Algorithm XXX: QNSTOP—Quasi-Newton Algorithm for Stochastic Optimization

BRANDON D. AMOS, DAVID R. EASTERLING, LAYNE T. WATSON
Virginia Polytechnic Institute and State University
WILLIAM I. THACKER
Winthrop University
and
BRENT S. CASTLE, MICHAEL W. TROSSET
Indiana University

QNSTOP consists of serial and parallel (OpenMP) Fortran 2003 codes for the quasi-Newton stochastic optimization method of Castle and Trosset for stochastic search problems. A complete description of QNSTOP for both local search with stochastic objective and global search with "noisy" deterministic objective is given here for the first time. For stochastic search problems, some convergence theory exists for particular algorithmic choices and parameter values. Both the parallel driver subroutine, which offers several parallel decomposition strategies, and the serial driver subroutine can be used for local stochastic search or global deterministic search, based on an input switch. Some performance data for computational systems biology problems is given.

Categories and Subject Descriptors: J.2 [**Computer Applications**]: Physical Science and Engineering — *Mathematics*; G.3 [**Mathematics of Computing**]: Probability and Statistics; G.4 [**Mathematics of Computing**]: Mathematical Software

General Terms: Algorithms, Design, Documentation

Additional Key Words and Phrases: derivative-free, deterministic global optimization, quasi-Newton, response surface methodology, stochastic search

This work was supported in part by Air Force Research Laboratory Grant FA8650-09-2-3938 and Air Force Office of Scientific Research Grant FA9550-09-1-0153.
Authors' addresses: B. D. Amos, D. R. Easterling, Department of Computer Science, L. T. Watson, Departments of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, VA 24061; e-mail: `bdamos@vt.edu`, {`dreast, ltw`}`@cs.vt.edu`; W. I. Thacker, Computer Science Department, Winthrop University, Rock Hill, SC 29733; e-mail: `thackerw@winthrop.edu`; B. S. Castle, School of Informatics and Computing, M. W. Trosset, Department of Statistics, Indiana University, Bloomington, IN 47405; e-mail: `mtrosset@indiana.edu`.

1. INTRODUCTION

Consider the problem of minimizing an objective function $f : \mathbb{R}^p \to \mathbb{R}$, subject to simple bound constraints $x \in B = \{x \in \mathbb{R}^p \mid \ell \leq x \leq u\}$, where ℓ, $u \in \mathbb{R}^p$, $\ell < u$. The algorithm described herein (QNSTOP) was studied by Castle [2012] in the setting of stochastic search, i.e., $f(x)$ cannot be computed and must be estimated via random sampling. Its utility for global optimization of noisy deterministic functions, i.e., $f(x)$ can be computed but f has large local total variation, was subsequently studied by Easterling et al. [2014].

QNSTOP was inspired by a connection, first noted by Trosset [2003], between ridge analysis, a popular technique in response surface methodology, and trust region methods for numerical optimization. It is quasi-Newton in the sense that it constructs local quadratic models of the objective function that are not derived from Taylor polynomials and derivatives. It was conceived for optimization problems in which evaluation of the objective function is stochastic rather than deterministic.

Attempts to incorporate trust-region methods into response surface methodology have been relatively rare and have lacked corresponding convergence theory. Castle [2012] adapted standard convergence theory for stochastic approximation to ensure convergence under certain conditions. This convergence theory motivated a novel constrained variation of the SR1 Hessian update and a novel strategy for updating the trust region that is qualitatively different from the standard one. Both ideas are described in detail in Section 3.

Although QNSTOP was conceived as a method for stochastic search, it has two features that make it attractive for globally optimizing noisy deterministic objectives. First, because QNSTOP smooths (by regression) observed responses to construct semilocal approximations, it automatically filters high-frequency oscillations in the objective. Second, QNSTOP uses space-filling designs in ellipsoidal trust regions to obtain semilocal information about the objective. In so doing, QNSTOP may serendipitously discover unexpectedly small objective values within the semilocal trust region $E_k(\tau_k)$.

An early version of Castle's algorithm for stochastic search was named QNSTOP and was implemented in Matlab. Positing that global optimization of noisy deterministic objective functions had some similarities to stochastic search, the original QNSTOP code was converted to Fortran 2003 and used for deterministic global optimization. Computational experience and considerations of numerical stability, efficiency, and convergence rate led to major modifications in the linear algebra, trust region update strategy, Hessian update, and start phase, resulting in two distinct versions of QNSTOP: one for stochastic search that relies on the convergence theory of Castle [2012], and one for deterministic global optimization problems that is based on the aforementioned considerations. Easterling et al. [2014] described the deterministic global variant of QNSTOP and presented results for several deterministic global optimization problems, but did not address QNSTOP parallelization or present the stochastic search variant of QNSTOP. Because these variants of QNSTOP have a common ancestry, and some calculations in common, QNSTOP is

viewed as a family of algorithms, with the variant being selected by a switch in a single Fortran subroutine. The short conference paper by Amos et al. [2014] gave an abbreviated description of QNSTOP with some performance results. The present paper contains the first complete description of the full (stochastic and global deterministic options) QNSTOP code.

The following sections provide background, discuss varying philosophies of stochastic search, describe QNSTOP in detail, provide some performance data on difficult systems biology problems, and conclude with some general observations.

2. OPTIMIZATION IN THE PRESENCE OF RANDOM NOISE

QNSTOP was conceived for optimization problems in which evaluation of the objective function is corrupted by the presence of random noise. To distinguish between the presence and absence of random noise, researchers in numerical optimization often distinguish *stochastic optimization* from *deterministic optimization*. The acronym QNSTOP uses the former phrase in this relatively narrow sense. Other communities use it more generally. For example, Powell [2019] identified 15 distinct stochastic optimization communities, including *stochastic search*, which he proposed as one of four meta-classes for stochastic optimization. In his taxonomy, stochastic approximation, response surface methodology, and QNSTOP are all examples of stochastic search. Note that stochastic search may also involve probabilistic constraints, which are not considered here.

Other approaches are possible. One might generate a very large number of realizations and create a discrete approximation of the original problem, then apply deterministic algorithms. This approach, sometimes called *sample path optimization*, can be highly effective on some problems but relies on information that is not likely to be available for the problems of interest here. Similarly, techniques for *stochastic programming*, which generate large deterministic problems from multiple scenarios, seem ill-suited for the problem types considered here. Spall [2003, Section 15.4] discusses sample path optimization; Birge and Louveaux [2011] provide a comprehensive account of stochastic programming.

2.1 Mathematical Formulation

To illustrate the phenomenon of random noise, suppose that one seeks to minimize $\mu : \mathbb{R}^p \to \mathbb{R}$. Given $x \in \mathbb{R}^p$, one would like to observe $\mu(x)$; instead, one observes $\mu(x) + \epsilon_x$, where ϵ_x is a random variable. This is the case of additive random noise. In this case, the underlying objective function μ is often called the regression function (in the stochastic approximation literature) or the response surface (in the response surface methodology literature).

Most formulations of optimization in the presence of random noise impose various assumptions on the ϵ_x. The assumption $E(\epsilon_x) = 0$, from which it follows that the expected value of an observation is the true value of the objective function, is inevitable. One might also assume that $\epsilon_x \sim \text{Normal}(0, \sigma_x^2)$ (normality), that

$\mathrm{Var}(\epsilon_x) = \sigma_x^2$ does not depend on x (homoscedasticity), and that the ϵ_x are independent (white noise). The preceding set of assumptions is referred to as the *standard example*.

Random noise may not be additive. Because there is no elegant way to catalog the many random mechanisms by which a deterministic objective function might be corrupted, the concept of optimizing in the presence of random noise is somewhat elusive. The usual approach is to begin with what one observes, not with what one seeks to optimize. Given $x \in \mathbb{R}^p$, suppose that one observes a random variable Y_x. One then *defines* the objective function to be $\mu(x) = E(Y_x)$. However, there are a number of meaningful problems that are more naturally expressed in a slightly different setting.

Let

$$\mathcal{P} = \{P(\cdot; x) \mid x \in \mathcal{C} \subseteq \mathbb{R}^p\}$$

denote a family of probability distributions indexed by x. Assume that the $P(\cdot; x)$ are completely unknown or analytically intractable, but that one can sample from any specified $P(\cdot; x)$. The first case might arise as one varies the prescribed operating characteristics of a manufacturing facility in search of an optimum. This is a typical concern of response surface methodology. In this case, observations are generated by a physical process for which a formal mathematical description is not available. The second case might arise when one is tuning the parameters of a simulated stochastic process, searching for settings that produce simulated data sets that resemble an actual data set. This is a useful approach to parameter estimation when the statistical model is defined implicitly, i.e., in terms of a generating stochastic mechanism rather than by specifying a parametric family of probability distributions. See, for example, Atkinson et al. [1983], Diggle and Gratton [1984], and Thompson [2000]. In none of these cases can one manipulate the $P(\cdot; x)$ as mathematical objects; instead one must rely on random sampling to obtain information about them.

Now let $T : \mathcal{P} \to \mathbb{R}$ and let $f(x) = T(P(\cdot; x))$. One seeks local solutions of

$$\min_{P \in \mathcal{P}} T(P), \tag{1}$$

or, equivalently, of

$$\min_{x \in \mathcal{C}} f(x). \tag{2}$$

Additional smoothness assumptions are imposed on T or f, as needed.

As stated, Problems (1) and (2) are unambiguous, deterministic optimization problems. They become stochastic when one cannot manipulate the $P(\cdot; x)$ as mathematical objects. When one must estimate $f(x) = T(P(\cdot; x))$ from a random sample

$$\omega_1(x), \ldots, \omega_n(x) \sim P(\cdot; x), \tag{3}$$

then function evaluation is random and Problems (1) and (2) necessitate optimization in the presence of random noise.

Given an independent and identically distributed random sample (3), let

$$\hat{P}_n(\cdot; x) = \sum_{i=1}^{n} \frac{1}{n} \delta_{\omega_i(x)} \tag{4}$$

denote the empirical distribution of the sample, i.e., the discrete probability distribution that assigns probability $1/n$ to each $\omega_i(x)$, where

$$\delta_t(s) = \chi_{\{t\}}(s) = \begin{cases} 1, & s = t, \\ 0, & s \neq t. \end{cases}$$

In the case of univariate probability distributions, the empirical distribution is usually identified as the empirical cumulative distribution function (cdf), i.e., the function (of y)

$$\hat{P}_n\left(\omega(x) \leq y; x\right) = \frac{\#\{\omega_i(x) \leq y\}}{n}.$$

Let $T_n(\omega_1(x), \ldots, \omega_n(x))$ denote a statistic, i.e., a real-valued quantity calculated from the sample. Then von Mises [1947] observed that many useful statistics are of the form

$$T_n\left(\omega_1(x), \ldots, \omega_n(x)\right) = T\left(\hat{P}_n(\cdot; x)\right),$$

in which context T is often called a statistical functional.

In what follows, the univariate probability distributions $P(\cdot; x)$ and $\hat{P}_n(\cdot; x)$ are identified with their corresponding cumulative distribution functions. Let

$$D_n = \sup_y \left| \hat{P}_n\left(\omega(x) \leq y; x\right) - P\left(\omega(x) \leq y; x\right) \right|.$$

The Glivenko-Cantelli Theorem states that $P\left(D_n \to 0\right) = 1$; hence, if T is continuous in a suitable sense, then one should find that

$$T\left(\hat{P}_n\left(\cdot; x\right)\right) \xrightarrow{P} T\left(P\left(\cdot; x\right)\right).$$

This says that one can consistently estimate $f(x) = T(P(\cdot; x))$ by sampling from $P(\cdot; x)$. In fact, one can usually say considerably more. The theory of statistical functionals is primarily concerned with connecting the differentiability of T to the asymptotic normality of $T(\hat{P}_n(\cdot; x))$. See, for example, Fernholz [1983].

2.2 Stochastic Approximation Versus Response Surface Methodology

QNSTOP borrows ideas from two standard approaches to optimization in the presence of random noise, stochastic approximation (SA) and response surface methodology (RSM). Both SA and RSM originated in the early 1950s. For SA, seminal papers include Robbins and Monro [1951], Kiefer and Wolfowitz [1952], Blum [1954], and Dvoretsky [1956]. See Kushner and Yin [1997], Spall [2003], and Marti [2005]

for modern surveys. For RSM, the seminal paper is Box and Wilson [1951]. See Myers and Montgomery [1995] for a modern survey.

Both SA and RSM evolved from attempts to adapt the method of steepest descent for numerical optimization. Both approaches construct local models (typically linear, but occasionally quadratic) of the objective function. Because the objective function cannot be manipulated directly, derivatives are not available and cannot be used to construct the local models. SA constructs local models from estimated derivatives, obtained by finite differencing. RSM constructs local models directly, from designed regression experiments.

In numerical optimization, the magnitude of the differences used in finite differencing schemes is extremely small. When function evaluation is corrupted by random noise, trends in the objective function cannot be detected with such small differences. Furthermore, once a descent direction has been estimated, line searches cannot reliably determine an optimal step length. As a result, SA relies on predetermined decreasing sequences of differences and step length multipliers. Convergence to a local solution is guaranteed by controlling the behavior of these sequences. Traditionally, the differences are $\mathcal{O}(1/k^3)$ and the step length multipliers are $\mathcal{O}(1/k)$, where k is the iteration counter.

SA relies on averaging. The models constructed for individual iterations may be quite crude (Spall's simultaneous perturbation stochastic approximation algorithm estimates a gradient from just two function evaluations); SA succeeds by taking a large number of steps. For fixed budgets, it may be better to choose $n = 1$ in (3) and take a great many steps than to choose $n \gg 1$ and settle for fewer steps of higher quality. One of the most significant advances in SA is due to Polyak and Juditsky [1992], who demonstrated that convergence could by accelerated by using larger step length multipliers and averaging the sequence of iterates.

In contrast, RSM typically takes a small number of carefully chosen steps. Whereas SA has produced a huge literature on asymptotic convergence theory, RSM has produced a huge literature on experimental design. There is virtually no overlap between the SA and RSM literatures.

2.3 Trust-Region Methods in Response Surface Methodology

As described by Conn et al. [2000], trust-region methods are widely used in deterministic optimization. In their seminal work on RSM, Box and Wilson [1951] recommended use of a primitive trust region. Trosset [2003] noted the essential equivalence of pioneering work in trust-region methods by Marquardt [1963] and in ridge analysis by Draper [1963], but attempts to incorporate modern trust-region methods into RSM have been infrequent.

QNSTOP is distinct from, but closely related to, (at least) three previous trust-region methods for stochastic search. First, Lawera and Thompson [1993] described a response surface method based on ideas in [Box and Hunter 1957]. Significant innovations include adaptive experimental designs and quasi-trust region step length control.

Second, Deng and Ferris [2006] proposed three novel modifications to Powell's [2002] UOBYQA (unconstrained optimization by quadratic approximation) algorithm for numerical optimization, endeavoring to adapt it for stochastic search. Their algorithm observes the response at each design site multiple times and interpolates the mean responses. A heuristic is used to determine how many observations should be taken at each design site so that the quadratic model and the constrained minimizer are stable. The constrained minimizer of the quadratic model is computed in the same way as in UOBYQA; however, a novel heuristic is used to decide whether to update the current iterate with the minimizer or leave it unchanged. They also describe termination criteria specific to the stochastic setting based upon having similar mean responses amongst a large portion of sites on the boundary of the trust region.

Finally, Chang, Hong, and Wan [2007] and Chang and Wan [2009] proposed the STRONG and STRONG-X algorithms. STRONG assumes normally distributed function evaluation errors, while STRONG-X relaxes this assumption to additive errors with bounded variance. Both algorithms adapt the standard two-phase RSM approach and utilize trust regions to control progress. The first phase constructs a linear model fit partially by least squares to multiple observations at design sites in an appropriate design (the authors recommend a fractional factorial or factorial design plus the current iterate). A line search is used in the direction of negative gradient within the trust region to choose the subsequent iterate. The second phase constructs a quadratic model by least squares. If sufficient progress is made, the algorithm steps to the Cauchy point, i.e., the minimizer of the quadratic in the direction of steepest descent subject to the trust region constraint. Heuristics are used to determine whether sufficient progress was obtained in each phase.

What distinguishes the above methods from the trust-region methods for derivative-free optimization (DFO) pioneered by Conn et al. [1997] is the effort to address the difficulties posed by the presence of random noise. DFO and UOBYQA interpolate observed function values, which seems unappetizing in the presence of random noise. Each of the above methods attempts to smooth the noise: Deng and Ferris observe multiple function values at each design site and interpolate the mean responses, whereas the other methods (and QNSTOP) smooth by regression. Simulation experiments performed by Castle [2012, Section 5.2] indicate that the latter approach is superior, or—more precisely—that "it might be beneficial to observe the objective at an unobserved design site rather than observing replications at a particular design site."

2.4 Optimization of Noisy Deterministic Functions

Besides the myriad instances where the objective function $f(x)$ is a random variable, there are optimization problems where the function $f(x)$ is deterministic but has very large local total variation, making $f(x)$ appear to be stochastic as x changes slightly. Large-scale scientific computations involving iterative adaptive algorithms often exhibit this latter type of behavior, e.g., the biomechanics gait model in [Easterling et al. 2014].

The high-frequency oscillations in noisy deterministic functions mean that derivative information is rarely useful, and that local comparisons of individual function values may fail to capture semilocal trends. When optimizing a noisy deterministic function, the need to filter these oscillations is not unlike the need to smooth random noise in stochastic search. Indeed, SA, which relies on coarse finite-differencing, is often recommended for optimizing noisy deterministic functions. Spall's Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm [Spall 1987, 1992, 1998], which constructs gradient estimates from just two function evaluations, has been widely used for this purpose. Kelley's implicit filtering algorithm [Gilmore and Kelley 1995], which relies on coarse stencil-based finite differencing to construct descent directions, is predicated on the same reasoning. Alternatively, one might attempt to filter high-frequency oscillations by RSM.

Both SA and RSM attempt to find local minimizers of an objective function. However successful they may be in filtering noise and identifying underlying trend, they are not designed to find the global minimizer of the underlying trend. If global optimization is desired, then SA or RSM can be used in conjunction with a sensible multistart strategy, e.g., by generating initial starting points via Latin hypercube (LHC) sampling. Easterling et al. [2014] found QNSTOP with an LHC multistart strategy to be highly effective on certain types of global optimization problems. However, as described in Section 3, effective global optimization of noisy deterministic functions calls for different parameter settings than effective local optimization in the presence of random noise. Section 5 reports QNSTOP performance results for a noisy deterministic optimization problem from computational biology. (Examples of other types of problems to which QNSTOP is applicable are given in a supplementary file, along with the algorithm code.)

3. QUASI-NEWTON METHODS FOR STOCHASTIC SEARCH

Quasi-Newton methods attempt to approximate second-order information about the objective function without recourse to second-order derivatives. The rationale for using quasi-Newton methods in stochastic search is the same as for numerical optimization: if second-order information exists and can be reliably extracted from data samples, then it is computationally advantageous to exploit it. If second-order information does not exist or cannot be reliably approximated, then the algorithm degenerates to a first-order method based on first-order (gradient) approximations.

Both RSM and SA mimic the method of steepest descent, but numerical optimization has advanced dramatically since the 1950s and the method of steepest descent is no longer the state of the art. The class of QNSTOP algorithms synthesizes ideas from RSM (semilocal approximations constructed from designed experiments by regression, confidence sets for constrained minimizers, ridge analysis) and SA (convergence analysis), combining them with ideas from modern numerical optimization (trust regions, secant updates).

QNSTOP was originally developed for stochastic search; however, with significant modification to certain steps, QNSTOP can also be used for deterministic global optimization. Both uses supported by the code are described simultaneously

in what follows, repeating, for completeness and clarity here, the details of the deterministic global optimization variant from [Easterling et al. 2014]. In iteration k, QNSTOP methods compute the gradient vector $\hat{g}_k$ and Hessian matrix $\hat{H}_k$ of a quadratic model

$$\widehat{m}_k(X - X_k) = \hat{f}_k + \hat{g}_k^T (X - X_k) + \frac{1}{2} (X - X_k)^T \hat{H}_k (X - X_k), \tag{5}$$

of the objective function f centered at X_k, where $\hat{f}_k$ is generally not $f(X_k)$.

In the unconstrained context, QNSTOP methods progress by

$$X_{k+1} = X_k - \left[\hat{H}_k + \mu_k W_k\right]^{-1} \hat{g}_k, \tag{6}$$

where μ_k is the Lagrange multiplier of a trust region subproblem and W_k is a scaling matrix. In the case where the feasible set Θ is a closed convex subset of $\mathbb{R}^p$, consider an algorithm of the form

$$X_{k+1} = \left(X_k - \left[\hat{H}_k + \mu_k W_k\right]^{-1} \hat{g}_k\right)_\Theta,$$

where $(\cdot)_\Theta$ denotes projection onto Θ.

3.1 Estimating the Gradient

Following a response surface methodology approach, QNSTOP designs regression experiments in a region of interest containing the current iterate. QNSTOP uses an ellipsoidal design region centered at the current iterate $X_k \in \mathbb{R}^p$. Let

$$W_\gamma = \left\{W \in \mathbf{R}^{p \times p} : W = W^T,\ \det(W) = 1,\ \gamma^{-1} I_p \preceq W \preceq \gamma I_p\right\}$$

for some $\gamma \geq 1$ where I_p is the $p \times p$ identity matrix. A typical value for γ is 20. The elements of the set W_γ are valid scaling matrices that control the shape of the ellipsoidal design regions with eccentricity constrained by γ. Let the ellipsoidal design regions

$$E_k(\tau_k) = \left\{X \in \mathbb{R}^p : (X - X_k)^T W_k (X - X_k) \leq \tau_k^2\right\}$$

where $W_k \in W_\gamma$. In the deterministic case $\tau_k = \tau_0 > 0$ is fixed if there is no gain, otherwise for gain $\zeta > 0$ (an input parameter)

$$\tau_k = \frac{\zeta}{\zeta + k} \tau_0.$$

In the stochastic case, the convergence theory [Castle 2012] requires that τ_k be decayed according to the formula $\tau_k = a(k+1)^{-b}$, where $a > 0$ and $b \in (0, 0.5)$.

In each iteration, QNSTOP methods choose a set of N_k design sites $\{X_{k1}, \ldots, X_{kN_k}\} \subset E_k(\tau_k) \cap \Theta$. In this implementation $N = N_k$ is fixed for each $k = 1, 2, \ldots$ and $X_{k1}, \ldots, X_{kN} \in E_k(\tau_k) \cap \Theta$ are uniformly sampled in each iteration.

Let $Y_k = (y_{k1},\ldots,y_{kN})^T$ denote the N-vector of responses where $y_{ki} = F(X_{ki}) +$ noise. The response surface is modeled by the linear model $y_{ki} = \hat{f}_k + X_{ki}^T \hat{g}_k + \epsilon_{ki}$ where ϵ_{ki} accounts for lack of fit. Let $\bar{X}_k = N^{-1} \sum_{i=1}^{N} X_{ki}$. The least squares estimate of the gradient $\hat{g}_k$, ignoring the estimate for $\hat{f}_k$, is obtained by observing the responses and solving

$$\left(D_k^T D_k\right) \hat{g}_k = D_k^T Y_k \tag{7}$$

where

$$D_k = \begin{bmatrix} \left(X_{k1} - \bar{X}_k\right)^T \\ \vdots \\ \left(X_{kN} - \bar{X}_k\right)^T \end{bmatrix}.$$

3.2 Updating the Model Hessian Matrix

In the stochastic context, QNSTOP methods constrain the Hessian matrix update to satisfy

$$-\eta I_p \preceq \hat{H}_k - \hat{H}_{k-1} \preceq \eta I_p \tag{8}$$

for some $\eta \geq 0$. Conceptually, this prevents the quadratic model from changing drastically from one iteration to the next. A variation of the SR1 (symmetric, rank one) update $\hat{H}_k$ that satisfies this constraint is computed as the solution to the problem

$$\begin{aligned} &\min_{H \in \mathbf{R}^{p \times p}} \left\| H\left(X_k - X_{k-1}\right) - \left(\hat{g}_k - \hat{g}_{k-1}\right) \right\|^2 \\ &\text{subject to } H = H^T,\ \text{rank}\left(H - \hat{H}_{k-1}\right) = 1,\ -\eta I_p \preceq H - \hat{H}_{k-1} \preceq \eta I_p. \end{aligned}$$

This problem has an easily computed explicit solution. However, the constraint (8) is simply relaxed in the deterministic case and the BFGS update is used, i.e., with the Hessian matrix updated according to

$$\hat{H}_k = \hat{H}_{k-1} - \frac{\hat{H}_{k-1} s_k s_k^T \hat{H}_{k-1}}{s_k^T \hat{H}_{k-1} s_k} + \frac{\nu_k \nu_k^T}{\nu_k^T s_k},$$

where $s_k = X_k - X_{k-1}$, $\nu_k = \hat{g}_k - \hat{g}_{k-1}$.

3.3 Step Length Control

QNSTOP methods utilize an ellipsoidal trust region concentric with the design region for controlling step length. In the deterministic case, the trust region ellipsoid radius ρ_k is taken equal to the design ellipsoid radius τ_k, and the following optimization problem is solved:

$$\min_{X \in E_k(\rho_k)} \quad \hat{g}_k^T \left(X - X_k\right) + \frac{1}{2} \left(X - X_k\right)^T \hat{H}_k \left(X - X_k\right). \tag{9}$$

The solution to (9) is on the arc

$$X(\mu) = X_k - \left[\hat{H}_k + \mu W_k\right]^{-1} \hat{g}_k. \tag{10}$$

It remains to estimate μ_k such that $X(\mu_k)$ solves (9). Using Lemma 6.4.1 from [Dennis and Schnabel 1983] and a little manipulation, it can be established that there is a unique $\mu_k \geq 0$ such that $\|X(\mu_k) - X_k\|_{W_k} = \rho_k$, unless $\|X(0) - X_k\|_{W_k} \leq \rho_k$ in which case $\mu_k = 0$. Estimating μ_k is difficult, but well understood. Chapter 7 in [Conn, Gould, and Toint 2000] is a comprehensive treatment. In particular, Algorithm 7.3.6 in [Conn, Gould, and Toint 2000] is robust and easily implemented.

In the stochastic case, the trust region ellipsoid radius ρ_k is different from the design ellipsoid radius τ_k, but rather than updating the trust region radius ρ_k and then solving for the Lagrange multiplier μ_k from (10), μ_k is directly updated, thereby defining the trust region radius implicitly rather than explicitly. Specifically, fix $c \geq 0$ and $d > \eta\gamma$ (γ from W_γ in Section 3.1 and η from Equation (8)), set $\mu_k = d(c + k + 1)$, and solve (6) to obtain X_{k+1}, the next iterate. Then $\rho_k = \|X_{k+1} - X_k\|_{W_k}$ is indirectly defined by μ_k. This strategy is dictated by the convergence theory of Castle [2012] that requires control of the Lagrange multipliers.

3.4 Updating the Experimental Design Region

The QNSTOP approach to constructing the ellipsoidal design regions is now described. To motivate the approach, consider the standard example (Section 2.1) with μ quadratic and the problem of minimizing μ subject to an ellipsoidal constraint. If a quadratic model is estimated by least squares regression, then the method of Stablein et al. [1983] can be used to derive a nonlinear inequality that characterizes a confidence set for the constrained minimizer of μ. The confidence set itself is intractable, but a convenient ellipsoidal approximation of it is available.

QNSTOP mimics the construction described above to construct a new ellipsoid from an ellipsoidal trust region subproblem. Because QNSTOP constructs a linear model by least squares regression, then updates the model Hessian matrix by a secant update, the interpretation of the ellipsoid as a confidence set is somewhat more tenuous. Regardless, the approximation for the covariance matrix of $\nabla \widehat{m}_k(X_{k+1} - X_k)$,

$$V_k = 4\sigma^2 (D_k^T D_k)^{-1}, \tag{11}$$

is computed, where σ^2 is the ordinary least squares estimate of the variance. Then

$$E_{k+1}(\chi_{p,1-\alpha}) = \left\{ X \in \mathbb{R}^p : (X - X_{k+1})^T W_{k+1} (X - X_{k+1}) \leq \chi^2_{p,1-\alpha} \right\},$$

where

$$W_{k+1} = \left(\hat{H}_k + \mu_k W_k\right)^T V_k^{-1} \left(\hat{H}_k + \mu_k W_k\right)$$

and $\chi^2_{p,1-\alpha}$ is the $1-\alpha$ quantile of a chi-squared distribution with p degrees of freedom.

Castle [2012] discovered that strict use of the above updates for W_{k+1} can lead to degenerate ellipsoids. To ensure useful design ellipsoids and guarantee convergence [Castle 2012], the constraints $\gamma^{-1}I_p \preceq W_{k+1} \preceq \gamma I_p$ and $\det(W_{k+1}) = 1$ are enforced by modifying the eigenvalues—hence, the definition of $\mathrm{W}_\gamma \ni W_{k+1}$.

3.5 Algorithm Summary

The Fortran code takes as optional arguments all the parameters mentioned above, as well as a few more not mentioned (e.g., one can bound the eccentricity of V_k in (11)). The only required arguments are those defining the problem and a mode—global deterministic or stochastic. Optional arguments not defined default to reasonable values. In both modes, it is generally desirable to run QNSTOP from multiple start points, and the code provides several different ways to acquire these start points. The algorithm described below is repeated for each start point.

Step 0 (initialization): Given a function evaluation budget $\tilde{B}$ per start point and operating mode (deterministic or stochastic), set values for $\tau_0 > 0$, $\mu_0 > 0$, $\gamma \geq 1$, $\eta \geq 0$, $\zeta \geq 0$, N, X_0, $k := 0$, $W_0 := \hat{H}_0 := I_p$.

Step 1 (regression experiment): Depending on the mode, compute τ_k. Uniformly sample $\{X_{k1}, \ldots, X_{kN}\} \subset E_k(\tau_k) \cap \Theta$. Observe the response vector $Y_k = (y_{k1}, \ldots, y_{kN})^T$. Compute $\hat{g}_k$ by solving (7).

Step 2 (secant update): If $k > 0$, compute the model Hessian matrix $\hat{H}_k$ using BFGS (deterministic) or SR1 variant (stochastic) update.

Step 3 (update iterate): Compute μ_k depending on the mode as described in Section 3.3, solve $[\hat{H}_k + \mu_k W_k]s_k = -\hat{g}_k$, and compute $X_{k+1} = \big(X_k + s_k\big)_\Theta$.

Step 4 (update subsequent design ellipsoid): Compute $W_{k+1} \in W_\gamma$ using the approach described in Section 3.4.

Step 5: If $(k+2)(N+1)+1 < \tilde{B}$ then increment k by 1 and go to **Step 1**. Otherwise, the algorithm terminates. (f is also observed at each ellipsoid center X_k.)

As a practical matter to deal with variable scaling, the feasible set (box) $\Theta = B = \{x \in \mathbb{R}^p \mid \ell \leqq x \leqq u\}$ is mapped to the unit hypercube $[0,1]^p$ internally by the code, and the algorithm effectively operates on $[0,1]^p$. All input and output is in the original problem coordinate system.

Other practical issues concern rare exceptional situations. For certain problems and unfortunate choices of the parameters γ and η in stochastic mode, the initial step $X_1 - X_0$ may be unreasonably large, even going far outside the feasible box Θ. Theoretically, and in computational practice, the algorithm does recover from an unreasonable initial step, but wastes considerable effort doing so. Hence, the initial step in stochastic mode is chosen the same as that for global deterministic mode, and subsequent steps by updating μ_k as described in Section 3.3. Given user

choices for γ and η, this initial step determines μ_0, from which appropriate values for the constants c and d are backed out (rather than being input values or fixed in the code).

Another rare possibility is $\sigma^2 \approx 0$ in the calculation of V_k, in which case the update of W_k using V_k^{-1} should be omitted, i.e., take $W_{k+1} = W_k$. Similarly in the global deterministic mode, if $\nu_k^t s_k \approx 0$ or $s_k \approx 0$, skip the BFGS update of $\hat{H}_{k-1}$ and take $\hat{H}_k = \hat{H}_{k-1}$.

Figure 1 shows a typical progression of QNSTOP over 20 iterations, from a difficult (deterministic, with severe numerical noise) biomechanics problem described in [Radcliffe et al. 2010, Easterling et al. 2014]. (Data from a high-dimensional stochastic problem where the randomness is not additive [Chen et al. 2019] is similar.) The solid line represents the lowest value found among 200 design sites for that iteration, while the dotted line represents the corresponding minimum found by the minimizer of the quadratic model. Note that while at times the model will give an imperfect minimum, the overall downward trend is significant.

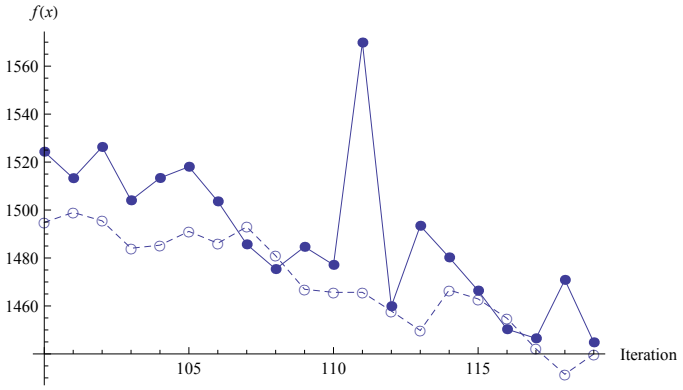

Fig. 1. A typical QNSTOP progression.

3.6 Convergence Theory in the Stochastic Case

Conventional trust-region methods for numerical optimization adjust the size of the trust region according to how successfully the model function predicts decrease in the objective function. Carter [1981] showed that exact information is not needed to ensure global convergence, but it is not clear how to extend Carter's proof techniques to the case of random noise. Instead, Castle [2012, Chapter 4] extended Fabian's [1971] convergence analysis of stochastic approximation to a particular subclass of QNSTOP algorithms. Doing so leads to QNSTOP algorithms that, like SA, rely on predefined sequences for steplength control. Like SA, the necessary assumptions are somewhat restrictive and do not include a number of interesting problems on which QNSTOP performs well in practice. Nevertheless, the theory proved valuable because it directly influenced algorithm design and led to more effective algorithms. In particular, it motivated two novel features of QNSTOP:

the constrained SR1 Hessian update, and the strategy of updating the Lagrange multipliers that determine the trust regions rather than directly updating the trust regions.

Castle's convergence analysis of QNSTOP requires certain conditions on the various QNSTOP parameters (stated earlier in this section in reference to the "stochastic case") and the following assumptions from stochastic approximation. Using the notation in Sections 2 and 3, assume

(1) the decaying τ_k and increasing μ_k have the earlier stated forms for the stochastic case;

(2) the gradient estimate $\hat{g}_k$ used in the quadratic model $\hat{m}_k$ is independent of the gradient estimate $\check{g}_k$ used to construct $\hat{H}_k$ (achieved by having two observed responses at each design site X_{ki} — $\hat{g}_k = \check{g}_k$ has been used in practice with no apparent ill effects);

(3) for each k and design points $\{X_{k1}, \ldots, X_{kN}\} \subset E_k(\tau_k) \cap \Theta$, the scaled design matrix

$$\frac{1}{2\tau_k \gamma^{1/2}} \begin{bmatrix} \big(X_{k1} - \bar{X}_k\big)^T \\ \vdots \\ \big(X_{kN} - \bar{X}_k\big)^T \end{bmatrix}$$

has singular values bounded below by $\Pi > 0$;

(4) $f(x) = T\big(P(\cdot; x)\big)$ with observations $\hat{f}_n(x) = T\big(\hat{P}_n(\cdot; x)\big) = T\big(P(\cdot; x)\big) + \epsilon_x$;

(5) the objective function f is twice continuously differentiable, bounded from below, and $\|\nabla^2 f(x)\| \leq L < \infty$ for some L and all $x \in \mathbb{R}^p$;

(6) the observed errors have zero mean and finite variance, i.e., $E[\epsilon_x] = 0$ and $E[\epsilon_x^2] \leq c_\epsilon$;

(7) the objective function has a unique minimizer θ^*,

$$\inf_{\|x-\theta^*\|>\phi} \|\nabla f(x)\| > 0$$

($\|\nabla f(x)\| \to 0$ only at θ^*), and

$$\inf_{\|x-\theta^*\|>\phi} \big(f(x) - f(\theta^*)\big) > 0$$

($f(x) \to f(\theta^*)$ only at θ^*) for all $\phi > 0$.

Then the iterates X_k generated by QNSTOP converge almost surely to the unique minimizer θ^* of f.

The multivariate Kiefer-Wolfowitz algorithm for stochastic approximation is

$$\theta_{k+1} = \theta_k - \frac{b_k}{2c_k} \begin{pmatrix} \hat{f}_1(\theta_k + c_k e_1) - \hat{f}_1(\theta_k - c_k e_1) \\ \vdots \\ \hat{f}_1(\theta_k + c_k e_p) - \hat{f}_1(\theta_k - c_k e_p) \end{pmatrix},$$

where $e_1, \ldots, e_p$ are unit vectors in the coordinate directions, $c_k > 0$ controls the width of the finite differencing interval, and $b_k > 0$ controls step length. Choosing $\mu_k = 1/b_k$, $\eta = 0$ (entailing $\hat{H}_k = \hat{H}_0$), $\gamma = 1$ (entailing $W_k = I_p$, which results in spherical experimental regions), and $N = 2p$ design sites at $\theta_k \pm c_k e_i$ yields Kiefer-Wolfowitz as a special case of QNSTOP. Allowing $\gamma > 1$ and placing the $2p$ design sites at the endpoints of the resulting ellipsoid's axes permits simulation experiments that investigate the value of replacing spherical design regions with elliptical regions that adapt to the contours of the objective function. Allowing $\eta > 0$ permits simulation experiments that investigate the value of using (some) second-order information. Castle's [2012, Chapter 5] simulation experiments suggest that both innovations have virtue.

Castle's [2012, Chapter 4] convergence analysis assumes that all observed function values are independent, foreclosing the possibility of storing function values and reusing them in subsequent iterations. In practice, this technique can be highly effective. Furthermore, Assumption (2) requires that gradients and Hessians be estimated independently. In practice, using the same observed function values to estimate both gradients and Hessians appears to be equally effective. Finally, note that Castle's convergence analysis only concerns the case of optimization in the presence of random noise. For global optimization of noisy deterministic functions, QNSTOP should be regarded as a (highly effective) heuristic search strategy.

3.6.1 Outline of QNSTOP Convergence Proof

Castle's [2012] convergence analysis of QNSTOP mimics Fabian's [1971] convergence analysis of generalized Kiefer-Wolfowitz algorithms for stochastic approximation. Both analyses use technical arguments to establish that a subsequence of $\{\|\nabla f(X_k)\|\}$ converges to 0 almost surely, from which if follows immediately that X_k converges to θ^* almost surely.

Castle [2012] studied algorithms of the form $X_{k+1} = X_k - B_k g_k$, where g_k estimates $\nabla f(X_k)$ and B_k is a symmetric and positive definite matrix-valued measurable function with eigenvalues in $[m_k, M_k]$. Write

$$X_{k+1} = X_k - B_k \nabla f\left(X_k\right) + B_k \beta_k + B_k \delta_k,$$

where $\beta_k = \nabla f(X_k) - E[g_k | X_0, \ldots, X_k]$ denotes "systematic error" and $\delta_k = E[g_k | X_0, \ldots, X_k] - g_k$ denotes "stochastic error". Suppose that these errors are bounded in such a way that $\|\beta_k\| \leq c_\beta \tau_k$ and $E[\|\delta_k\|^2 | X_0, \ldots, X_k] \leq c_\delta / \tau_k^2$, with $\tau_k \to 0$. Suppose, moreover, that the bounds on the eigenvalues of B_k satisfy

$$\left(\tau_k + M_k\right) M_k / m_k \to 0, \quad \sum_{k=0}^{\infty} m_k = \infty, \quad \sum_{k=0}^{\infty} M_k \tau_k < \infty, \text{ and } \sum_{k=0}^{\infty} M_k^2 / \tau_k^2 < \infty.$$

Castle [2012] then deduced the inequalities

$$E\left[u_k \mid X_0, \ldots, X_k\right]^T \nabla f\left(X_k\right) \geq a_k D_k^2 - b_k - c_k D_k$$

and

$$E\left[\|u_k\|^2 \mid X_0, \ldots, X_k\right] \leq b_k + c_k D_k + d_k D_k^2,$$

where $u_k = B_k g_k$, $D_k = \|\nabla f(X_k)\|$, $a_k = m_k$, $b_k = 2c_\delta M_k^2/\tau_k^2 + 4c_\beta^2 \tau_k^2 M_k^2$, $c_k = c_\beta \tau_k M_k$, and $d_k = 4M_k^2$. It then follows from Fabian's [2012] Lemma 3.3 that $D_k \to 0$ almost surely.

In contrast to stochastic approximation, QNSTOP obtains the gradient estimates g_k by regression. Following the proof technique for Theorem 2.13 in [Conn et al., 2009, Errata], Castle [2012] assumed that the designs of these regression experiments are Π-poised, i.e., after centering the design and scaling it so that each design site lies in the unit ball, the smallest singular value of the design matrix is at least $\Pi > 0$. It then follows that both the systematic and stochastic errors described above are suitably bounded.

Finally, Castle's [2012] implementation of QNSTOP sets $B_k = (H_k + \mu_k W_k)^{-1}$ in such a way that the eigenvalues of B_k are bounded below by m_k and above by M_k. Critical to this construction was the specification that $\mu_k = d(c + k + 1)$, meaning that the Lagrange multipliers of the trust region subproblems are predetermined by a gain sequence. Thus, the convergence analysis of QNSTOP led directly to a novel implementation of the trust-region framework.

A subtle assumption of Castle's [2012] convergence analysis is that $E[B_k g_k \mid X_0, \ldots, X_k] = B_k E[g_k \mid X_0, \ldots, X_k]$. Because B_k depends on H_k, which is constructed from g_k via a secant update, this equation can only be ensured by performing two independent regression experiments, one for the purpose of forming g_k, the other for the purpose of forming H_k. In practice, however, QNSTOP seems to perform just as well when a single experiment is used to form both g_k and H_k. ■

4. PARALLEL IMPLEMENTATION

QNSTOP, unlike, say, the massively parallel direct search code VTDIRECT95 [Jones et al. 1993, Jones 2001, Deng and Ferris 2007, He et al. 2009], requires no exotic data structures or sophisticated communication management. There are just three potentially significant sources of parallelism: the individual function evaluations $f(X_{ki})$, the loop ($i = 1, \ldots, N$) over the samples in an experimental design, and the loop over the start points (of size `NSTART`). These three levels are nested. If each evaluation $f(X_{ki})$ were a large-scale parallel simulation using MPI, then a master-slave paradigm with the master farming out points X_{ki} to the slaves for evaluation is a reasonable approach entirely based on MPI. If the distributed memory nodes are multicore, then a mixed programming model makes sense, but the shared memory (OpenMP) component would be within the function evaluations, not at the level of the two outer loops. On a large shared memory machine, there will be ample parallelism at the two outer loops, motivating an OpenMP approach.

Due to the exception handling limitations of OpenMP threads, the logical flow of the parallel driver subroutine `QNSTOPP` is significantly different from that of the serial (without OpenMP directives) driver subroutine `QNSTOPS`. Consequently, the

serial version QNSTOPS execution is more efficient than the parallel version QNSTOPP execution with a single thread. This is the justification for providing both serial and parallel subroutines, even though in principle the OpenMP code QNSTOPP can be run serially.

The parallel (OpenMP) implementation of QNSTOP has four choices for parallelization, controlled by an optional argument to the Fortran subroutine QNSTOPP: (1) serial (no parallelization at all, the default), (2) parallelize only the outer loop over the start points, (3) parallelize only the second outermost loop over the experimental design samples, or (4) do both (2) and (3). The choice (4), because of nesting, could generate a very large number of threads, so should be used with care. Figures 2–4 show speedup results for a eukaryotic cell cycle model problem [Oguz et al. 2013] from the systems biology literature. The model is a system of 26 stiff ODEs with 149 parameters. There is experimental data on 119 mutants, each of which corresponds to a modification of the base (or "wild type") system of ODEs, and the optimization problem is to estimate the 149 parameters so as to best fit the experimental data for all the mutants. Each mutant is classified as "viable", "inviable", or "neither", and the objective function value at a particular 149-dimensional parameter vector is simply the (negative) count of how many mutants' behavior is matched by the model. One objective function evaluation $f(X)$ on a single PowerPC G4 processor typically takes about 15 s, but can take an order of magnitude more depending on the parameter vector, due to the different ODE trajectories (being tracked with LSODAR).

The optional argument OMP, referenced in Figs. 2–4, defining the parallel decomposition has the values 1, 2, 3 corresponding to dynamically scheduled loop parallelization over the start points, design ellipsoid sample points, or both, respectively. For these experiments, the number of start points is NSTART = 64 and the number of design ellipsoid sample points (at which the objective function is observed) is N = 256. Each data point represents the mean of three runs (for which the variance is so small that the point is shown without error bars) or five runs (point shown with error bars). It is not surprising that OpenMP nesting (OMP = 3) performs significantly better than no nesting, since there are fewer threads (square root of the total number of threads) at each level of parallelism. The speedup plots (Figs. 2–4) are consistent with Amdahl's Law, and show the limitations of coarse grained parallelization (even with dynamic loop scheduling) when there is limited problem parallelism and the function evaluation times are highly variable (typical of optimization problems with black box simulation function values).

5. PERFORMANCE

Easterling et al. [2014] reported performance data only for the deterministic global optimization variant of QNSTOP, including the best-known result for a biomechanics problem. New results for both variants of QNSTOP applied to a systems biology problem are reported here, including the best-known result for this problem.

The systems biology literature on cell cycle models contains a parameter vector X^0 (called the TL set) obtained by biochemistry knowledge and manual twiddling,

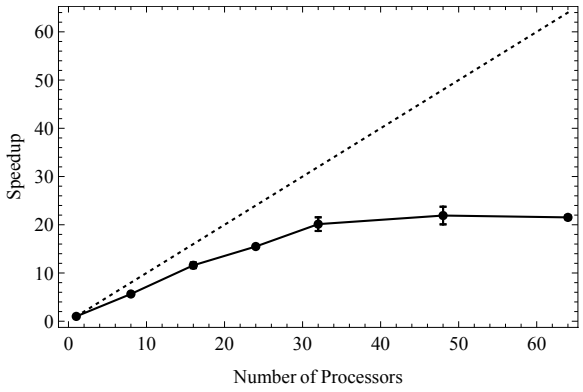

Fig. 2. Speedup of the parallel QNSTOPP over the serial QNSTOPS for the cell cycle problem with OMP = 1 (parallel loop over start points). The mean speedup is plotted with error bars at one standard deviation.

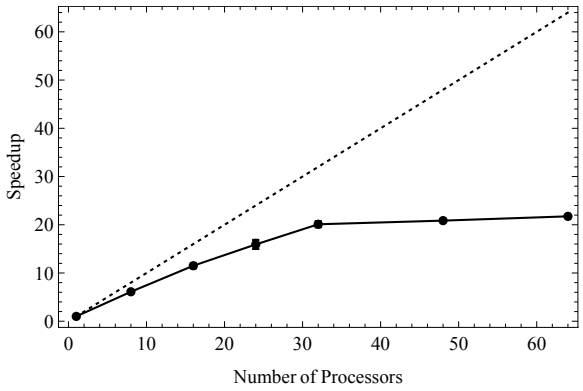

Fig. 3. Speedup of the parallel QNSTOPP over the serial QNSTOPS for the cell cycle problem with OMP = 2 (parallel loop over design ellipsoid sample points). The mean speedup is plotted with error bars at one standard deviation.

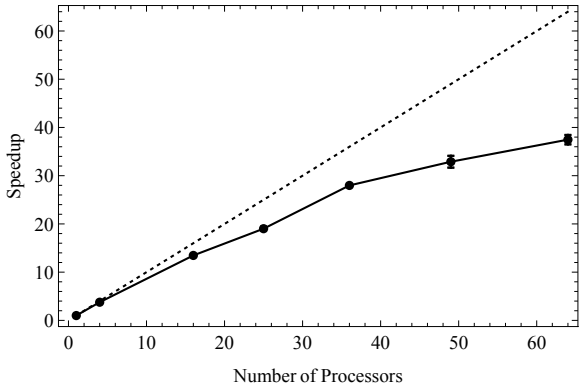

Fig. 4. Speedup of the parallel QNSTOPP over the serial QNSTOPS for the cell cycle problem with OMP = 3 (both OMP = 1 and OMP = 2, nesting). The mean speedup is plotted with error bars at one standard deviation.

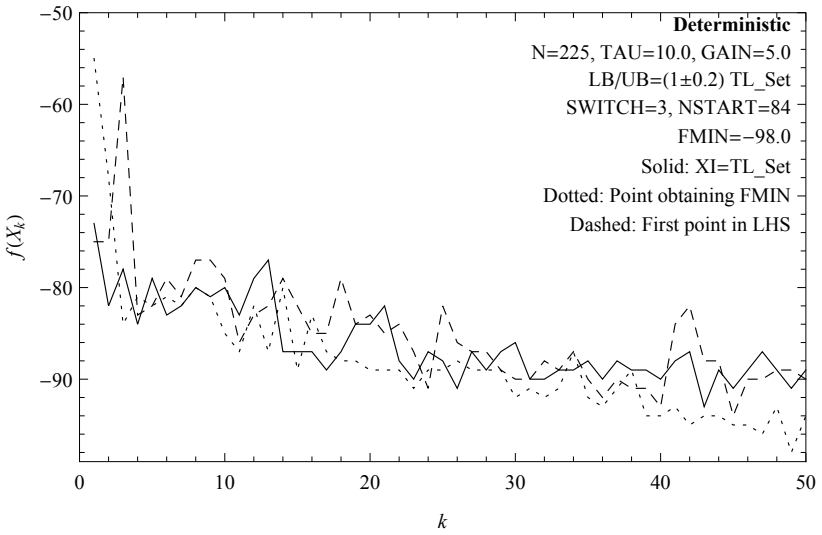

Fig. 5. Execution traces of QNSTOP in deterministic mode for three start points in the ±20% box. One trace starts at the center of the box (where $f(X) = -73$) and another trace contains the best point of the entire run (where $f(X) = -98$). Another run with TAU = 2.2 (scaled from TAU = 10.0 for the ±90% box) yielded a best value of −97.

considered in the ballpark of the correct values. Searches for the optimal parameter vector generally are conducted in a box defined by X^0 plus or minus some percent of X^0, say 20%, 40%, 90% defining the boxes $[0.8\,X^0, 1.2\,X^0]$, $[0.6\,X^0, 1.4\,X^0]$, $[0.1\,X^0, 1.9\,X^0]$, respectively. For the particular model known as "Oak's deterministic model" [Oguz et al. 2013], the best-known value of $f(X)$ is −110 (obtained using LSODAR, or −111 obtained using a less accurate fixed step Euler method as done by Oguz et al. [2013]), where $f(X^0) = -73$. Using `NSTART` = 84 and `N` = 225 (from the statistical rule of thumb that at least $1.5p$ data points are needed to estimate p parameters, and the model gradient $\hat{g}_k$ here has dimension $p = 149$), Figs. 5–7 show the iteration histories for three start points (out of 84) for each of the three ±20%, ±40%, ±90% boxes, running QNSTOP in deterministic global optimization mode with the other relevant algorithm parameters shown in the figure legends. These legends list the subroutine `QNSTOP[P|S]` arguments: `N` is the number of design ellipsoid sample points; `TAU` is the initial ellipsoid radius τ_0; `GAIN` is the gain ζ (cf. §3.1); [`LB`, `UB`] is the feasible box; `SWITCH` controls how start points are provided, with values 1, 2, 3, 4 corresponding to a single start point `XI`, a given list of start points, an automatically generated Latin hypercube design (containing `XI`) of start points, adaptive generation of a sequence of start points (beginning with `XI`) by a user provided procedure, respectively; `NSTART` is the number of start points (for `SWITCH` = 3 or 4); `XI` is the initial specified start point.

The trajectories for *all* start points are similar to the general downward trend of the three start point trajectories shown. The best values found for $f(X)$ during the three runs for the ±20%, ±40%, ±90% boxes were −98, −105, −112, respectively, improving on the best-known value in the literature. For the runs depicted in

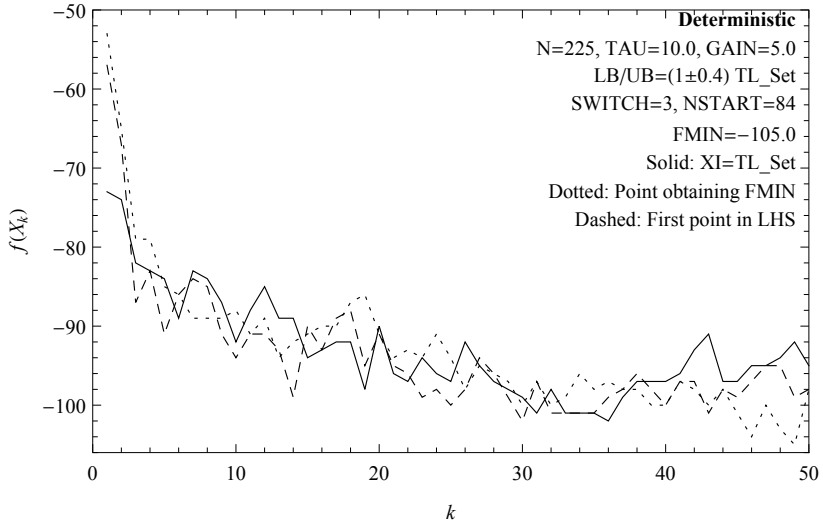

Fig. 6. Execution traces of QNSTOP in deterministic mode for three start points in the ±40% box. One trace starts at the center of the box (where $f(X) = -73$) and another trace contains the best point of the entire run (where $f(X) = -105$). Another run with TAU = 4.4 (scaled from TAU = 10.0 for the ±90% box) yielded a best value of −104.

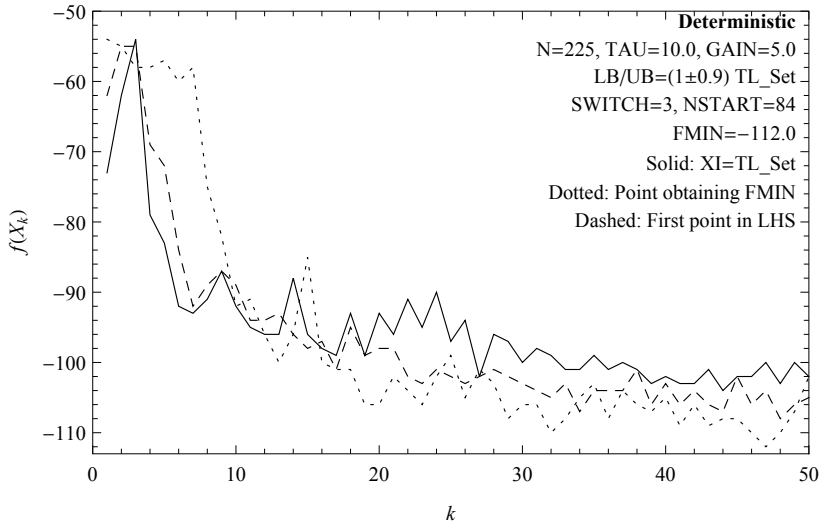

Fig. 7. Execution traces of QNSTOP in deterministic mode for three start points in the ±90% box. One trace starts at the center of the box (where $f(X) = -73$) and another trace contains the best point of the entire run (where $f(X) = -112$).

Figs. 5–7, Table I gives the statistics for the best $f(X)$ value found with each of the 84 starting points. The global deterministic (stochastic) mode is denoted by 'G' ('S').

Figure 8 shows a trace plot for the stochastic mode (S) for the ±90% box similar to Fig. 7 for the global deterministic mode (G), and the statistics for that stochastic

Table I. Statistics for Best $f(X)$ Value Found with Each of the 84 Starting Points, for Each of the ±20%, ±40%, ±90% Boxes.

box	min	median	max	$\bar{\sigma}$	mode
±20%	−98	−92	−88	1.97	G
±40%	−105	−100	−95	2.19	G
±90%	−112	−105	−55	7.42	G
±90%	−109	−101	−55	18.88	S

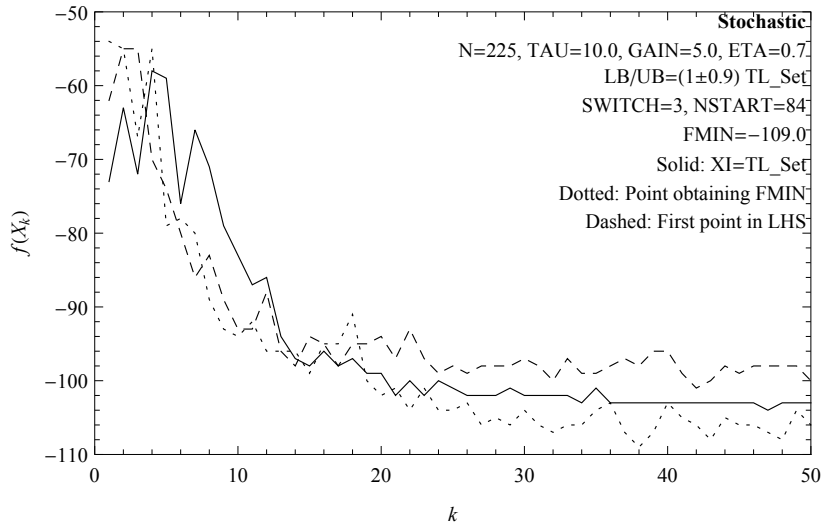

Fig. 8. Execution traces of QNSTOP in stochastic mode for three start points in the ±90% box. One trace starts at the center of the box (where $f(X) = -73$) and another trace contains the best point of the entire run (where $f(X) = -109$).

mode run are included in Table I. Execution traces and statistics for the stochastic mode for the ±20% and ±40% boxes are what would be expected for these smaller boxes, and thus are omitted. Since the stochastic mode has to protect against unknown random fluctuations, the convergence is much slower than for the global deterministic mode (for this deterministic cell cycle problem). Castle [2012] reports results for QNSTOP in stochastic mode applied to a truly stochastic tumor growth model.

Comparison of QNSTOP to other algorithms, both deterministic and nondeterministic, is not done here since that has already been done in the literature [Easterling et al. 2014] for some very hard "noisy" scientific optimization problems. Parametric studies (e.g., [Amos et al. 2014]) are not included here, because they are generally not useful. All the algorithm parameters (`N`, `TAU`, `GAIN`, `ETA`, etc.) are optional arguments, which if omitted default to reasonable values. Changing these values by, say, 50%, makes little difference in performance, but changing them by an order of magnitude can (depending on the problem) make a huge difference.

6. CONCLUSION

The current version of QNSTOP reflects computational experience since 2010 on a wide variety of optimization problems, including both local optimization in the presence of random noise and global optimization of noisy deterministic functions. The stochastic mode ('S') of QNSTOP has been quite successful [Castle 2012; Amos et al. 2014; Chen et al. 2019] on true stochastic search problems for which the probability distributions of observations of $f(X)$ are either unknown or analytically intractable. The deterministic mode ('G') of QNSTOP is definitely competitive for global optimization, as reported in [Radcliffe et al. 2010] and [Easterling et al. 2014].

There are multiple levels of parallelism in QNSTOP, which can be easily exploited, but because in typical engineering and science applications the function evaluation time has a large variance, load balancing becomes difficult. The parallel results here reflect this situation. If the function evaluations $f(X)$ themselves were parallelized, then the load balancing and overall parallel efficiency could be excellent.

BIBLIOGRAPHY

Amos, B. D., Easterling, D. R., Watson, L. T., Castle, B. S., Trosset, M. W., and Thacker, W. I. 2014. Fortran 95 implementation of QNSTOP for global and stochastic optimization. In *Proc. 2014 Spring Simulation Multiconference, 22nd High Performance Computing Symposium*, K. Rupp, L. T. Watson, W. Thacker, and M. Sosonkina, eds., Society for Modelling and Simulation Internat., Vista, CA, 111–118.

Atkinson, E. N., Bartoszyński, B., Brown, B. W., and Thompson, J. R. 1983. Simulation techniques for parameter estimation in tumor related stochastic processes. In *Proc. 1983 Computer Simulation Conference*, North-Holland, New York, 754–757.

Birge, J. R. and Louveaux, F. 2011. *Introduction to Stochastic Programming, Second Edition*. Springer, New York.

Blum, J. R. 1954. Multidimensional stochastic approximation methods. *Annals of Mathematical Statistics 25*, 737–744.

Box, G. E. P. and Hunter, J. S. 1957. Multi-factor experimental designs for exploring response surfaces. *Annals of Mathematical Sciences 28*, 195–241.

Box, G. E. P. and Wilson, K. B. 1951. On the experimental attainment of optimum conditions. *J. Royal Statistical Society, Series B, 13*, 1–45.

Carter, R. G. 1981. On the global convergence of trust region algorithms using inexact gradient information. *SIAM J. Numer. Anal. 28*, 1, 251–265.

Castle, B. S. 2012. Quasi-Newton methods for stochastic optimization and proximity-based methods for disparate information fusion. Ph.D. thesis, Indiana University, Bloomington, IN. http://mypage.iu.edu/~mtrosset/Students/Brent/brent-dissertation.pdf.

Chang, K. H., Hong, L. J. and Wan, H. 2007. Stochastic trust region gradient-free method (STRONG) —a new response-surface-based algorithm in simulation optimization. In *Proc. 2007 Winter Simulation Conference*, S. G. Henderson, B. Biller, M. H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds., 346–354.

Chang, K. H. and Wan, H. 2009. Stochastic trust region response surface convergent method for generally-distributed response surface. In *Proc. 2009 Winter Simulation Conference*, M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds., 563–573.

Chen, M., Amos, B. D., Watson, L. T., Tyson, J. J., Cao, Y., Shaffer, C. A., Trosset, M. W., Oguz, C., and Kakoti, G. 2019. Quasi-Newton stochastic optimization algorithm for parameter estimation of a stochastic model of the budding yeast cell cycle. *IEEE/ACM Trans. Comput. Biol. Bioinformatics 16*, 301–311.

Conn, A. R., Gould, N. I. M., and Toint, P. L. 2000. *Trust-Region Methods*. MPS-SIAM Series on Optimization, SIAM, Philadelphia.

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

Conn, A. R., Scheinberg, K., and Vicente, L. N. 2009. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization, SIAM, Philadelphia.

Deng, G. and Ferris, M. C. 2006. Adaptation of the UOBYQA algorithm for noisy functions. In *Proc. 2006 Winter Simulation Conference*, L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds., 312–319.

Deng, G. and Ferris, M. C. 2007. Extension of the DIRECT optimization algorithm for noisy functions. In *Proc. 2007 Winter Simulation Conference*, S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds., 497–504.

Dennis, J. E. and Schnabel, R. B. 1983. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, New Jersey.

Diggle, P. J. and Gratton, R. J. 1984. Monte Carlo methods of inference for implicit statistical models. *J. Royal Statistical Society, Series B, 46*, 193–227.

Draper, N. R. 1963. "Ridge analysis" of response surfaces. *Technometrics 5*, 4, 469–479.

Dvoretsky, A. 1956. On stochastic approximation. In *Third Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 39–55.

Easterling, D. R., Watson, L. T., Madigan, M. L., Castle, B. S., and Trosset, M. W. 2014. Parallel deterministic and stochastic global minimization of functions with very many minima. *Comput. Optim. Appl. 57*, 2, 469–492.

Fernholz, L. T. 1983. *von Mises Calculus for Statistical Functionals*. Springer-Verlag, New York.

Gilmore, P. and Kelley, C. T. 1995. An implicit filtering algorithm for optimization of functions with many local minima. *SIAM J. Optim. 5*, 2, 269–285.

He, J., Watson, L. T., and Sosonkina, M. 2009. Algorithm 897: VTDIRECT95: serial and parallel codes for the global optimization algorithm DIRECT. *ACM Trans. Math. Software 36*, Article 17, 1–24.

Jones, D. R. 2001. The DIRECT global optimization algorithm. In *Encyclopedia of Optimization*, Vol. 1, Kluwer Academic Publishers, Dordrecht, 431–440.

Jones, D. R., Pertunen, C. D., and Stuckman, B. E. 1993. Lipschitzian optimization without the Lipschitz constant. *J. Optimization Theory and Applications 79*, 1, 157–181.

Kiefer, J. and Wolfowitz, J. 1952. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics 23*, 462–466.

Kugele, S. C., Trosset, M. W., and Watson, L. T. 2008. Numerical integration in statistical decision-theoretic methods for robust design optimization. *Structural Multidisciplinary Optim. 36*, 457–475.

Kushner, H. J. and Yin, G. G. 1997. *Stochastic Approximation Algorithms and Application*. Springer, New York.

Lawera, M. and Thompson, J. R. 1993. A parallelized, simulation based algorithm for parameter estimation. In *Proc. Thirty-Eighth Conference on the Design of Experiments in Army Research Development and Testing*, B. Bodt, ed., 321–341.

Marquardt, D. 1963. An algorithm for least-squares estimation of nonlinear parameters. *J. SIAM 11*, 2, 431–441.

Marti, K. 2005. *Stochastic Optimization Methods*. Springer, Berlin.

Myers, R. H. and Montgomery, D. C. 1995. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley & Sons, New York.

Oguz, C., Laomettachit, T., Chen, K. C., Watson, L. T., Baumann, W. T., and Tyson, J. J. 2013. Optimization and model reduction in the high dimensional parameter space of a budding yeast cell cycle model. *BMC Systems Biol. 7:53*, 1–17.

Polyak, B. T. and Juditsky, A. B. 1992. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optimization 30*, 838–855.

Powell, M. J. D. 2002. UOBYQA: Unconstrained optimization by quadratic approximation. *Math. Prog. 92*, 555–582.

Powell, W. B. 2019. A unified framework for stochastic optimization. *European J. Operational Res. 275*, 795–821.

Radcliffe, N. R., Easterling, D. R., Watson, L. T., Madigan, M. L., and Bieryla, K. A. 2010. Results of two global optimization algorithms applied to a problem in biomechanics. In *Proc. 2010 Spring Simulation Multiconference, High Performance Computing Symposium*, A. Sandu, L. Watson, and W. Thacker, eds., Soc. for Modelling and Simulation Internat., Vista, CA, 117–123.

Robbins, H. and Monro, S. 1951. A stochastic approximation method. *Annals of Mathematical Statistics 22*, 400–407.

Spall, J. C. 1987. A stochastic approximation technique for generating maximum likelihood parameter estimates. In *Proc. American Control Conference*, Minneapolis, MN, June 10-12, 1161–1167.

Spall, J. C. 1992. Multivariate stochastic approximation using simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control 37*, 3, 332–341.

Spall, J. C. 1998. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Trans. Aerospace Electronic Systems 34*, 3, 817–823.

Spall, J. C. 2003. *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, New York.

Stablein, D. M., Carter, Jr., W. H., and Wampler, G. L. 1983. Confidence regions for constrained optima in response-surface experiments. *Biometrics 39*, 759–763.

Thompson, J. R. 2000. *Simulation: A Modeler's Approach*. John Wiley & Sons, New York, NY.

Trosset, M. W. 2003. Trust regions and ridge analysis. In *ASA Proceedings of the Joint Statistical Meetings*, CD-ROM, 4287–4291,
http://mypage.iu.edu/~mtrosset/Research/trust-jsm03.pdf.

von Mises, R. 1947. On the asymptotic distribution of differentiable statistical functions. *Annals of Mathematical Statistics 18*, 309–348.