (2018), ‘A derivative-free trust-region algorithm for the optimization of functions smoothed via Gaussian convolution using adaptive multiple importance sampling’, *SIAM J. Optim.* **28**, 1478–1507.

M. Marazzi and J. Nocedal (2002), ‘Wedge trust region methods for derivative free optimization’, *Math. Program.* **91**, 289–305.

A. March and K. Willcox (2012), ‘Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives’, *AIAA J.* **50**, 1079–1089.

D. W. Marquardt (1963), ‘An algorithm for least-squares estimation of nonlinear parameters’, *J. Soc. Ind. Appl. Math.* **11**, 431–441.

J. M. Martínez and F. N. C. Sobral (2012), ‘Constrained derivative-free optimization on thin domains’, *J. Global Optim.* **56**, 1217–1232.

J. Matyas (1965), ‘Random optimization’, *Automat. Remote Control* **26**, 246–253.

J. H. May (1974), Linearly constrained nonlinear programming: A solution method that does not require analytic derivatives. PhD thesis, Yale University.

J. H. May (1979), ‘Solving nonlinear programs without using analytic derivatives’, *Oper. Res.* **27**, 457–484.

R. L. McKinney (1962), ‘Positive bases for linear spaces’, *Trans. Amer. Math. Soc.* **103**, 131–131.

K. I. M. McKinnon (1998), ‘Convergence of the Nelder–Mead simplex method to a nonstationary point’, *SIAM J. Optim.* **9**, 148–158.

M. Menickelly (2017), Random models in nonlinear optimization. PhD thesis, Lehigh University.

M. Menickelly and S. M. Wild (2019), ‘Derivative-free robust optimization by outer approximations’, *Math. Program.* doi:10.1007/s10107-018-1326-9

A. G. Mersha and S. Dempe (2011), ‘Direct search algorithm for bilevel programming problems’, *Comput. Optim. Appl.* **49**, 1–15.

R. Mifflin (1975), ‘A superlinearly convergent algorithm for minimization without evaluating derivatives’, *Math. Program.* **9**, 100–117.

J. Mockus (1989), *Bayesian Approach to Global Optimization: Theory and Applications*, Springer.

J. J. Moré (1978), The Levenberg–Marquardt algorithm: Implementation and theory. In *Numerical Analysis: Dundee 1977* (G. A. Watson, ed.), Vol. 630 of Lecture Notes in Mathematics, Springer, pp. 105–116.

J. J. Moré and D. C. Sorensen (1983), ‘Computing a trust region step’, *SIAM J. Sci. Statist. Comput.* **4**, 553–572.

J. J. Moré and S. M. Wild (2009), ‘Benchmarking derivative-free optimization algorithms’, *SIAM J. Optim.* **20**, 172–191.

J. J. Moré and S. M. Wild (2011), ‘Estimating computational noise’, *SIAM J. Sci. Comput.* **33**, 1292–1314.

J. J. Moré and S. M. Wild (2012), ‘Estimating derivatives of noisy simulations’, *ACM Trans. Math. Software* **38**, 19:1–19:21.

J. J. Moré and S. M. Wild (2014), ‘Do you trust derivatives or differences?’, *J. Comput. Phys.* **273**, 268–277.

B. Morini, M. Porcelli and P. L. Toint (2018), ‘Approximate norm descent methods for constrained nonlinear systems’, *Math. Comput.* **87** (311), 1327–1351.

J. Müller (2016), ‘MISO: Mixed-integer surrogate optimization framework’, *Optim. Engng* **17**, 177–203.

J. Müller and M. Day (2019), ‘Surrogate optimization of computationally expensive black-box problems with hidden constraints’, *INFORMS J. Comput.*, to appear.

J. Müller and J. D. Woodbury (2017), ‘GOSAC: Global optimization with surrogate approximation of constraints’, *J. Global Optim.* **69**, 117–136.

J. Müller, C. A. Shoemaker and R. Piché (2013*a*), ‘SO-I: A surrogate model algorithm for expensive nonlinear integer programming problems including global optimization applications’, *J. Global Optim.* **59**, 865–889.

J. Müller, C. A. Shoemaker and R. Piché (2013*b*), 'SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems', *Comput. Oper. Res.* **40**, 1383–1400.

R. Munos (2011), Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor *et al.*, eds), Curran Associates, pp. 783–791.

S. G. Nash (2000), 'A multigrid approach to discretized optimization problems', *Optim. Methods Software* **14**, 99–116.

L. Nazareth and P. Tseng (2002), 'Gilding the lily: A variant of the Nelder–Mead algorithm based on golden-section search', *Comput. Optim. Appl.* **22**, 133–144.

J. A. Nelder and R. Mead (1965), 'A simplex method for function minimization', *Comput. J.* **7**, 308–313.

Y. Nesterov (2004), *Introductory Lectures on Convex Optimization: A Basic Course*, Vol. 87 of Applied Optimization, Springer.

Y. Nesterov and V. Spokoiny (2017), 'Random gradient-free minimization of convex functions', *Found. Comput. Math.* **17**, 527–566.

A. Neumaier (2004), Complete search in continuous global optimization and constraint satisfaction. In *Acta Numerica*, Vol. 13, Cambridge University Press, pp. 271–369.

A. Neumaier, H. Fendl, H. Schilly and T. Leitner (2011), 'VXQR: Derivative-free unconstrained optimization based on QR factorizations', *Soft Comput.* **15**, 2287–2298.

E. Newby and M. M. Ali (2015), 'A trust-region-based derivative free algorithm for mixed integer programming', *Comput. Optim. Appl.* **60**, 199–229.

R. Oeuvray (2005), Trust-region methods based on radial basis functions with application to biomedical imaging. PhD thesis, EPFL.

R. Oeuvray and M. Bierlaire (2009), 'BOOSTERS: A derivative-free algorithm based on radial basis functions', *Internat. J. Model. Simul.* **29**, 26–36.

P.-M. Olsson (2014), Methods for network optimization and parallel derivative-free optimization. PhD thesis, Linköping University.

E. O. Omojokun (1989), Trust region algorithms for optimization with nonlinear equality and inequality constraints. PhD thesis, University of Colorado at Boulder.

C. Paquette and K. Scheinberg (2018), A stochastic line search method with convergence rate analysis. arXiv:1807.07994

R. Pasupathy (2010), 'On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization', *Oper. Res.* **58**, 889–901.

R. Pasupathy, P. Glynn, S. Ghosh and F. S. Hashemi (2018), 'On sampling rates in simulation-based recursions', *SIAM J. Optim.* **28**, 45–73.

N. R. Patel, R. L. Smith and Z. B. Zabinsky (1989), 'Pure adaptive search in Monte Carlo optimization', *Math. Program.* **43**, 317–328.

G. Peckham (1970), 'A new method for minimising a sum of squares without calculating gradients', *Comput. J.* **13**, 418–420.

V. Picheny, R. B. Gramacy, S. M. Wild and S. Le Digabel (2016), Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. In *Advances in Neural Information Processing Systems 29* (D. D. Lee *et al.*, eds), Curran Associates, pp. 1435–1443.

T. D. Plantenga (2009), HOPSPACK 2.0 user manual. Technical report SAND-2009-6265, Sandia National Laboratories.

E. Polak (1971), *Computational Methods in Optimization: A Unified Approach*, Academic Press.

E. Polak and M. Wetter (2006), 'Precision control for generalized pattern search algorithms with adaptive precision function evaluations', *SIAM J. Optim.* **16**, 650–669.

B. T. Polyak (1987), *Introduction to Optimization*, Optimization Software.

M. Porcelli and P. L. Toint (2017), 'BFO: A trainable derivative-free brute force optimizer for nonlinear bound-constrained optimization and equilibrium computations with continuous and discrete variables', *ACM Trans. Math. Software* **44**, 1–25.

T. Pourmohamad (2016), Combining multivariate stochastic process models with filter methods for constrained optimization. PhD thesis, UC Santa Cruz.

M. J. D. Powell (1964), 'An efficient method for finding the minimum of a function of several variables without calculating derivatives', *Comput. J.* **7**, 155–162.

M. J. D. Powell (1965), 'A method for minimizing a sum of squares of non-linear functions without calculating derivatives', *Comput. J.* **7**, 303–307.

M. J. D. Powell (1975), 'A view of unconstrained minimization algorithms that do not require derivatives', *ACM Trans. Math. Software* **1**, 97–107.

M. J. D. Powell (1994), A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis* (S. Gomez and J. P. Hennart, eds), Vol. 275 of Mathematics and its Applications, Springer, pp. 51–67.

M. J. D. Powell (1997), Trust region calculations revisited. In *Numerical Analysis 1997* (D. F. Griffiths *et al.*, eds), Vol. 380 of Pitman Research Notes in Mathematics, Addison Wesley Longman, pp. 193–211.

M. J. D. Powell (1998*a*), Direct search algorithms for optimization calculations. In *Acta Numerica*, Vol. 7, Cambridge University Press, pp. 287–336.

M. J. D. Powell (1998*b*), The use of band matrices for second derivative approximations in trust region algorithms. In *Advances in Nonlinear Programming* (Y. Yuan, ed.), Vol. 14 of Applied Optimization, Springer, pp. 3–28.

M. J. D. Powell (2001), 'On the Lagrange functions of quadratic models that are defined by interpolation', *Optim. Methods Software* **16**, 289–309.

M. J. D. Powell (2002), 'UOBYQA: Unconstrained optimization by quadratic approximation', *Math. Program.* **92**, 555–582.

M. J. D. Powell (2003), 'On trust region methods for unconstrained minimization without derivatives', *Math. Program.* **97**, 605–623.

M. J. D. Powell (2004*a*), 'Least Frobenius norm updating of quadratic models that satisfy interpolation conditions', *Math. Program.* **100**, 183–215.

M. J. D. Powell (2004*b*), 'On the use of quadratic models in unconstrained minimization without derivatives', *Optim. Methods Software* **19**, 399–411.

M. J. D. Powell (2004*c*), On updating the inverse of a KKT matrix. Technical report DAMTP 2004/NA01, University of Cambridge.

M. J. D. Powell (2006), The NEWUOA software for unconstrained optimization without derivatives. In *Large-Scale Nonlinear Optimization* (G. Di Pillo and M. Roma, eds), Vol. 83 of Nonconvex Optimization and its Applications, Springer, pp. 255–297.

M. J. D. Powell (2007), A view of algorithms for optimization without derivatives. Technical report DAMTP 2007/NA03, University of Cambridge.

M. J. D. Powell (2008), 'Developments of NEWUOA for minimization without derivatives', *IMA J. Numer. Anal.* **28**, 649–664.

M. J. D. Powell (2009), The BOBYQA algorithm for bound constrained optimization without derivatives. Technical report DAMTP 2009/NA06, University of Cambridge.

M. J. D. Powell (2010), 'On the convergence of a wide range of trust region methods for unconstrained optimization', *IMA J. Numer. Anal.* **30**, 289–301.

M. J. D. Powell (2012), 'On the convergence of trust region algorithms for unconstrained minimization without derivatives', *Comput. Optim. Appl.* **53**, 527–555.

M. J. D. Powell (2013), 'Beyond symmetric Broyden for updating quadratic models in minimization without derivatives', *Math. Program.* **138**, 475–500.

M. J. D. Powell (2015), 'On fast trust region methods for quadratic models with linear constraints', *Math. Program. Comput.* **7**, 237–267.

W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery (2007), *Numerical Recipes in Fortran: The Art of Scientific Computing*, third edition, Cambridge University Press.

C. J. Price and P. L. Toint (2006), 'Exploiting problem structure in pattern search methods for unconstrained optimization', *Optim. Methods Software* **21**, 479–491.

C. J. Price, I. D. Coope and D. Byatt (2002), 'A convergent variant of the Nelder–Mead algorithm', *J. Optim. Theory Appl.* **113**, 5–19.

M. L. Ralston and R. I. Jennrich (1978), 'Dud: A derivative-free algorithm for nonlinear least squares', *Technometrics* **20**, 7–14.

K. Rashid, S. Ambani and E. Cetinkaya (2012), 'An adaptive multiquadric radial basis function method for expensive black-box mixed-integer nonlinear constrained optimization', *Engng Optim.* **45**, 185–206.

L. A. Rastrigin (1963), 'The convergence of the random search method in the extremal control of many-parameter system', *Automat. Remote Control* **24**, 1337–1342.

S. J. Reddi, A. Hefny, S. Sra, B. Poczos and A. Smola (2016), Stochastic variance reduction for nonconvex optimization. In *33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds). *Proc. Mach. Learn. Res.* **48**, 314–323.

J. Regier, M. I. Jordan and J. McAuliffe (2017), Fast black-box variational inference through stochastic trust-region optimization. In *Advances in Neural Information Processing Systems 30* (I. Guyon *et al.*, eds), Curran Associates, pp. 2402–2411.

R. G. Regis (2013), 'Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points', *Engng Optim.* **46**, 218–243.

R. G. Regis (2015), 'The calculus of simplex gradients', *Optim. Lett.* **9**, 845–865.

R. G. Regis (2016), 'On the properties of positive spanning sets and positive bases', *Optim. Engng* **17**, 229–262.

R. G. Regis and C. A. Shoemaker (2007), 'A stochastic radial basis function method for the global optimization of expensive functions', *INFORMS J. Comput.* **19**, 457–509.

R. G. Regis and S. M. Wild (2017), 'CONORBIT: Constrained optimization by radial basis function interpolation in trust regions', *Optim. Methods Software* **32**, 552–580.

A. H. G. Rinnooy Kan and G. T. Timmer (1987*a*), 'Stochastic global optimization methods, I: Clustering methods', *Math. Program.* **39**, 27–56.

A. H. G. Rinnooy Kan and G. T. Timmer (1987*b*), 'Stochastic global optimization methods, II: Multi level methods', *Math. Program.* **39**, 57–78.

L. M. Rios and N. V. Sahinidis (2013), 'Derivative-free optimization: A review of algorithms and comparison of software implementations', *J. Global Optim.* **56**, 1247–1293.

H. Robbins (1952), 'Some aspects of the sequential design of experiments', *Bull. Amer. Math. Soc.* **58**, 527–536.

H. Robbins and S. Monro (1951), 'A stochastic approximation method', *Ann. Math. Statist.* **22**, 400–407.

R. Rockafellar and R. J.-B. Wets (2009), *Variational Analysis*, Springer.

H. H. Rosenbrock (1960), 'An automatic method for finding the greatest or least value of a function', *Comput. J.* **3**, 175–184.

D. Ruppert (1991), Stochastic approximation. In *Handbook of Sequential Analysis* (B. K. Ghosh and P. K. Sen, eds), Vol. 118 of Statistics: A Series of Textbooks and Monographs, CRC Press, pp. 503–529.

D. Russo and B. Van Roy (2016), 'An information-theoretic analysis of Thompson sampling', *J. Mach. Learn. Res.* **17**, 2442–2471.

A. S. Rykov (1980), 'Simplex direct search algorithms', *Automat. Remote Control* **41**, 784–793.

J. H. Ryu and S. Kim (2014), 'A derivative-free trust-region method for biobjective optimization', *SIAM J. Optim.* **24**, 334–362.

J. Sacks (1958), 'Asymptotic distribution of stochastic approximation procedures', *Ann. Math. Statist.* **29**, 373–405.

A. Saha and A. Tewari (2011), Improved regret guarantees for online smooth convex optimization with bandit feedback. In *14th International Conference on Artifical Intelligence and Statistics (AISTATS). Proc. Mach. Learn. Res.* **15**, 636–642.

P. R. Sampaio and P. L. Toint (2015), 'A derivative-free trust-funnel method for equality-constrained nonlinear optimization', *Comput. Optim. Appl.* **61**, 25–49.

P. R. Sampaio and P. L. Toint (2016), 'Numerical experience with a derivative-free trust-funnel method for nonlinear optimization problems with general nonlinear constraints', *Optim. Methods Software* **31**, 511–534.

S. Sankaran, C. Audet and A. L. Marsden (2010), 'A method for stochastic constrained optimization using derivative-free surrogate pattern search and collocation', *J. Comput. Phys.* **229**, 4664–4682.

K. Scheinberg and P. L. Toint (2010), 'Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization', *SIAM J. Optim.* **20**, 3512–3532.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams and N. de Freitas (2016), 'Taking the human out of the loop: A review of Bayesian optimization', *Proc. IEEE* **104**, 148–175.

O. Shamir (2013), On the complexity of bandit and derivative-free stochastic convex optimization. In *26th Annual Conference on Learning Theory (COLT 2013)* (S. Shalev-Shwartz and I. Steinwart, eds). *Proc. Mach. Learn. Res.* **30**, 3–24.

O. Shamir (2017), 'An optimal algorithm for bandit and zero-order convex optimization with two-point feedback', *J. Mach. Learn. Res.* **18** (52), 1–11.

S. Shashaani, F. S. Hashemi and R. Pasupathy (2018), 'ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free stochastic optimization', *SIAM J. Optim.* **28**, 3145–3176.

S. Shashaani, S. R. Hunter and R. Pasupathy (2016), ASTRO-DF: Adaptive sampling trust-region optimization algorithms, heuristics, and numerical experience. In *2016 Winter Simulation Conference (WSC)*, IEEE.

C. A. Shoemaker and R. G. Regis (2003), MAPO: using a committee of algorithm-experts for parallel optimization of costly functions. In *Proceedings of the ACM Symposium on Parallel Algorithms and Architectures*, ACM, pp. 242–243.

J. C. Spall (1992), 'Multivariate stochastic approximation using a simultaneous perturbation gradient approximation', *IEEE Trans. Automat. Control* **37**, 332–341.

J. C. Spall (2005), *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, Wiley.

W. Spendley (1969), Nonlinear least squares fitting using a modified simplex minimization method. In *Optimization* (R. Fletcher, ed.), Academic Press, pp. 259–270.

W. Spendley, G. R. Hext and F. R. Himsworth (1962), 'Sequential application of simplex designs in optimisation and evolutionary operation', *Technometrics* **4**, 441–461.

N. Srebro, K. Sridharan and A. Tewari (2011), On the universality of online mirror descent. In *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor *et al.*, eds), Curran Associates, pp. 2645–2653.

T. A. Sriver, J. W. Chrissis and M. A. Abramson (2009), 'Pattern search ranking and selection algorithms for mixed variable simulation-based optimization', *European J. Oper. Res.* **198**, 878–890.

S. U. Stich, C. L. Müller and B. Gärtner (2013), 'Optimization of convex functions with random pursuit', *SIAM J. Optim.* **23**, 1284–1309.

M. Taddy, H. K. H. Lee, G. A. Gray and J. D. Griffin (2009), 'Bayesian guided pattern search for robust local optimization', *Technometrics* **51**, 389–401.

V. Torczon (1991), ‘On the convergence of the multidirectional search algorithm’, *SIAM J. Optim.* **1**, 123–145.

V. Torczon (1997), ‘On the convergence of pattern search algorithms’, *SIAM J. Optim.* **7**, 1–25.

A. Törn and A. Žilinskas (1989), *Global Optimization*, Springer.

A. Tröltzsch (2016), ‘A sequential quadratic programming algorithm for equality-constrained optimization without derivatives’, *Optim. Lett.* **10**, 383–399.

P. Tseng (1999), ‘Fortified-descent simplicial search method: A general approach’, *SIAM J. Optim.* **10**, 269–288.

M. Valko, A. Carpentier and R. Munos (2013), Stochastic simultaneous optimistic optimization. In *30th International Conference on Machine Learning (ICML 13). Proc. Mach. Learn. Res.* **28**, 19–27.

B. Van Dyke and T. J. Asaki (2013), ‘Using QR decomposition to obtain a new instance of mesh adaptive direct search with uniformly distributed polling directions’, *J. Optim. Theory Appl.* **159**, 805–821.

F. Vanden Berghen (2004), CONDOR: A constrained, non-linear, derivative-free parallel optimizer for continuous, high computing load, noisy objective functions. PhD thesis, Université Libre de Bruxelles.

F. Vanden Berghen and H. Bersini (2005), ‘CONDOR: A new parallel, constrained extension of Powell’s UOBYQA algorithm: Experimental results and comparison with the DFO algorithm’, *J. Comput. Appl. Math.* **181**, 157–175.

A. Verdério, E. W. Karas, L. G. Pedroso and K. Scheinberg (2017), ‘On the construction of quadratic models for derivative-free trust-region algorithms’, *EURO J. Comput. Optim.* **5**, 501–527.

L. N. Vicente (2013), ‘Worst case complexity of direct search’, *EURO J. Comput. Optim.* **1**, 143–153.

L. N. Vicente and A. Custódio (2012), ‘Analysis of direct searches for discontinuous functions’, *Math. Program.* **133**, 299–325.

K. K. Vu, C. D’Ambrosio, Y. Hamadi and L. Liberti (2016), ‘Surrogate-based methods for black-box optimization’, *Internat. Trans. Oper. Res.* **24**, 393–424.

Y. Wang, S. S. Du, S. Balakrishnan and A. Singh (2018), Stochastic zeroth-order optimization in high dimensions. In *21st International Conference on Artificial Intelligence and Statistics* (A. Storkey and F. Perez-Cruz, eds). *Proc. Mach. Learn. Res.* **84**, 1356–1365.

H. Wendland (2005), *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press.

S. M. Wild (2008*a*), Derivative-free optimization algorithms for computationally expensive functions. PhD thesis, Cornell University.

S. M. Wild (2008*b*), MNH: A derivative-free optimization algorithm using minimal norm Hessians. In *Tenth Copper Mountain Conference on Iterative Methods.* http://grandmaster.colorado.edu/~copper/2008/SCWinners/Wild.pdf

S. M. Wild (2017), Solving derivative-free nonlinear least squares problems with POUNDERS. In *Advances and Trends in Optimization with Engineering Applications* (T. Terlaky *et al.*, eds), SIAM, pp. 529–540.

S. M. Wild and C. A. Shoemaker (2011), ‘Global convergence of radial basis function trust region derivative-free algorithms’, *SIAM J. Optim.* **21**, 761–781.

S. M. Wild and C. A. Shoemaker (2013), 'Global convergence of radial basis function trust-region algorithms for derivative-free optimization', *SIAM Review* **55**, 349–371.

S. M. Wild, R. G. Regis and C. A. Shoemaker (2008), 'ORBIT: Optimization by radial basis function interpolation in trust-regions', *SIAM J. Sci. Comput.* **30**, 3197–3219.

D. H. Winfield (1969), Function and functional optimization by interpolation in data tables. PhD thesis, Harvard University.

D. H. Winfield (1973), 'Function minimization by interpolation in a data table', *J. Inst. Math. Appl.* **12**, 339–347.

D. J. Woods (1985), An interactive approach for solving multi-objective optimization problems. PhD thesis, Rice University.

M. H. Wright (1995), Direct search methods: Once scorned, now respectable. In *Numerical Analysis 1995* (D. F. Griffiths and G. A. Watson, eds), Vol. 344 of Pitman Research Notes in Mathematics, Addison Wesley Longman, pp. 191–208.

S. Xiong, P. Z. G. Qian and C. F. J. Wu (2013), 'Sequential design and analysis of high-accuracy and low-accuracy computer codes', *Technometrics* **55**, 37–46.

J. Xu and L. Zikatanov (2017), Algebraic multigrid methods. In *Acta Numerica*, Vol. 26, Cambridge University Press, pp. 591–721.

W.-C. Yu (1979), 'Positive basis and a class of direct search techniques', *Scientia Sinica, Special Issue of Mathematics* **1**, 53–67.

Y.-X. Yuan (1985), 'Conditions for convergence of trust region algorithms for nonsmooth optimization', *Math. Program.* **31**, 220–228.

Z. B. Zabinsky and R. L. Smith (1992), 'Pure adaptive search in global optimization', *Math. Program.* **53**, 323–338.

D. Zhang and G.-H. Lin (2014), 'Bilevel direct search method for leader–follower problems and application in health insurance', *Comput. Oper. Res.* **41**, 359–373.

H. Zhang and A. R. Conn (2012), 'On the local convergence of a derivative-free algorithm for least-squares minimization', *Comput. Optim. Appl.* **51**, 481–507.

H. Zhang, A. R. Conn and K. Scheinberg (2010), 'A derivative-free algorithm for least-squares minimization', *SIAM J. Optim.* **20**, 3555–3576.

L. Zhang, T. Yang, R. Jin and Z.-H. Zhou (2015), Online bandit learning for a special class of non-convex losses. In *29th AAAI Conference on Artificial Intelligence (AAAI '15)*, AAAI Press, pp. 3158–3164.

Z. Zhang (2014), 'Sobolev seminorm of quadratic functions with applications to derivative-free optimization', *Math. Program.* **146**, 77–96.

A. A. Zhigljavsky (1991), *Theory of Global Random Search*, Springer.