Parallel Deterministic and Stochastic Global Minimization of Functions with Very Many Minima

DAVID R. EASTERLING dreast@vt.edu

Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

LAYNE T. WATSON

Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

MICHAEL L. MADIGAN

Department of Engineering Science and Mechanics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

BRENT S. CASTLE

School of Informatics and Computing, Indiana University, Bloomington, IN 47405, USA

MICHAEL W. TROSSET

Department of Statistics, Indiana University, Bloomington, IN 47405, USA

Abstract. The optimization of three problems with high dimensionality and many local minima are investigated under five different optimization algorithms: DIRECT, simulated annealing, Spall's SPSA algorithm, the KNITRO package, and QNSTOP, a new algorithm developed at Indiana University.

Keywords: stochastic optimization, deterministic optimization, biomechanics, quadratic optimization, simulated annealing, DIRECT, QNSTOP, KNITRO

1 Introduction

The minimization of difficult functions is an important topic in numerical analysis. An optimization method that solves one problem efficiently and effectively can fail to yield useful results for another problem, even if the two problems under consideration share broad similarities. Three problems are considered here: a biomechanical balance problem with an unknown global minimum, a nonconvex nonsmooth quadratic minimization problem with a global minimum known by construction, and a smooth problem in reflection reduction with a global minimum that can be directly calculated. While the problems are of similar dimensionality and all have a large number of local minima, the character of each problem is distinct.

The balance problem, while deterministic, contains enough modeling noise to cause deterministic optimization algorithms difficulty. (The biomechanics 'model' consists of splicing together published empirical models over different motion regimes. These models are inconsistent at their interfaces, and the resulting combined model is thus discontinuous across manifolds in the domain. The numerical noise (ODEs, integrals) is significant but dominated by the modelling noise.) The balance problem is a constrained optimization problem of 57 dimensions. The nonconvex nonsmooth quadratic minimization problem is a reformulation of an integer programming problem, and the function considered here is carefully constructed to contain a large number of local minima

and a single global optimum point. This minimization problem is an unconstrained problem with a scalable number of dimensions, chosen here to be 57. The wave annihilation problem is a smooth minimization problem with an even number of variables, chosen as 56 to keep the size comparable to that of the other two problems. Together, these three problems provide a useful context in which to compare the performance of the optimization algorithms on moderately large and qualitatively different problems.

Five optimization algorithms are considered for each problem: the simultaneous perturbation stochastic approximation algorithm, two parallel implementations of a simulated annealing scheme, a parallel implementation of the DIRECT algorithm, the direct interior point method found in the commercial KNITRO optimization package, and a new quasi-Newton stochastic algorithm, QNSTOP.

The simultaneous perturbation stochastic approximation algorithm (SPSA) is an unconstrained stochastic optimization algorithm notable for the small number of objective function evaluations per iteration [33]. This allows it to scale to higher dimensions better than the finite difference methods from which it is derived [35]. Like the finite difference methods, it suffers from a tendency to become trapped at local minima. SPSA is more usually employed as a local optimization algorithm, but it may function as a global optimization algorithm under certain broad conditions [26]. The parallel implementation employed here is a naive one, with minimal interprocessor communication. This parallel SPSA is applied to these three problems to test its suitability for problems with a high dimensionality and a large number of local minima. For more information on the SPSA algorithm, see [34] and [36].

Simulated annealing is an unconstrained stochastic global optimization algorithm commonly used for difficult biomechanics problems. The parallel implementation employed here, simulated parallel annealing within a neighborhood (SPAN), is designed to minimize interprocessor communication while maximizing the use of multiple processors to compute objective function values at the desired points [19]. For more on simulated annealing, see [13], [21], and [25]. [29] discusses other parallel simulated annealing algorithms.

The DIRECT algorithm is a highly parallelizable box-constrained deterministic global optimization algorithm [22]. The parallel implementation employed here, pVTdirect, is designed to preserve the deterministic nature of the algorithm while exploiting its natural parallelism [18], [15]. For more information on DIRECT, see [23], [16], and [17].

KNITRO contains a collection of algorithms for local nonlinear optimization developed by Ziena Optimization, LLC. While all the optimization algorithms in the KNITRO package are designed for twice continuously differentiable problems, KNITRO nevertheless contains code for approximation of derivatives and can be used on nonsmooth problems as well, though for such problems the performance may degrade. As the only truly gradient-driven optimization technique considered here, KNITRO provides a contrast to the other algorithms employed here to show the usefulness of such a technique with the test problems here. For more information on KNITRO, see [3], [4], and [5].

QNSTOP, an algorithm under development at Indiana University, is a local search strategy for stochastic optimization that synthesizes ideas from numerical optimization (secant updates, trust regions) and response surface methodology (ridge analysis). Here, stochastic optimization describes problems in which function evaluation is uncertain, i.e., instead of computing $y = \mu(x)$, y is drawn from a probability distribution $P(x)$. For example, the case of additive normal errors with equal variance σ^2 is $y \sim \text{Normal}(\mu(x), \sigma^2)$. Some modifications are therefore necessary to apply this algorithm to the deterministic problems being considered.

The paper is organized as follows. Sections 2, 3, and 4 describe the three test problems. The parallel DIRECT is described in Section 5, simulated annealing in Section 6, Spall's SPSA algorithm in Section 7, KNITRO in Section 8, and QNSTOP in Section 9. Section 10 contains a discussion of experimental results and concludes.

2 Biomechanics

The first problem under consideration is a two-dimensional musculoskeletal model utilizing forward dynamic simulations [2]. The task investigated involves maintaining bipedal balance without stepping after an abrupt backwards support surface displacement.

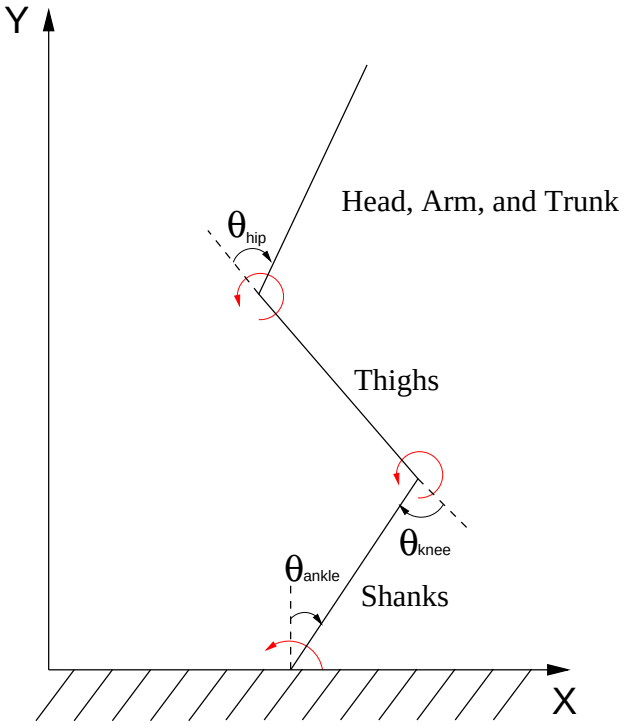

Figure 1. Schematic drawing of the three segment sagittal plane model representing the human body.

The musculoskeletal model is a sagittal plane representation of the volunteer including three rigid segments representing the shanks, thighs, and head-arms-trunk (HAT) connected by frictionless pin joints (see Figure 1) activated by three joint torques representing the torques of the ankles, knees, and hips. The joint torques are the sum of the passive and active joint torques $T = T_p + T_a$ and represent all flexor and extensor contributions to the joints. The feet are neglected in the model because the volunteer from which experimental data was derived exhibited minimal heel rise during trials. As such, the joint representing the ankle is assumed to simply connect the distal end of the shanks to the floor. The inputs to the dynamic model are the joint torques and the time-varying position of the moving platform.

Equations of motion for the model are derived from Lagrange dynamics, and are uniquely determined from the segments' length, mass, center of mass, and moments of inertia. These constants are calculated from the subject's height (1.6 m) and weight (60 kg) by the formulas

presented in [27], and are given here for convenience. The segment masses are calculated as 5.39 kg for the shank segment, 13.0 kg for the thigh segment, and 37.3 kg for the trunk segment. The length of the segments are 0.408 m for the shank, 0.402 m for the thigh, and 0.475 m for the trunk. The center of mass for each segment, in distance from the proximal joint, is 0.171 m for the shank, 0.157 m for the thigh, and $2.5 \cdot 10^{-4}$ m for the trunk. The moments of inertia are calculated as 0.0584 kg $\cdot$ m^2 for the shank, 0.228 kg $\cdot$ m^2 for the thigh, and 2.07 kg $\cdot$ m^2 for the trunk. Finally, the initial angles for the model are -0.0114 rad for the ankle, 3.152 rad for the knee, and 3.272 rad for the hip.

Passive torque (N $\cdot$ m)

$$T_p = 2(T_{p,a},\ T_{p,k},\ -T_{p,h})^{\mathrm{T}}$$

is calculated using equations taken from [30], which generate passive torque with respect to the ankle $T_{p,a}$, knee $T_{p,k}$, and hip $T_{p,h}$. Since these equations are in degrees, joint angles must be converted to degrees in order to use them. Given that $\theta_{a,\circ}$ represents the angle between the ground and the shank, $\theta_{k,\circ}$ represents the angle between the shank and the thigh, and $\theta_{h,\circ}$ represents the angle between the hip and the torso, $T_{p,a} = e^{(a-b\,\theta_{a,\circ}-c\,\theta_{k,\circ})} - e^{(-d+f\,\theta_{a,\circ}+g\,\theta_{k,\circ})} - 1.792$ represents the passive torque associated with a single ankle joint, where $a = 2.1016$, $b = 0.0843$, $c = 0.0176$, $d = 7.97634$, e is Euler's constant, $f = 0.1949$, and $g = 0.0008$. $T_{p,k} = e^{(h-j\,\theta_{a,\circ}-k\,\theta_{k,\circ}+l\,\theta_{h,\circ})} - e^{(-m+n\,\theta_{a,\circ}+o\,\theta_{k,\circ}-p\,\theta_{h,\circ})} + e^{(q-r\,\theta_{k,\circ})} - 4.820$ represents the passive torque associated with a single knee joint, where $h = 1.8$, $j = 0.0460$, $k = 0.0352$, $l = 0.0217$, $m = 3.971$, $n = 0.0004$, $o = 0.0495$, $p = 0.0128$, $q = 2.220$, and $r = 0.150$. $T_{p,h} = e^{(s-t\,\theta_{k,\circ}-u\,\theta_{h,\circ})} - e^{(v-w\,\theta_{k,\circ}+y\,\theta_{h,\circ})} - 8.072$ represents the passive torque associated with a single hip joint, where $s = 1.4655$, $t = 0.0034$, $u = 0.0750$, $v = 1.3403$, $w = 0.0226$, and $y = 0.0305$.

Active torque (N $\cdot$ m)

$$T_a = 4(T_{a,a},\ T_{a,k},\ T_{a,h})^{\mathrm{T}}$$

is defined as the maximum isometric torque scaled by three functions that are known to influence torque production. (The value "4" is that used by Bieryla [2], but should be "2" in a correct model.) Each active torque (ankle $T_{a,a}$, knee $T_{a,k}$, and hip $T_{a,h}$) is the result of the torques generated by forces applied by muscles in two directions (extension and flexion). The active torque with respect to an individual joint j and a force direction f generated at time t (s) with joint angle θ_j (rad) and angular velocity ω_j (rad/s) is

$$T_{a,j,f}(t, \theta_j, \omega_j) = T_{j,f,\max}\ r_{j,f}(\theta_j)\ h_j(\omega_j)\ A_j(t).$$

Depending on the activation at a given moment in time $A_j(t)$, either the extension or flexion formulas will be used to calculate $T_{a,j,f}$. Positive activation for a joint corresponds to extension for the ankle and hip and flexion for the knee, and vice versa for negative activation.

$T_{a,e,\max} = 0.125hwt_s$ is the maximum isometric torque (N $\cdot$ m) for the ankle in the extension direction, where h is the height of the subject (m), w is the weight of the subject (N), and $t_s = 1.2$ is a (unitless) variable based on the strength of the subject. Similarly, $T_{a,f,\max} = 0.022hwt_s$, $T_{k,e,\max} = 0.124hwt_s$, $T_{k,f,\max} = 0.060hwt_s$, $T_{h,e,\max} = 0.138hwt_s$, and $T_{h,f,\max} = 0.081hwt_s$. These maximum isometric torques are determined for a single lower extremity from a strength model of female older adults [1].

The torque-angle relation $r_{j,f}(\theta_j)$ (unitless) is obtained from previously published experimental data [20] and varies from zero to one. If the values for the angles fall outside the range allowed

by the model during the simulation, they are set to the limit of the model. The angle limits (rad) and relation formulas are

$$-0.52 < \theta_a < 0.61,\ 0.0 < \theta_k < 2.27,\ -0.17 < \theta_h < 2.27,$$

$$\begin{aligned}
r_{a,f} &= -0.1731(\theta_a^3) - 0.5882(\theta_a^2) + 0.3357(\theta_a) + 0.9502,\\
r_{a,e} &= 2.742(\theta_a^4) + 1.6115(\theta_{a^3}) - 2.8579(\theta_a^2) - 0.4996(\theta_a) + 0.9699,\\
r_{k,f} &= -0.2543(\theta_k^5) + 1.5215(\theta_k^4) - 2.9033(\theta_k^3) + 1.4916(\theta_k^2) + 0.2539(\theta_k) + 0,7643,\\
r_{k,e} &= 0.2334(\theta_k^4) - 0.4944(\theta_k^3) - 1.0148(\theta_k^2) + 2.051(\theta_k) + 0.1865,\\
r_{h,f} &= 0.4450(\theta_h^7) - 3.1958(\theta_h^6) + 8.5726(\theta_h^5) - 10.2750(\theta_h^4) + 4.7283(\theta_h^3) - 0.1678(\theta_h^2)\\
&\quad + 0.2293(\theta_h) + 0.6449,\\
r_{h,e} &= 0.2056(\theta_h^4) - 0.5625(\theta_h^3) - 0.2723(\theta_h^2) + 0.9446(\theta_h) + 0.6095.
\end{aligned}$$

The torque-angular velocity relation

$$h(\omega_j) = \begin{cases} (\omega_0 - \omega_j)/(\omega_0 + \Gamma\omega_j), & \omega_j/\omega_0 \le 1,\\ 0, & \omega_j/\omega_0 > 1, \end{cases}$$

also varies from zero to one and is obtained from [32], where ω_j is the angular velocity of the joint j (rad/s), ω_0 (± 20 rad/s) is the maximum angular velocity for all joints, and $\Gamma = 2.5$ is the shape factor describing the torque-angular velocity curve [32]. If the angular velocity and activation level have opposite signs, indicative of eccentric muscle contraction, $h(\omega_j)$ is increased to a maximum value of 1.5.

$A(t)$ (unitless) is allowed to vary from -1 to 1 to allow for flexion and extension of each joint. Because activation dynamics are not instantaneous, joint torque activation rate of change is limited to a maximum absolute value of 1/0.08 per second [7]. To enforce this rate of change, a bijective conformal mapping is employed [10]. Nineteen nodes, equally spaced 100 ms apart, are used to represent the joint torque activation profile of the ankles, knees, and hips combined (57 nodes total). Linear interpolation is used to define the activation levels between consecutive nodes. These nodes represent the variables for the objective function f.

The four optimization algorithms under consideration are used to attempt to determine the values for the joint activations (57 nodes) that minimize the performance criterion

$$\begin{aligned}
f &= w_1 \int_{t_0}^{t_f} (X_C(t) - X_A)dt + w_2 \int_{t_0}^{t_f} e(\theta(t))dt\\
&\quad + w_3 \int_{t_0}^{t_f} e(\dot{\theta}(t))dt + w_4 \int_{t_0}^{t_f} \left(\sum_{i=1}^{3} \dot{q}_i(t)^2\right)^{1/2} dt\\
&\quad + w_5 \int_{t_0}^{t_f} \dot{X}_C(t)dt + w_6 \int_{t_0}^{t_f} \ddot{X}_C(t)dt\\
&\quad + w_7 \sum_{i=1}^{3} \int_{t_0}^{t_f} (\tau_i(t)^2)dt
\end{aligned}$$

adapted from [39], [40], where $e(s(t)) = \sum_{i=1}^{3} \phi(s_i(t))$ and

$$\phi(s_i(t)) = \begin{cases} s_i(t)^- - s_i(t), & s_i(t) < s_i(t)^-, \\ 0, & s_i(t)^- \leq s_i(t) \leq s_i(t)^+, \\ s_i(t) - s_i(t)^+, & s_i(t) > s_i(t)^+, \end{cases}$$

with $s_i(t)^-$ and $s_i(t)^+$ representing the lower and upper physical bounds of the joint angles, respectively.

The first term in the objective function (unitless) minimizes the maximum horizontal displacement of the center of mass $X_C(t) - X_A$, where $X_C(t)$ is the center of mass of the body on the displacement platform (m) and X_A is the position of the ankle on the displacement platform (m). These values are taken from the experimental data. The second and third terms restrict joint angle $\theta(t)$ (rad) and angular velocity $\dot{\theta}(t)$ (rad/s) to remain within previously published physiologic limits. The joint angle minimums are -0.873 rad for the ankle, 0 rad for the knee and -0.524 rad for the hip. The joint angle maximums are 0.524 rad for the ankle, 2.269 rad for the knee and 2.182 rad for the hip. The angular velocity minimums are -6.2 rad/s for the ankle, -7.3 rad/s for the knee, and -8.5 rad/s for the hip. The angular velocity maximums are 8 rad/s for the ankle, 15 rad/s for the knee, and 10 rad/s for the hip. The fourth, fifth, and sixth terms minimize segment angular velocity $\dot{q}_i(t)$ (rad/s), center of mass velocity $\dot{X}_C(t)$ in (m/s), and center of mass acceleration $\ddot{X}_C(t)$ in (m/s^2), respectively, over the entire simulation. The seventh term minimizes the integral of the square of the joint torques $\tau_i(t)$, the sum of active and passive torques. The weights for f are $w_1 = 1000$, $w_2 = 500$, $w_3 = 500$, $w_4 = 50$, $w_5 = 100$, $w_6 = 25$, and $w_7 = 0.025$. The initial joint angle configuration is derived from experimental data, and the initial joint angular velocities are set to zero. The duration of the simulation time is $t_f = 1.8$ s, allowing for full recovery from the perturbation.

The minimum time to boundary (TTB) of the center of mass is used to quantify model performance with respect to balance. TTB is calculated as the instantaneous anterior-posterior (A/P) distance from the center of mass to the base of support divided by the instantaneous absolute value of the center of mass A/P velocity [31]. The base of support boundary is defined as the position of the first metatarsal based on the volunteer's anthropometry. The minimum TTB value describes the smallest amount of time for the participant to reach their limit of stability. Loss of balance occurs for TTB ≤ 0 s. A higher TTB indicates a longer period of time until the participant reaches their limit of stability. If the participant reaches the base of support a step occurs. Therefore, a decrease in TTB is seen as a degradation in balance.

In summary, given that x is the torque activation level function $A(t)$ discretized to nineteen distinct nodes for the three joints, each optimization algorithm attempts to minimize the stated $f(x)$, subject to the constraints that $-1 \leq x \leq 1$ and that within a single activation profile, the activation level between two consecutive discrete nodes may differ by no more than 1.25.

The best known solution to this problem is $f(x^*) = 1222.05$. This solution was found in the course of this investigation by a parallel run of pVTdirect, centered at a point found by the QNSTOP algorithm with a single experiment centered on the origin that was granted a function evaluation budget of 10^5. This measure $f(x)$ is unitless and only relevant when compared to other function evaluations to determine the relative fitness of a minimizing point.

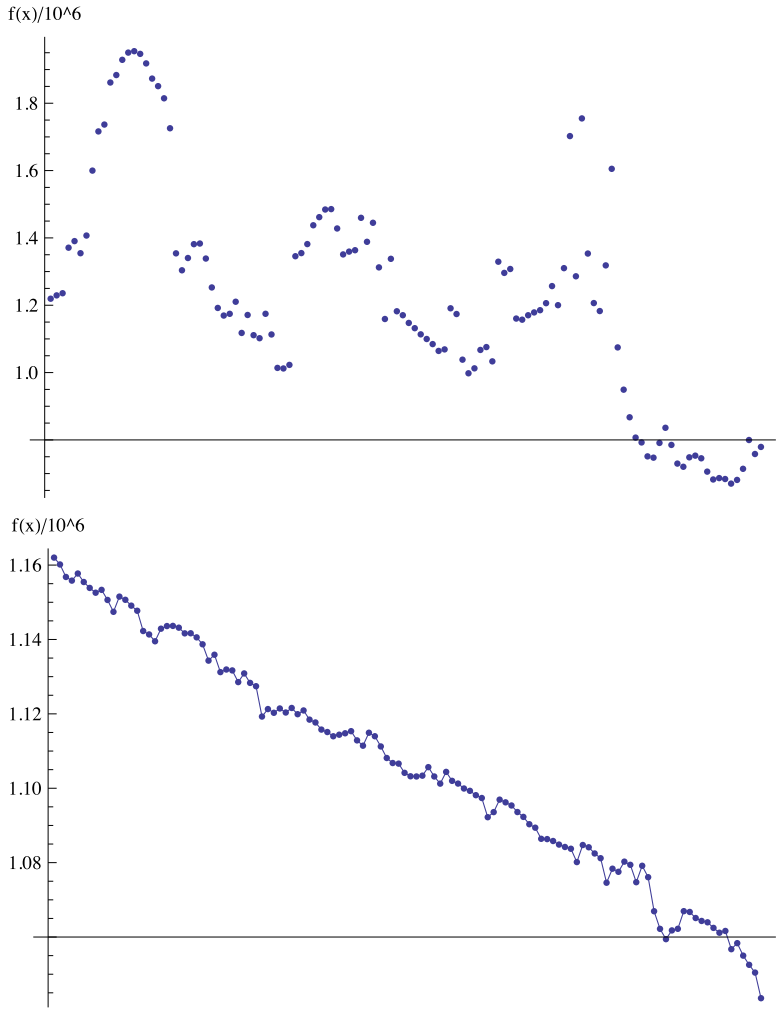

Figure 2. $f(x)$ along the line through two points of the biomechanics problem (top), and a zoomed view (bottom).

The highly variable nature of the biomechanics objective function $f(x)$ is shown in Figure 2, which shows the function evaluated along the line between two widely separated points and a zoomed view of a "smooth" part of the plot, demonstrating the presence of local noise.

3 Nonconvex Quadratic Minimization

The second problem of interest is the nonconvex box-constrained quadratic minimization problem:

$$(\mathcal{P}_b) : \min \left\{ P(x) = \frac{1}{2} x^{\mathrm{T}} A x - f^{\mathrm{T}} x : x \in X_b \right\},$$

where

$$X_b = \{ x \in \mathbb{R}^n | -1 \leq x_i \leq 1, \forall i = 1, \ldots, n \}.$$

Replacing the feasible set X_b by its vertices

$$\delta X_b = \{ x \in \mathbb{R}^n | x \in \{-1, 1\}^n \}$$

gives the integer programming problem

$$(\mathcal{P}_{\rm ip}) : \min \left\{ P(x) = \frac{1}{2} x^{\rm T} A x - x^{\rm T} f : x \in \delta X_b \right\}.$$

Using the canonical duality theory of Gao et al. [11], [12], the integer programming problem $(\mathcal{P}_{\rm ip})$ may be reformulated as

$$(\mathcal{P}^d_{\rm ip}) : \min \left\{ Q(\sigma) = \frac{1}{2} \sigma^{\rm T} \sigma - \sum_{i=1}^{n} \left| f_i + (B^{\rm T} \sigma)_i \right| : \sigma \in \mathbb{R}^m \right\},$$

where $\sigma = (\sigma_1, \ldots, \sigma_m)$, $f = (f_1, \ldots, f_n)$, and the $m \times n$ real matrix B is related to A. The reformulation is a nonconvex nonsmooth unconstrained minimization problem. Here $m = 57$, $n = 190$,

$$\hat{B} = \begin{bmatrix} 1 & -1 & 0 & -1 & 2 & 0 & 1 & -2 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 0 & -2 & 2 & 0 & 1 \\ 2 & 2 & -1 & -1 & 2 & -2 & 0 & 0 & -1 & 1 \end{bmatrix},$$

$$\begin{aligned} \hat{f} = 10^{-2}[&1.491803633709836,\ 3.0717213019723066, \\ &5.246230264266409,\ -6.718373452055033, \\ &3.969549763760797,\ 7.502845410079123, \\ &5.622108089244097,\ -1.9585631018739558, \\ &-2.729844702016424,\ 8.26721052052138], \end{aligned}$$

$$B = I_{19\times 19} \otimes \hat{B}, \text{ and } f = e_{19} \otimes \hat{f},$$

where $e_{19} = (1, \ldots, 1) \in \mathbb{R}^{19}$. This problem has exactly 2^{19} known local minimum points and the unique global minimum $Q(\sigma^{(1)})$ is located at

$$\sigma^{(1)} = (\,6 \quad -4 \quad 12 \quad \ldots \quad 6 \quad -4 \quad 12\,).$$

All the local minima are within 0.5% of the global minimum $Q(\sigma^{(1)}) = -1866.01$.

4 Wave Annihilation Problem

The wave annihilation problem studied here was first presented in [14]. That study developed a method for producing a coating of total thickness T distributed evenly in n layers of varying properties between two media to eliminate the reflection of waves over a band of frequencies [Ω_0, Ω_1] in one of those media. Reflections are eliminated entirely at n specific frequencies and reduced significantly for other frequencies within this band; as n approaches infinity, reflections within this band are eliminated entirely. This process is treated as an acoustic application in [14], but as is pointed out, it can easily be adapted to electromagnetism or any other phenomena governed by variants of the linear wave equation.

Given n, a crucial component to this process is to determine the n specific coatings such that the reflection $r = 0$ at frequency

$$\omega_k = \Omega_0 + \left(\frac{k-1}{n-1}\right)(\Omega_1 - \Omega_0)$$

for $k = 1, 2, \ldots, n$, where the complex-valued

$$r(n, \gamma, \kappa, \omega_k) = \frac{(\Gamma_-,\ \gamma_1)\prod_{j=1}^{n} A_j \begin{pmatrix} -1 \\ 1 \end{pmatrix}}{(\Gamma_-,\ \gamma_1)\prod_{j=1}^{n} A_j \begin{pmatrix} 1 \\ 1 \end{pmatrix}},$$

$$A_j = \begin{pmatrix} \gamma_j e_j^+ & \gamma_{j+1} e_j^- \\ \gamma_j e_j^- & \gamma_{j+1} e_j^+ \end{pmatrix}, \qquad \gamma_{n+1} = \Gamma_+, \qquad e_j^{\pm} = \exp\left(\frac{2\gamma_j \Delta x \omega_k i}{\kappa_j}\right) \pm 1,$$

$i = \sqrt{-1}$, Γ_+ and Γ_- are the impedances of the half-spaces surrounding the coating, $\Delta x = T/n$, and γ_j and κ_j are the impedance and stiffness of layer j, respectively. Note that unlike the other problems studied here, r is differentiable and the nonlinear system of equations can be solved using a variation of Newton's method [14]. An objective function $f = r * r$ may be formed by observing that the inner product of the complex vector $\big(r(\omega_1), \ldots, r(\omega_n)\big)$ with itself yields a scalar real value with a known minimum of zero where the original vector is zero. By choosing $n = 28$, a problem of 56 real variables is constructed that can be studied using the algorithms presented here, with both the impedance and the stiffness of the n layers being used as arguments to r, while the frequencies ω_k are determined directly from n. For our purposes, Γ_+ is chosen to be 1 and Γ_- is chosen to be 28.14776 (the ratio between the two is the same as the ratio between the reflectivity of water and steel), $T = 1$ m, and $\Omega_0 = 0.09091$ Hz while $\Omega_1 = 10\Omega_0$.

5 DIRECT

The DIRECT (DIviding RECTangles) algorithm by D.R. Jones [22] is a deterministic global optimization algorithm. The DIRECT algorithm does not require the computation of the gradient of the objective function, or even objective function values (ranking information is sufficient). It performs Lipschitzian optimization, but does not require knowledge of the Lipschitz constant.

The DIRECT algorithm works as follows [18]. The algorithm begins with an initial box normalized to the unit hypercube. The objective function F (assumed to satisfy a Lipschitz condition) is evaluated at the center of this box. The current minimum value is initialized to this value. An evaluation counter m and an iteration counter t are initialized to $m = 1$ and $t = 0$. The following process is repeated until m or t reaches some prespecified limit.

Selection. Identify the set S of "potentially optimal" boxes. Here "potentially optimal" means that (1) for some Lipschitz constant K, the box potentially contains a point with smaller objective function value than any other box, and (2) $F(c) - K \cdot L/2 \leq f_{min} - \epsilon|f_{min}|$, where F is the objective function, c is the center point of the box, K is the same Lipschitz constant, L is the box diameter, f_{min} is the current minimum value for the objective function, and ϵ is a small, nonnegative, fixed constant.

Sampling. Select one of the potentially optimal boxes B from S. For box B, identify the set I of dimensions with maximum side length L and let $\delta = \frac{1}{3}L$. Sample the function at all points of the form $c \pm \delta e_i$ for each $i \in I$, where c is the center of the rectangle and e_i is the ith standard basis vector. Update m.

Division. Divide the box containing the point c into thirds along the dimensions in I, beginning with the dimension with the least value of $\min\{F(c+\delta e_i), F(c-\delta e_i)\}$, and ending with the dimension with the greatest such value. Update the minimum value.

Iteration. Remove the box B from the set of potentially optimal boxes S. If $S = \emptyset$, then increment t and go to **Selection**. Otherwise, go to **Sampling**.

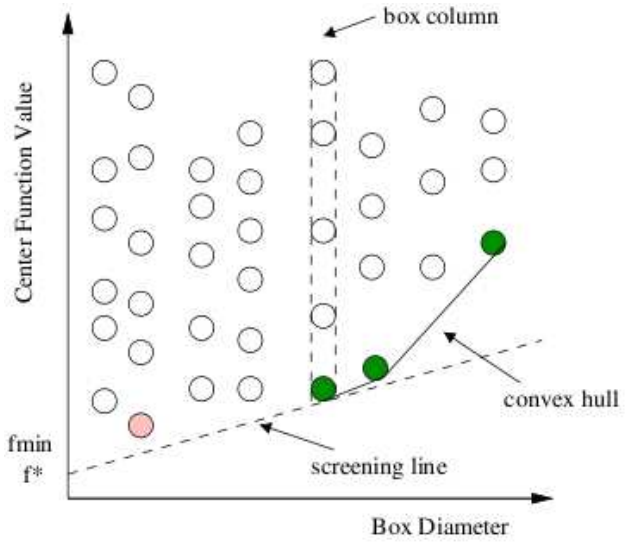

Figure 3. An example of a box scatter plot.

The method of choosing the sub-box according to both objective function value and box size gives DIRECT its local and global aspects. The DIRECT algorithm performs a convex hull computation to determine potentially optimal boxes without using the Lipschitz constant directly (see Figure 3 for an illustration). From Figure 3, it is clear that the convex hull consists of boxes with objective function values that are minimal amongst all boxes of the same size (notice that the set of boxes of the same size form a "box column"). Since every box is ultimately examined, DIRECT will not get stuck at a local optimum, but will instead perform a global search of the feasible set. Further details can be found in [22].

A parallel implementation of the DIRECT algorithm, pVTdirect, developed at Virginia Tech [18], is used here. The pVTdirect implementation contains some modifications to the original DIRECT algorithm in order to meet the needs of various applications and improve the performance on large scale parallel systems. First, an optional domain decomposition step is added to create multiple subdomains, each with a starting point for a DIRECT search. Empirical results have shown that this approach significantly improves load balancing among a large number of processors, and likely shortens the optimization process for problems with asymmetric or irregular structures. The **Selection** step has two new features. The first is an "aggressive" switch that originally appeared in Watson et al. [38]. The switch generates more function evaluation tasks that may help balance the workload under the parallel environment. The second is an adjustable ϵ, which is the minimum improvement required to update the current minimum objective function value. In

general, smaller ϵ values make the search more local and generate more function evaluation tasks. On the other hand, larger ϵ values bias the search toward broader exploration, exhibiting slower convergence. The value of ϵ is taken as zero by default, but can be specified by the user depending on problem characteristics and optimization goals.

In the serial version of DIRECT, **Sampling** samples one box at a time to conserve memory. In pVTdirect, more tasks are performed in parallel, so new points are sampled around all boxes in S along their longest dimensions during **Sampling**. This obviates the need for the step **Iteration** and simplifies the loop. Since box center function values may be identical or nearly so, the parallel **Sampling** may yield a different box sequence in each box column (i.e., ordered column of equal-sized boxes) as the parallel scheme varies. Thus boxes will be subdivided in a different order, causing pVTdirect to become nondeterministic. To maintain its deterministic property, pVTdirect performs lexicographical order comparison between box center coordinates, thereby keeping the boxes in the same sampling sequence on the same platform. However, the computed values can depend on the particular machine or compiler one uses, so the results for the same problem may vary slightly from system to system.

Finally, pVTdirect allows more stopping conditions for the termination of the algorithm. First, the minimum diameter variable `MIN_DIA` causes an exit when the diameter of the best box reaches the value `MIN_DIA`. Second, the objective function convergence tolerance variable `OBJ_CONV` causes an exit when the relative change in the optimum objective function value has reached the given value. These new stopping conditions address a complaint by Jones et al. [22] that the original stopping criterion for DIRECT was somewhat artificial and unconvincing for many real-world optimization problems.

6 Simulated Annealing

Simulated annealing is a stochastic algorithm, generally well suited to hard problems in biomechanics [19]. It begins with an initial feasible guess X_0 and the current minimum is set to $F(X_0)$. The algorithm then pseudorandomly generates points in a neighborhood of X_0 until it generates a feasible point X_1. If $F(X_1) \leq F(X_0)$, then the current point is set to X_1. If $F(X_1) - F(X_0) = d > 0$, then the current point is set to X_1 with probability $e^{-d/T}$, where T is the *temperature*. With probability $1 - e^{-d/T}$, X_1 is rejected and the algorithm continues to generate points around X_0 until a new feasible point is generated. This process is repeated until the distance between successive iterates is less than some small, fixed value.

The temperature begins at some high value T and is continually lowered throughout the search according to some temperature schedule. Since $e^{-d/T} \approx 1$ for $T \gg d$, a relatively large number of random moves will be made at the beginning of the search, when T is large. Thus the beginning of the search is primarily global in nature. As T decreases throughout the search, $e^{-d/T}$ gets closer to zero, and therefore the search becomes increasingly greedy, eventually performing similarly to a gradient descent method.

Simulated Parallel Annealing within a Neighborhood (SPAN) [19], one of the implementations used here, was developed for parallel computation. A straightforward serial simulated annealing algorithm consists of three nested loops: a *temperature reduction* loop that causes the temperature to gradually decline as the algorithm proceeds, an inner *search radius* loop that causes the neighborhood to gradually shrink while maintaining the same temperature, and an innermost *control*

loop that handles the perturbation of the variables to find the minimum point. The SPAN implementation parallelizes the independent search radius loop, so that all processors being utilized have all the information required to do a function evaluation (the current X and the radius of the search). The processors then communicate their results in a global communication (gather) before the search radius is adjusted so that roughly half of all the generated points in the previous neighborhood are acceptable in the new neighborhood. This communication overhead scales linearly with the number of processors involved, causing a notable degradation of performance on a large number of processors, especially with fast function evaluations [19]. For comparison's sake, a more traditional naive parallel implementation of simulated annealing (with random X_0) was also constructed, using the same underlying algorithm [13] with only one gather operation at the end of the optimization to collect all results. Both of the implementations used an initial temperature of $T = 5000$ and a cooling schedule of $T_{\text{next}} = 0.85(T_{\text{current}})$ for all three problems.

7 SPSA

Spall's simultaneous perturbation for stochastic approximation (SPSA) algorithm is, like simulated annealing, a stochastic global optimization algorithm [35]. SPSA is similar to the standard finite difference stochastic approximation (FDSA) algorithm [24] with one primary difference. The general form of both SPSA and the FDSA algorithm is $X_{k+1} = X_k - a_k \cdot g(X_k)$, where X_k is the variable vector, a_k is the kth element of the gain sequence a, and $g(X_k)$ is meant to approximate the gradient of the objective function at X_k. Whereas the FDSA algorithm perturbs the components of the vector X in the objective function $F(X)$ one at a time, SPSA performs a simultaneous perturbation of each component of X. This might appear to reduce the ability of the algorithm to search the problem space effectively when compared to component-by-component perturbation, but Spall [33]claims that "one properly chosen simultaneous random change in all the variables in a problem provides as much information for optimization as a full set of one-at-a-time changes of each variable".

As stated above, the main difference between SPSA and the FDSA algorithm is the method of perturbation of the components of X. The FDSA algorithm explores in each dimension around the point X_k, seeking the steepest descent (negative gradient) direction. Formally, the ith component of the (two-sided) gradient approximation for FDSA is computed as

$$g_i(X_k) = \frac{\hat{F}(X_k + c_k e_i) - \hat{F}(X_k - c_k e_i)}{2c_k},$$

where $\hat{F}(X) = F(X) + \text{noise}$, c_k is the kth element of a sequence c that converges monotonically to zero slowly as $k \to \infty$, and e_i is the ith standard basis vector. For an n-dimensional X, $2n$ objective function evaluations per iteration are required.

SPSA generates a vector v using a Monte Carlo method [34], evaluates the objective function at two points $X_k + v$ and $X_k - v$ at each iteration to approximate the gradient at X_k, and then adjusts X_k based on the resulting estimation of the gradient. Consequently, SPSA uses two objective function evaluations at each iteration. Formally,

$$g_i(X_k) = \frac{\hat{F}(X_k + c_k \Delta_k) - \hat{F}(X_k - c_k \Delta_k)}{2c_k \Delta_{ki}},$$

where $\Delta_k = (\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kn})$ is the user-specified random perturbation vector and $v = c_k\Delta_k$. The distribution of Δ_k must satisfy certain conditions in order to guarantee convergence; in particular, each component of Δ_k must be nonzero [33].

In the implementation of SPSA used for the biomechanics problem, an adaptation called *blocking* is employed that requires an extra function evaluation at each iteration. The idea is to "block" updates to X_k if the update will produce a new objective function value that is significantly worse than the current objective function value. This requires that the objective function be evaluated at X_k, as well as at $X_k + v$ and $X_k - v$. The extra function evaluation at each iteration increases the total number of evaluations by 33%, but can result in faster convergence of the algorithm. However, this technique can also reduce the likelihood that the algorithm will achieve a global minimum [34]. Projection is employed to prevent the algorithm from moving outside the feasible set.

When attempting to solve the nonconvex quadratic minimization problem dual and the wave annihilation problem, blocking is not used and instead an adaptation called *injected noise* is employed that simulates a random element in the objective function, with the intent of inducing the algorithm to abandon local minima. While the resulting implementation may take longer to converge to a minimum, the likelihood of global convergence is increased [26]. This adaptation is not necessary in the biomechanics problem because the noisiness inherent in the objective function fills the same role.

8 KNITRO

The KNITRO optimization package contains three optimization algorithms, but only one of them is utilized here, the direct interior point method [4]. Since the problems here are unconstrained except for upper and lower bound constraints, the sequential quadratic programming (SQP) method used by the KNITRO solver should be very efficient. However, the interior point method, like all the methods in the KNITRO package, assumes that the objective function is twice continuously differentiable. This is not the case for either the biomechanics problem or the nonconvex quadratic minimization problem dual, so a central difference method, included in the package, is invoked to supply second derivative input values; as a result, the efficacy of the direct interior point method suffers.

It is important to note that the direct interior point method employed by KNITRO, while very efficient at solving general constrained nonlinear optimization problems, loses some efficiency compared with other optimization algorithms for problems with only bound constraints [4]. Nevertheless, as a widely used commercial gradient-dependent optimization package, KNITRO represents a class of local optimization algorithms that apply to these difficult nonlinear problems.

The interior point direct algorithm seeks to find Karush-Kuhn-Tucker (KKT) points using sequential quadratic programming and trust region methods. As with all nonlinear optimization algorithms, the goal is to solve problems of the form

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & c_E(x) = 0, \\ & c_I(x) \geq 0, \end{aligned}$$

where here $f : \mathbb{R}^n \to \mathbb{R}$. The interior point direct algorithm first defines the barrier subproblem

$$\begin{aligned} \min_{x,s} \quad & f(x) - \mu \sum_{i=1}^{m} \ln s_i \\ \text{subject to} \quad & c_E(x) = 0, \\ & c_I(x) - s = 0, \end{aligned}$$

where $c_E : \mathbb{R}^n \to \mathbb{R}^l$, $c_I : \mathbb{R}^n \to \mathbb{R}^m$, the barrier parameter $\mu > 0$ and the vector of slack variables $s \in \mathbb{R}^m$ is positive.

Following [3], the KKT conditions for the above system can be written as

$$\begin{aligned} \nabla f(x) - {A_E}^{\mathrm{T}}(x)y - {A_I}^{\mathrm{T}}(x)z &= 0 \\ -\mu e + Sz &= 0 \\ c_E(x) &= 0 \\ c_I(x) - s &= 0, \end{aligned}$$

where $e = (1, \ldots, 1)^{\mathrm{T}}$, $S = \mathrm{diag}(s_1, \ldots, s_m)$, A_E and A_I are the Jacobian matrices corresponding to the equality and inequality constraint vectors respectively, and y and z represent vectors of Lagrange multipliers. The line search algorithm, used here, applies Newton's method to the above system, backtracking if necessary to ensure that $s, z > 0$, and ensuring that the merit function $\psi(x, s) = f(x) - \mu \sum_{i=1}^{m} \ln s_i$ is sufficiently reduced.

Applying Newton's method in the variables x, s, y, z gives

$$\begin{bmatrix} \nabla^2_{xx} L & 0 & -{A_E}^{\mathrm{T}}(x) & -{A_I}^{\mathrm{T}}(x) \\ 0 & Z & 0 & S \\ A_E(x) & 0 & 0 & 0 \\ A_I(x) & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_s \\ d_y \\ d_z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - {A_E}^{\mathrm{T}}(x)y - {A_I}^{\mathrm{T}}(x)z \\ Sz - \mu e \\ c_E(x) \\ c_I(x) - s \end{bmatrix}$$

where $L(x, s, y, z) = f(x) - \mu \sum_{i=1}^{m} \ln s_i - y^{\mathrm{T}} c_E(x) - z^{\mathrm{T}}(c_I(x) - s)$ is the Lagrangian of the above problem and $Z = \mathrm{diag}(z_1, \ldots, z_m)$.

If the inertia of the above matrix is $(n + m, l + m, 0)$, then the step d determined above is a descent direction for the merit function ψ. If so, the scalars

$$\begin{aligned} \alpha_s^{\max} &= \max\{\alpha \in (0, 1] : s + \alpha d_s \geq (1 - \tau)s\}, \\ \alpha_z^{\max} &= \max\{\alpha \in (0, 1] : z + \alpha d_z \geq (1 - \tau)z\}, \end{aligned}$$

are computed with $\tau = 0.995$. If $\min\{\alpha_s^{\max}, \alpha_z^{\max}\}$ is not too small, a backtracking line search is used to compute the steplengths $\alpha_s \in (0, \alpha_s^{\max}]$, $\alpha_z \in (0, \alpha_z^{\max}]$ that provide a sufficient decrease in ψ. The next iteration, with a reduced barrier variable, is then computed with

$$\begin{aligned} x^+ &= x + \alpha_s d_x, \quad s^+ = s + \alpha_s d_s, \\ y^+ &= y + \alpha_z d_y, \quad z^+ = z + \alpha_z d_z. \end{aligned}$$

However, if the inertia of the matrix is not of the desired form or the steplengths α_s or α_z are too small, a trust region method is employed to compute the current step d. This has the benefit of guaranteeing a successful step even in the presence of negative curvature or singularity.

The trust region method employed, which is also the standard step of the Interior/CG KNITRO algorithm, takes the following form. First, the normal step $v = (v_x, v_s)$ is computed by solving the subproblem

$$\min_{v} \quad \|A_E v_x + c_E\|_2^2 + \|A_I v_x - v_s + c_I - s\|_2^2 \tag{8.1}$$

$$\text{subject to} \quad \|(v_x, S^{-1} v_s)\|_2 \leq 0.8\Delta. \tag{8.2}$$

This subproblem is solved using a dogleg approach that minimizes (8.1) along a piecewise linear path composed of a steepest descent step in the norm used in (8.2) and a Newton step with respect to the same norm. The scaling $S^{-1}v_s$ in the norm tends to limit the extent to which the bounds on the slack variable are violated.

Once the normal step $v = (v_x, v_s)$ is computed, the tangential subproblem is then defined as

$$\min_{d_x, d_s} \quad \nabla f^t d_x - \mu e^t S^{-1} d_s + \frac{1}{2}\left(d_x^t \nabla_{xx}^2 L d_x + d_s^t S^{-1} Z d_s\right) \tag{8.3}$$

$$\text{subject to} \quad A_E d_x = A_E v_x \tag{8.4}$$

$$A_I d_x - d_s = A_I v_x - v_s \tag{8.5}$$

$$\|(d_x, S^{-1} d_s)\|_2 \leq \Delta. \tag{8.6}$$

To find the approximate tangential solution d, first the scaling $\tilde{d_s} \leftarrow S^{-1} d_s$ is applied to convert (8.6) into a sphere, and then a standard projected conjugate gradient method is applied to this transformed quadratic program, iterating in the linear manifold defined by (8.4)-(8.5). Finally, the step d is truncated if necessary to preserve $s > 0$.

9 QNSTOP

QNSTOP is a class of quasi-Newton methods for stochastic optimization. The implementation considered features several variations specific to global, deterministic optimization. In iteration k, QNSTOP methods compute the gradient vector $\hat{g}_k$ and Hessian matrix $\hat{H}_k$ of a quadratic model

$$\widehat{m}_k(X - X_k) = \hat{f}_k + \hat{g}_k^{\mathrm{T}}(X - X_k) + \frac{1}{2}(X - X_k)^{\mathrm{T}} \hat{H}_k (X - X_k), \tag{9.1}$$

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

$$X_{k+1} = X_k - \left[\hat{H}_k + \mu_k W_k\right]^{-1} \hat{g}_k, \tag{9.2}$$

where μ_k is the Lagrange multiplier of a trust region subproblem and W_k is a scaling matrix. In these cases, where the feasible set Θ is a 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.

9.1 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

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

for some $\gamma \geq 1$ where I_p is the $p \times p$ identity matrix. Here γ is fixed at 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

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

where $W_k \in \mathrm{W}_\gamma$. For this implementation $\tau_k = \tau > 0$ is fixed at $\tau = 1$ for each iteration.

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

Let $Y_k = (y_{k1}, \ldots, y_{kN})^{\mathrm{T}}$ denote the N-vector of responses where $y_{ki} = F(X_{ki}) + \text{noise}$. The response surface is modeled by the linear model $y_{ki} = \hat{f}_k + X_{ki}^{\mathrm{T}} \hat{g}_k + \epsilon_{ki}$ 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

$$\left(D_k^{\mathrm{T}} D_k\right) \hat{g}_k = D_k^{\mathrm{T}} Y_k \tag{9.3}$$

where

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

9.2 Updating the Model Hessian Matrix

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

$$-\eta I_p \preceq \hat{H}_k - \hat{H}_{k-1} \preceq \eta I_p \tag{9.4}$$

for some $\eta \geq 0$. Conceptually, this prevents the quadratic model from changing drastically from one iteration to the next. In [6], a variation of the SR1 (symmetric, rank one) update that satisfies this constraint is proposed. However, this constraint is simply relaxed in the deterministic case and the BFGS update is used, i.e., with the Hessian matrix updated according to

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

where

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

9.3 Step Length Control

QNSTOP methods utilize an ellipsoidal trust region concentric with the design region for controlling step length. Typically, in the stochastic case, the volume of the ellipsoid is adjusted from iteration to iteration. Here, the volume of the ellipsoid (controlled by some $\rho > 0$) is fixed with $\rho = 1$, and the following optimization problem is solved:

$$\min_{X \in \mathrm{E}_k(\rho)} \quad \hat{g}_k^{\mathrm{T}} (X - X_k) + \frac{1}{2} (X - X_k)^{\mathrm{T}} \hat{H}_k (X - X_k) \tag{9.5}$$

The solution to (9.5) is on the arc

$$X(\mu) = X_k - \left[\hat{H}_k + \mu W_k\right]^{-1} \hat{g}_k. \tag{9.6}$$

It remains to estimate μ_k such that $X(\mu_k)$ solves (9.5). Using [9] [Lemma 6.4.1], and a little manipulation, it can be established that there is a unique $\mu_k \geq 0$ such that $\|X(\mu_k) - X_k\|_{W_k} = \rho$, unless $\|X(0) - X_k\|_{W_k} \leq \rho$ in which case $\mu_k = 0$. Estimating μ_k is difficult, but well understood. Chapter 7 in [8] is a comprehensive treatment. In particular, Algorithm 7.3.6 in [8] is robust and easily implemented.

9.4 Updating the Experimental Design Region

The QNSTOP approach to constructing the ellipsoidal design regions is here considered. [37] considers confidence regions for the constrained minimizer of a quadratic model fit by regression. An early suggestion for the QNSTOP approach was a convenient ellipsoidal approximation of the confidence set for the minimizer of a quadratic subject to a trust region constraint.

However, if a linear model is fit by least squares and the model Hessian matrix is updated by a secant update then a different approach is warranted. This implementation uses an approximation derived in [6]. First, the approximation for the covariance matrix of $\nabla \widehat{m}_k(X_{k+1} - X_k)$,

$$V_k = 4\sigma^2 (D_k^{\mathrm{T}} D_k)^{-1}, \tag{9.7}$$

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

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

is an ellipsoidal approximation of the $1 - \alpha$ percentile confidence set for the minimizer where

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

Strictly using the updates for W_{k+1} above can lead to degenerate ellipsoids. To obtain useful design ellipsoids 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 $\mathrm{W}_\gamma \ni W_{k+1}$.

9.5 Algorithm Overview

The QNSTOP implementation used in this paper is summarized in Algorithm 1. Each run of the algorithm in the experiments was terminated when a budget of function evaluations B had been exhausted.

Algorithm 1. *QNSTOP-GLOBAL*

Step 0 (initialization) : *Fix* $\tau = 1$, $\rho = 1$, $N = 100$, *and* $\gamma \geq 1 = 20$. *Fix scaling matrix* $W_0 = I_p$ *and model Hessian matrix* $\hat{H}_0 = I_p$. *Choose an initial iterate* X_0 *and set* $k = 0$.

Step 1 (regression experiment) : *Uniformly sample* $\{X_{k1}, \ldots, X_{kN}\} \subset \mathrm{E}_k(\tau) \cap \Theta$. *Observe the response vector* $Y_k = (y_{k1}, \ldots, y_{kN})^{\mathrm{T}}$. *Compute* $\hat{g}_k$ *by solving (9.3).*

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

Step 3 (update iterate) : *Compute* μ_k *using the method described in Section 9.3, solve* $[\hat{H}_k + \mu_k W_k]s_k = -\hat{g}_k$, *and compute* $X_{k+1} = X_k + s_k$.

Step 4 (update subsequent design ellipsoid) : *Compute* $W_{k+1} \in \mathrm{W}_\gamma$ *using the approach described in section 9.4.*

Step 5 : *If* $(k+2)N < B$ *then increment* k *by 1 and go to* Step 1. *Otherwise, the algorithm terminates.*

Figure 4 shows a typical progression of QNSTOP over 20 iterations. 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.

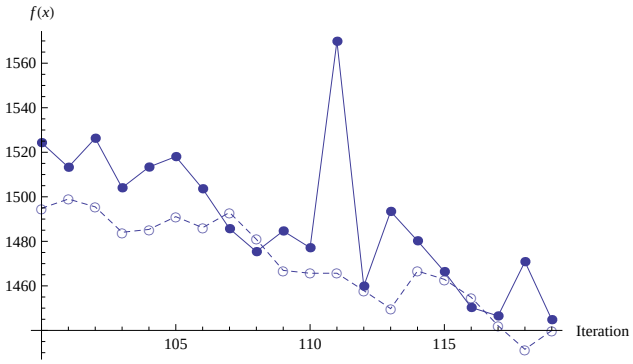

Figure 4. A typical QNSTOP progression.

10 Experimental Results and Discussion

Each of the optimization algorithms here is run 50 times using starting points selected from a Latin hypercube design based on the calculated bounds for each of the three problems. For the biomechanics problem, the bounds on each variable run from -1 to 1. While the nonconvex quadratic minimization problem dual is unconstrained and the reflection annihilation problem has nonnegative variables, both have bounds that can be calculated a priori that allow the Latin hypercube design to be constructed. For the nonconvex quadratic minimization problem dual, the

bounds on each variable are ± 41.569. For the reflection annihilation problem, the lower and upper bounds are vectors of 0 and 40, respectively.

DIRECT ($\epsilon = 10^{-4}$), given the constraints of the initial box, is run for each of the problems centered on a point derived from a variation of the Latin hypercube design used for the starting points of the other algorithms: the fifty starting points are all divided by 100 to allow as much of the space to be explored as possible while still differentiating the starting points from each other. The naive simulated annealing algorithm uses the Latin hypercube starting points for only one of the eight processors run in parallel. SPAN similarly uses only eight processors per experiment, since the utilization of more processors results in extreme communication overhead; the number of processors for the naive implementation was limited to the same number of processors that SPAN employed to directly see the advantage of one longer annealing versus the eight shorter ones. The naive parallel implementation of SPSA uses the Latin hypercube starting points in only one of the 640 processors used in parallel for each experiment. KNITRO 8.0 uses the Latin hypercube starting point for only one of its randomly generated multistart points per experiment, and the rest are generated by the built-in pseudorandom generation. Each of the fifty experiments was run on only a single processor for KNITRO 8.0. In a slight difference from the other algorithms, QNSTOP uses 10,000 samples for each of 100 Latin hypercube starting points per experiment, forgoing the standard Latin hypercube starting point in order to provide a more global search strategy. QNSTOP is currently implemented only as a serial algorithm, so going beyond 100 processors per experiment is not possible.

It must be noted that the SPAN implementation, while using the same algorithm as the naive implementation, chooses its evaluation points based on a number of different seeds for the pseudorandom number generator equal to the number of processors, to avoid duplicating values. The naive implementation, which uses the same parameters as the SPAN implementation and performs eight annealings (at 125,000 function evaluations each), therefore examines different points than the single annealing performed over multiple processors for the SPAN implementation. In fact, without changing the random seed input for the SPAN implementation, simply changing the number of processors used can change the outcome based on the change to the local seeds. Of course, given the identical algorithms, the tradeoff here is simply one of time for a single annealing performed in parallel versus the time for multiple serial annealings performed simultaneously.

Table 1 displays the number of function evaluations used by each optimization algorithm as it pursues the global minimum over each of the 50 experiments. Each optimization algorithm is given a function evaluation budget of 10^6 for each experiment and run until it reaches the function evaluation budget or terminates according to the rules of the algorithm.

Tables 2–4 display, for each of the problems described, the minimum, maximum, first, second, and third quartile objective function values for each of the algorithms over the fifty experiments, along with the best known minimum for each of the problems and the method of discovery. Note that the experiments that discovered the best known minimum for these problems are not part of the set of experiments listed here, as they tend to have higher function evaluation limits to allow for exhaustive searching.

The worst performer was the naive parallel SPSA, for several reasons. First, SPSA is designed to be a local optimization algorithm, here used in three global optimization applications; attempts to increase the number of starting points to compensate for this shortcoming were rather unsuccessful in the face of the number of dimensions involved in each problem space. Second, the approach taken by SPSA suffers in any case when a large number of variables is encountered, causing it to be much slower to discover local minima. Third, the naive parallel implementation

Table 1
Average function evaluations per experiment for each problem.

	DIRECT	SPAN	SA (naive)	SPSA	QNSTOP	KNITRO
Biomech	1008497	1000000	1000000	1000000	1000000	1010492
Quad Dual	1000593	1000000	1000000	1000000	1000000	1075451
Wave	1001585	1000000	1000000	1000000	1000000	1000781

Table 2
Results for the biomechanics problem.
Best known minimum value: 1222.05 *(QNSTOP/DIRECT).*

	Minimum	1^{st} quartile	2^{nd} quartile	3^{rd} quartile	Maximum
DIRECT	8501	24227	28594	34321	77567
SA (naive)	12606	14414	15677	16352	17723
SPAN	3295	4460	4833	5567	82297
SPSA	33447	46040	56503	61939	84676
KNITRO	19545	28180	30688	35390	40234
QNSTOP	10134	13359	14710	17185	45828

Table 3
Results for the nonconvex quadratic minimization problem dual.
Best known minimum value: -1866.01 *(DIRECT).*

	Minimum	1^{st} quartile	2^{nd} quartile	3^{rd} quartile	Maximum
DIRECT	-1864.32	-1863.03	-1862.21	-1861.79	-1860.00
SA (naive)	-1146.16	-1110.06	-1095.66	-1084.64	-1030.77
SPAN	-1861.53	-1859.93	-1859.20	-1858.69	-1857.43
SPSA	253.30	604.44	688.00	759.65	893.00
KNITRO	-1864.74	-1827.05	-1825.67	-1808.06	-1609.60
QNSTOP	-1863.90	-1862.63	-1862.21	-1861.37	-1860.52

Table 4
Results for the wave annihilation problem.
Best known minimum value: 0 *(DIRECT).*

	Minimum	1^{st} quartile	2^{nd} quartile	3^{rd} quartile	Maximum
DIRECT	$8.19 * 10^{-7}$	$1.02 * 10^{-3}$	$5.76 * 10^{-3}$	$5.74 * 10^{-2}$	$2.7 * 10^{-1}$
SA (naive)	26.87	27.26	27.36	27.53	27.76
SPAN	2.71	3.35	25.20	26.25	26.62
SPSA	12.94	523.35	2902.51	8031.26	206193.00
KNITRO	27.09	28.00	28.00	28.00	28.00
QNSTOP	26.64	27.10	27.19	27.30	27.48

utilized here simply divided the number of function evaluations by the number of processors; while

this may have made the algorithm more likely to start close to a good minimum, the tradeoff was arguably not favorable in any instance except the wave annihilation problem, the most amenable to traditional derivative-based solution methods. Finally, the injected noise technique, while arguably useful for the result obtained for the wave annihilation problem, was certainly not helpful for the quadratic dual. SPSA's best performance was merely mediocre. To determine the best performer, however, requires a look at each problem individually.

Biomechanics problem. SPAN did very well on the biomechanics problem compared to the naive simulated annealing algorithm. In fact, the vast majority of the solutions found by SPAN beat even the best solutions found by the rest of the algorithms for that problem. It's of interest to note the impact that local searching has on improving the minimums for the biomechanics problem, since the only difference between SPAN and the naive annealing is the longer local search time, where the annealing is done over a fairly small neighborhood and the temperature is quite "cool." DIRECT similarly benefitted from the transition to local searching inherent in its execution, resulting in a good best result and reasonable results for most of its experiments. The QNSTOP global strategy employed here, Latin hypercube sampling, is perhaps not the best strategy to use to show the strengths of this algorithm when applied to this problem; the best known minimum for this problem is the result of a previous QNSTOP experiment. Even so, the admittedly inferior global strategy utilized here yielded reasonable results compared to the multistart KNITRO, SPSA, and naive parallel simulated annealing strategies, and consistently beat the experiments performed by DIRECT. Note, however, that DIRECT found the best known value in one large run, and none of the other methods, in these or other experiments, ever found this best value. The deterministic DIRECT is guaranteed to monotonically decrease the objective function with more work, whereas the nondeterministic methods (SA, SPAN, SPSA, QNSTOP) are only *likely* to do so.

Quadratic dual problem. Recall that for the nonconvex quadratic minimization problem dual, all the local minima are within 0.5% of the global minimum -1866.01. The layout of this problem is particularly devastating for SPSA, which has difficulty settling into the local minimum points at which the function is nondifferentiable, resulting in SPSA's worst showing; the injected noise technique, which was employed in an attempt to escape the vast number of local minima, did not encourage SPSA to settle for any minimum within the number of function evaluations employed. DIRECT, QNSTOP, and KNITRO are almost tied for the best results, although DIRECT and QNSTOP are much more consistent in their performance, as KNITRO encounters difficulties as well with the nondifferentiability of the function at the local minimum points. Finally, once again the lack of local search in the naive parallel simulated annealing is revealed, as the naive annealing experiments found the broad basins that held the local minimum points but failed to refine to a local minimum point. Again DIRECT, in a larger run, found the best known value (theoretical global minimum, for this problem), and similar comments as for the biomechanics problem also apply here.

Wave annihilation problem. DIRECT consistently found the optimum solution for the wave annihilation problem, while the other algorithms (with the notable exception of SPSA) were consistent in finding local minima near 28. SPSA benefitted the most with this problem from the "shotgun" approach utilized here, coming in third for the best result found, with the notable downside that its performance was otherwise abysmal. QNSTOP, KNITRO, and the naive simulated annealing all found local minima near 28, while SPAN found good minima in about half of its experiments and near 28 in the other half.

The following general conclusions are immediately obvious. Firstly, SPSA is entirely unfit for an optimization problem with a large number of dimensions and a large number of minima. This is

not surprising, as this is not the purpose for which SPSA was developed. Secondly, any approach that relied on local information for gradient approximations, namely SPSA and KNITRO, had an unfortunate tendency to restrict its search prematurely and thus lost significantly in global exploration compared to the other algorithms presented here, particularly in the biomechanics and wave annihilation problems. While the multistart strategy allowed for some automatic global searching, it was clearly not enough to overcome the difficulties inherent in these problems. Finally, the multistart strategy employed by QNSTOP needs refinement before drawing any conclusions about the true value of this algorithm.

Some general conclusions are also in order about the two newest algorithms considered here — the massively parallel implementation pVTdirect of DIRECT, and the quasi-Newton stochastic algorithm QNSTOP. Because pVTdirect maintains a history of all samples, it makes more efficient use of samples than highly parallel independent sampling stochastic algorithms do, and thus is likely to scale better with more processors. Deterministic algorithms like DIRECT may perform very well on noisy functions (like the biomechanics problem here), and local stochastic algorithms like QNSTOP may perform very well on global optimization problems (as here). In the context of ever increasing parallelism, higher dimensions, and global optimization, algorithms like (deterministic) pVTdirect and (stochastic) QNSTOP, and hybrids thereof, seem well worth pursuing.

Acknowledgements This work was supported in part by AFOSR Grant FA9550-09-1-0153, AFRL Grant FA8650-09-2-3938, and the computing facilities at Indiana University and Virginia Polytechnic Institue & State University.

References

1. D. E. Anderson; M. L. Madigan; M. A. Nussbaum. 2007, "Maximum voluntary joint torque as a function of joint angle and angular velocity: model development and application to the lower limb", Journal of Biomechanics, 40, no. 14, 3105–3113.
2. K. A. Bieryla. 2009, "An investigation of perturbation-based balance training as a fall prevention intervention for older adults", Ph.D. thesis, Department of Mechanical Engineering, VPI & SU, Blacksburg, VA.
3. R. H. Byrd; J. Nocedal; R. A. Waltz. 2006, *Large-Scale Nonlinear Optimization*, Springer-Verlag, 35–59.
4. R. H. Byrd; M. E. Hribar; J. Nocedal. 1999, "An interior point method for large scale nonlinear programming", SIAM Journal on Optimization, 9, no. 4, 877–900.
5. R. H. Byrd; J.C. Gilbert; J. Nocedal. 2000, "A trust region method based on interior point techniques for nonlinear programming", Mathematical Programming A, 89, 149–185.
6. B. S. Castle. 2012, *Quasi-Newton Methods for Stochastic Optimization and Proximity-Based Methods for Disparate Information Fusion*, Ph.D.thesis, Indiana University, Bloomington, IN.
7. K. B. Cheng. 2008, "The relationship between joint strength and standing vertical jump performance", Journal of Applied Biomechanics, 24, no. 3, 224–233.
8. A. R. Conn; N. I. M. Gould; P. L. Toint. 2000, "Trust-Region Methods", MPS-SIAM Series on Optimization, SIAM, Philadelphia.
9. J.E. Dennis, Jr.; R.B. Schanbel. 1996, *Numerical methods for unconstrained optimization and nonlinear equations (2nd ed.)*, SIAM, Philadelphia.
10. D.R. Easterling; L.T. Watson; M. L. Madigan. 2010, "The DIRECT algorithm applied to a problem in biomechanics with conformal mapping", in Proc. 2010 International Conference on Scientific Computing, CSC '10, H. Arabnia and G. Grawanis (eds.), CSREA Press, USA, 2010, 128–133.
11. D. Y. Gao. 2000, *Duality Principles in Nonconvex Systems: Theory, Methods, and Applications*, Kluwer Academic Publishers, 472 pp.
12. D. Y. Gao. 2000, "Canonical dual transformation method and generalized triality theory in nonsmooth global optimization", Journal of Global Optimization, 17(1/4), 127–160.
13. W.L. Goffe; G.D. Ferrier; J. Rogers. 1994, "Global optimization of statistical functions with simulated annealing", Journal of Econometrics, 60, 65–100.
14. W. W. Hager; R. Rostamian; D. Wang. 2000., "The wave annihilation technique and the design of nonreflective coatings", SIAM Journal on Applied Mathematics, 60, no. 4, 1388–1424.

15. J. He; A. Verstak; L. T. Watson; M. Sosonkina. 2008., "Design and implementation of a massively parallel version of DIRECT", Computational Optimization and Applications, 40, no. 2, 217–245.
16. J. He; A. Verstak; L.T. Watson; M. Sosonkina. 2009., "Performance modeling and analysis of a massively parallel DIRECT: part 1", International Journal of High Performance Computing Applications, 23, no. 1, 14–28.
17. J. He; L.T. Watson; N. Ramakrishnan; C. A. Shaffer; A. Verstak; J. Jiang; K. Bae; W.H. Tranter. 2002, "Dynamic data structures for a direct search algorithm", Computational Optimization and Applications, 23, no. 1, 5–25.
18. J. He; L. T. Watson; M. Sosonkina. 2009, "Algorithm 897: VTDIRECT95: serial and parallel codes for the global optimization algorithm DIRECT", ACM Transactions on Mathematical Software, 36, no. 3, Article 17, 1–24.
19. J. S. Higginson; R.R. Neptune; F.C. Anderson. 2004., "Simulated parallel annealing within a neighborhood for optimization of biomechanical systems.", Journal of Biomechanics., 38, no. 9, 1938–1942.
20. M. G. Hoy; F. E. Zajac; M. E. Gordon. 1990, "A musculoskeletal model of the human lower extremity: the effect of muscle, tendon, and moment arm on the moment-angle relationship of musculotendon actuators at the hip, knee, and ankle", Journal of Biomechanics, 23, no. 2, 157–169.
21. L. Ingber. 1993, "Simulated annealing: practice versus theory", Mathematical Computer Modeling, 18, no. 11, 29–57.
22. D. R. Jones; C. D. Perttunen; B. E. Stuckman. 1993, "Lipschitzian optimization without the Lipschitz constant", Journal of Optimization Theory and Applications, 79, no. 1, 157–181.
23. D. R. Jones. 2001, "The DIRECT global optimization algorithm", Encyclopedia of Optimization, Vol. 1, Dordrecht : Kluwer Academic Publishers, 431–440.
24. J. Kiefer; J. Wolfowitz. 1952, "Stochastic estimation of a regression function", Annals of Mathematical Statistics, 23, 462–466.
25. S. Kirkpatrick; C. D. Gelatt; M.P. Vecchi. 1983, "Optimization by simulated annealing", Science, New Series, 220, no. 4598, 671–680.
26. J. L. Maryak; D. C. Chin. 2008, "Global random optimization by simultaneous perturbation stochastic approximation", IEEE Transactions on Automatic Control, 53, no. 3, 780–783.
27. M. J. Pavol; T. M. Owings; M.D. Grabiner. 2002, "Body segment inertial parameter estimation for the general population of older adults", Journal of Biomechanics, 35, 707–712.
28. N. R. Radcliffe; D. R. Easterling; L. T. Watson; M. L. Madigan; K. A. Bieryla. 2010, "Results of two global optimization algorithms applied to a problem in biomechanics.", in Proc. 2010 Spring Simulation Multiconference, High Performance Computing Symp, A. Sandu, L. Watson, and W. Thacker (eds), Soc. for Modelling and Simulation Internat., Vista, CA, 2010, 117–123.
29. D. J. Ram; T. H. Sreenivas; K. G. Subramaniam. 1996, "Parallel simulated annealing algorithms", Journal of Parallel and Distributed Computing, 37, 207–212.
30. R. Riener; T. Edrich. 1999, "Identification of passive elastic joint moments in the lower extremities", Journal of Biomechanics, 32, no. 5, 539–544.
31. B. W. Schulz; J. A. Ashton-Miller; N. B. Alexander. 2006, "Can initial and additional compensatory steps be predicted in young, older, and balance-impaired older females in response to anterior and posterior waist pulls while standing?", Journal of Biomechanics, 39, no. 8, 1444–1453.
32. W. S. Selbie; G. E. Caldwell. 1996, "A simulation study of vertical jumping from different starting postures", Journal of Biomechanics, 29, no. 9, 1137–1146.
33. J. C. Spall. 1998, "An overview of the simultaneous perturbation method for efficient optimization", John Hopkins APL Tech. Digest, 19, no. 4, 482–492.
34. J. C. Spall. 1998, "Implementation of the simultaneous perturbation algorithm for stochastic optimization", IEEE Transactions on Aerospace and Electronic Systems, 34, no. 3, 817–823.
35. J. C. Spall. 1987, "A stochastic approximation technique for generating maximum likelihood parameter estimates", in Proc. American Control Conference (Minneapolis, MN, June 10-12), 1161–1167.
36. J. C. Spall. 1992, "Multivariate stochastic approximation using simultaneous perturbation gradient approximation", IEEE Trans. Autom. Control, 37, no. 3, 332–341.
37. D. M. Stablein; W. H. Carter, Jr.; G. L. Wampler. 1983, "Confidence regions for constrained optima in response-surface experiments", Biometrics, 39, 759–763.
38. L. T. Watson; C. A. Baker. 2001, "A fully-distributed parallel global search algorithm", Engineering Computations, 18, no. 1–2, 155–169.
39. F. Yang; F. C. Anderson; Y. C. Pai. 2007, "Predicted threshold against backward balance loss in gait", Journal of Biomechanics, 40, no. 4, 804–811.
40. F. Yang; F. C. Anderson; Y. C. Pai. 2008, "Predicted threshold against backward balance loss following a slip in gait", Journal of Biomechanics, 41, no. 9, 1823–1831.