Gradient Estimation in Global Optimization Algorithms

Megan Hazen, *Member, IEEE* and Maya R. Gupta, *Member, IEEE*

***Abstract*— The role of gradient estimation in global optimization is investigated. The concept of a regional gradient is introduced as a tool for analyzing and comparing different types of gradient estimates. The correlation of different estimated gradients to the direction of the global optima is evaluated for standard test functions. Experiments quantify the impact of different gradient estimation techniques in two population-based global optimization algorithms: fully-informed particle swarm (FIPS) and multiresolutional estimated gradient architecture (MEGA).**

I. Introduction

In this paper we explore the importance of using gradient information to direct searches for global optima, and how different types of gradient estimates may affect the performance of population-based global algorithms. Consider the optimization problem of finding a global minimum $x^* \in \mathcal{R}^D$ of an objective function $f(x) \in \mathcal{R}$ such that

$$x^* = \underset{x \in \mathcal{S}}{\operatorname{argmin}} f(x), \tag{1}$$

where $\mathcal{S} \subset \mathcal{R}^D$ is a compact set. Many functions of interest will also have multiple local minima.

A standard approach for finding local minima is to calculate or estimate a gradient and follow the path of steepest descent. Line search algorithms have been developed for problems without gradient information as well, in which steps are taken in conjugate directions, or descent directions are estimated in trust regions [13]. This paper proposes a regional gradient for use in these line searches. In global optimization problems this line search may be combined with a method of escaping from, or searching over, local minima so that the global location is found. Understanding the role that a gradient search plays in optimization is important to understanding global search methods.

Many real-world problems of interest have functions with some structure, such that using more information about the function will result in a more efficient search. However, in many problems the objective function $f(x)$ is modeled as a black box: $f(x)$ can only be obtained for a specific x by running a program, taking measurements, or modeling a system. Functional information, in particular the function gradient, is unobtainable, but estimates of gradients based on previously-evaluated operating points may be useful. In this paper we compare a few different methods of estimating a gradient direction.

Many gradient-free global optimization methods have been developed [11], [17], [2]. Purely stochastic methods are able to converge to a global minimum, but it is our contention that moving downhill can direct the search more quickly toward the optimal area. In fact, many popular global optimization algorithms implicitly or explicitly estimate which direction is downhill at a given operating point. For example, finite difference methods require $2D$ extra function evaluations to estimate a gradient. Simultaneous perturbation stochastic approximation (SPSA) uses a modified finite difference method that approximately estimates the downhill direction using only two extra function evaluations [14]. Trust region optimization uses a linear model of previously evaluated operating points to determine a search direction, and has proven convergence on single-minima functions [13].

In this paper we focus on two population-based optimization algorithms which incorporate rough estimates of the downhill direction. Specifically, particle swarm optimization (PSO) [4], and its variant fully informed particle swarm (FIPS) [10], use a point difference approach to direct the search, while the multiresolution estimated gradient architecture (MEGA) [8] algorithm uses a regression-based gradient estimate. The FIPS and MEGA algorithms were chosen for this work because the structure of these algorithms allows for the insertion of different gradient estimates. The goal of this paper is not to compare these two algorithms, but rather to explore the interaction of the gradient estimates with two different global optimization algorithms.

First, to enable discussion of gradients from a multiresolutional perspective, we introduce the concept of a regional gradient in Section II. An experiment compares different gradient estimation techniques for global search using randomly generated sample points in Section III. The more practical question of the impact of different gradient estimates in conjunction with the architecture of a global optimization algorithm is considered for the FIPS and MEGA algorithms in Section IV. Section V ends the paper with a discussion of the findings and some open questions.

II. Regional Gradients and Multiresolutional Searching

Global optimization algorithms often search at many different resolutions. A coarse resolution search finds the most promising area of a large region, while a finer resolution search finds the local minima in a small region. In this section we discuss the challenges of searching at multiple resolutions and define a regional gradient that enables analysis of multiresolutional search.

Some algorithms change the search resolution sequentially: first finding a region of attraction, then searching it locally to find that local optima, then broadening the search again to find another region of attraction. Tabu search

Megan Hazen is with the Applied Physics Laboratory, University of Washington, Seattle, WA 98195, USA (email: megan@apl.washington.edu).

Maya R. Gupta is with the Dept. of Electrical Engineering, University of Washington, Seattle, WA 98195, USA (email: gupta@ee.washington.edu).

describes this process explicitly with the processes of 'intensification' (combining known good solutions to explore their local region), and 'diversification' (searching previously unexplored areas) [5]. Some algorithms accomplish multiresolutional search with a hybrid approach, combining stochastic movement for the global search and more structured gradient descent movements for honing in on optima [16], [12]. Other global optimization algorithms, such as PSO, search at different resolutions simultaneously, moving the particles toward both locally-promising and globally-promising regions. The authors recently proposed a population-based algorithm that explicitly directs the search using gradient estimates, termed multiresolution estimated gradient architecture (MEGA) [8]. MEGA alternates between fine and coarse resolution searches by estimating gradients over different regions of the search space.

A. Regional Gradient Definition

The gradient defines the downhill direction at a particular point in the search space. When the functional space has local minima or noise, the gradient may be ineffective or even misleading as a search direction. Rather, we hypothesize that one is interested in estimating what constitutes the descent direction over different regions of the search space. To this end, we define a regional gradient for the region S_p to be the vector $\beta^* \in \mathcal{R}^D$ that best fits a hyperplane to the function $f(x)$ over the region $S_p \subset \mathcal{R}^D$, that is:

$$[\beta^*, \beta_0^*] = \underset{\beta,\beta_0}{\operatorname{argmin}} \int_{x \in S_p} (f(x) - \beta^T x - \beta_0)^2 dx. \qquad (2)$$

The regional gradient is well-defined whether or not the function f is differentiable. If f is differentiable, then the following proposition holds:

Proposition: If the gradient of $f(x)$ exists and is bounded for all $x \in S_p$, then the regional gradient β^* given in (2) is the average of the gradients in the region S_p:

$$\beta^* = \frac{1}{\text{volume}(S_p)} \int_{x \in S_p} \nabla f(x) dx. \qquad (3)$$

Proof: The proposition follows from the fact that the mean minimizes the integral of squared error if the integral is well-defined. That is, (2) is solved by

$$x^T \beta^* + \beta_0^* = \frac{\int_{x \in S_p} f(x) dx}{\int_{x \in S_p} dx}. \qquad (4)$$

Take the gradient of both sides, and because of the proposition's assumptions, one can swap the gradient and integration operators to yield (3).

The proposition establishes that the regional gradient β^* is a robust description of the descent direction in the sense that it is the average gradient in the region. In Figure 1 two regional gradient estimations are shown, for different resolutions, where the resolution is determined by size of the marked circular region. The resolution in this description is similar to the trust region used in linear-interpolation trust region methods, although the precise definition of trust regions varies with algorithm specifics [13].

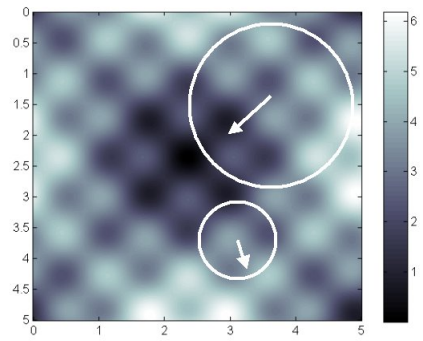

Fig. 1. This figure shows the sinusoidal test function and two estimated regional gradients of the function (gradient directions are shown as arrows, with regions outlined with circles). The finer regional gradient points towards a local minima, while the coarser regional gradient may help avoid it.

B. Approximate Regional Gradients in Global Optimization

Many global optimization algorithms can be interpreted as using approximate regional gradients to direct their search, where the regional gradients are calculated for different resolutions.

Simultaneous perturbation stochastic approximation (SPSA) estimates the gradient by stochastically choosing a direction in which to perturb the operating point to calculate a finite difference gradient [15]. Let $\nabla f(x)[d]$ denote the d^{th} vector component of the gradient $\nabla f(x)$. Then the SPSA gradient estimate $\widehat{\nabla f(x_k)}$ for the k^{th} iteration of the algorithm has dth vector component

$$\widehat{\nabla f(x)}[d] = \frac{f(x_k + \delta_k \Delta_k[d] e_d) - f(x_k - \delta_k \Delta_k[d] e_d)}{2\delta_k \Delta_k[d]}, \qquad (5)$$

where e_d is a D-component vector with 1 for the dth component and 0 for all other components, δ_k is a scalar that generally shrinks as the number of iterations k increases, and Δ_k is a random D-component perturbation vector. The SPSA gradient estimate can be interpreted as an estimate of the gradient over a region defined by a hypersphere with center x_k, and radius $\delta_k \Delta_k$.

Point-differences in global optimization can also be interpreted as approximations of the regional gradient. A point-difference is similar to the SPSA estimation, but does not require new points to be evaluated. Given two previously-evaluated sample pairs $(x_1, f(x_1))$ and $(x_2, f(x_2))$ where $f(x_2) < f(x_1)$, the point-difference is defined to be the vector $x_1 - x_2$, or the normalized version $(x_1 - x_2)/\|x_1 - x_2\|$. The point-difference estimate is a simple and elegant method of obtaining a notion of "downhill," and is used in PSO. This point-difference estimate can be considered a regional gradient for the local

region of x_1, x_2.

PSO: PSO is an evolutionary algorithm in which many agents, or particles, search the space collaboratively [4]. The particles' behavior is modeled after the natural flocking behavior of birds. If on the i^{th} iteration the location of a particle is x_i, then on the $(i+1)^{th}$ iteration the particle's position evolves according to the equations:

$$\begin{aligned} x_{i+1} &= x_i + v_{i+1}, \\ v_{i+1} &= c_0 v_i + c_1 r_1 (p^* - x_i) + c_2 r_2 (g^* - x_i), \end{aligned} \quad (6)$$

where p^* is the current personal best for that particle, g^* is the current global best for all the particles, r_1, r_2 are randomly drawn, and c_0, c_1, c_2 are scalar parameters. There is a summary of current PSO information at [1], [3], which includes information about parameter settings. In the PSO update equation (6), the velocity v_{i+1} specifies the search direction and step size for the particle. The velocity can be interpreted as a weighted combination of a point-difference estimate of a coarse regional gradient $(g^* - x_i)$ and a point-difference estimate of a generally finer regional gradient $(p^* - x_i)$.

FIPS: In this paper, we investigate gradient estimation for a recent variant of PSO called the fully-informed particle swarm (FIPS) [10]. FIPS is structured in a way that makes it more amenable to replacing the point-difference gradient estimates with other gradient estimates, and has been shown to outperform the canonical PSO [10].

The FIPS algorithm defines a fixed neighborhood $\mathcal{N}$ of N nearby particles for each particle. Then the location x_i of a particle at the i^{th} iteration evolves according to the FIPS equations:

$$\begin{aligned} x_{i+1} &= x_i + v_i, \\ v_i &= \chi(v_{i-1} + \varphi(P_i - x_i)), \quad (7) \\ P_i &= \frac{\sum_{k \in \mathcal{N}} \varphi_k p_k^*}{\sum_{k \in \mathcal{N}} \varphi_k}, \quad (8) \end{aligned}$$

where p_k^* is the personal best location of the kth particle at the ith iteration, $\chi = .7298$ and $\Phi = 4.1$. The φ_k are drawn independently and identically from the uniform distribution $U[0, \frac{\Phi}{N}]$, and φ is drawn from $U[0, \Phi]$. These parameter values, suggested variants and a discussion of different neighborhood definitions are given in [10]. In this paper, the connected ring neighborhood is used, which gives each operating point two neighbors and was shown to work well [10].

The equations for the FIPS update can be interpreted as an update with a point-difference regional gradient estimate

$$\delta_i = P_i - x_i, \quad (9)$$

where P_i (defined in (8)) is a randomly-weighted average of the personal best values in the particle x's neighborhood.

MEGA: The MEGA algorithm was motivated by the idea of gradient descent at multiple resolutions [8], [7]. In MEGA a population of *active points* evolves, where an active point is a point that is used to calculate the next step of the search. The population is initialized at the start of the optimization with $(D+1)^2$ randomly drawn points x_i for $i = 1$ to $(D+1)^2$, which are evaluated to form the initial active population $\{x_i, f(x_i)\}$. Each time a new operating point $(x_{new}, f(x_{new}))$ is evaluated, the current worst point is removed, and then the newly-evaluated point is added to the active population. This replacement forces the population to evolve.

At each iteration a set of line-search steps creates new points. First, fine resolution steps are taken by clustering the active points into $D+1$ clusters of spatially-adjacent points, and taking a step from the cluster center in the direction of the regional gradient associated with that cluster. Second, the $D+1$ new points are used to fit a coarse-resolution regional gradient, and an additional step is taken from the mean of the new points in the direction of that regional gradient.

The regional gradients described in the last paragraph are obtained by fitting a hyperplane to the $\{x_j, f(x_j)\}$ pairs in a given cluster. The squared error is minimized such that the hyperplane has slope coefficients ρ^* such that

$$[\rho^*, \rho_0^*] = \underset{\rho, \rho_0}{\operatorname{argmin}} \sum_{j \in \mathcal{J}_n} (f(x_j) - \rho^T x_j - \rho_0)^2, \quad (10)$$

where ρ_0^* is an offset that does not form part of the gradient direction.

We interpret ρ^* as an estimate of the unknown regional gradient β^* for the region spanned by the clusters' points.

The closed-form solution for ρ^* is

$$\rho^* = (X^T X)^{-1} X^T y, \quad (11)$$

where X is a matrix whose jth column is x_j, and y is a vector whose jth element is $f(x_j)$. In practice, we use the Moore-Penrose pseudoinverse to calculate ρ^* such that any singular values of X below a small fixed tolerance value are disregarded. The pseudoinverse provides the unique solution to (10) with minimum $\rho^T \rho$. In this way the pseudoinverse provides a low estimation variance estimate even when fewer than D points are used to fit the hyperplane [6].

III. Gradient Estimation Performance

The regional gradient is a way to capture functional trends over a region, and one expects that the regional gradient can be more helpful than the analytic gradient when searching for a global optima. To test this hypothesis, we begin with a study of the effectiveness of different regional gradient estimation techniques abstracted from any specific global optimization algorithm.

A. Experimental Details

We compare the angle θ between each normalized gradient or regional gradient estimate a calculated from a random draw of points to the true direction b to the global optima. That is,

$$\cos(\theta) = \frac{a^T b}{\|a\| \|b\|}.$$

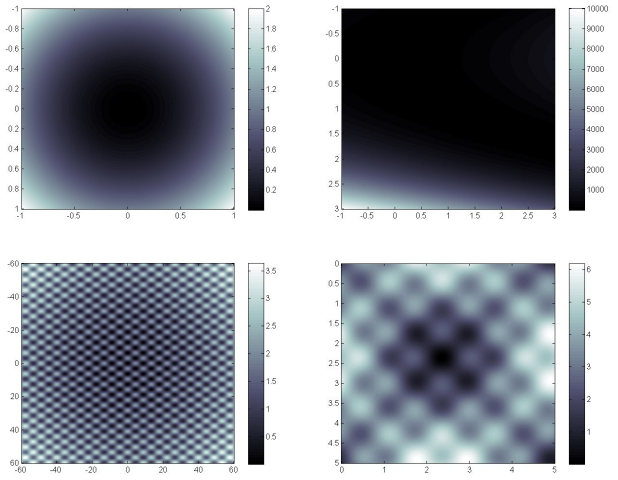

Fig. 2. Illustrations of the four test functions in two-dimensions. Clockwise from upper left: quadratic, Rosenbrock, Griewank, sinusoidal.

The experiment was run on four different test functions for which the true analytical gradient and the global minimum are known and can thus be used to evaluate the estimated gradients. These test functions were also chosen for their varying degrees of difficulty for a global search to solve. At the same time each function has some structure to it because these are the type of problems for which we expect a regional gradient based search to be powerful. For each run, $(D+1)$ points $x_1, x_2, \ldots, x_{D+1} \in S$ were randomly drawn, and their mean $\bar{x}$ was calculated. The $D+1$ points were drawn uniformly across the hypercube defined by the domain of the test function. However, the maximum difference between the points in any single dimension was limited to a fraction of the entire domain, and any points exceeding that distance from the original set were thrown out. This fraction limits the region for the gradient estimate, and was set to 0.9 for the results in Tables I and II. The analytical gradient is taken to be the true gradient at $f(\bar{x})$. The regional gradient is calculated over a region defined by a hypersphere with $\bar{x}$ as its center and $\max(\|x_i - \bar{x}\|)$ as its radius, and the formula (2) is approximated by a least-squares hyperplane fit to 1000 points drawn uniformly over the region.

A pseudoinverse regional gradient estimate ρ^* is calculated as in equation (11) for each draw of $D+1$ points. A second regional gradient estimate is formed for each draw of $D+1$ points by the coefficients of a regularized least squares fit hyperplane coefficients. Regularized least-squares linear regression penalizes the slope of the fitted hyperplane, in order to reduce estimation variance. We regularized using ridge regression [6], [9], such that the estimated coefficients ϱ^* are defined by

$$[\varrho^*, \varrho_0^*] = \underset{\varrho, \varrho_0}{\operatorname{argmin}} \left\{ \sum_{i \in \mathcal{J}_n} \left(f(x_i) - \varrho^T x_i - \varrho_0 \right)^2 + \lambda \varrho^T \varrho \right\}, \tag{12}$$

where λ is a parameter controlling the regularization. Based on preliminary experiments, a fixed value of $\lambda = .01$ was used throughout this work. A third estimate of the regional gradient is a point-difference (PD) which was designed to emulate the PD estimate found in the FIPS algorithm (see equation (9)), such that

$$\begin{aligned} PD &= \bar{x} - x_{i^*}, \\ i^* &= \underset{i}{\operatorname{argmin}}\, f(x_i). \end{aligned} \tag{13}$$

TABLE I

ANGLE θ VALUES IN RADIANS FOR SINGLE MINIMA TEST FUNCTIONS

	2D Quadratic	2D Rosenbrock	10D Rosenbrock
analytic gradient	0.000	0.827	0.562
regional gradient	0.035	0.831	0.852
pseudoinverse	0.646	1.096	1.133
ridge	0.541	1.054	0.879
point-difference	0.673	1.198	1.039

B. Experimental Results Comparing Estimated Gradients

Results of the average values of the angle θ to the optimal direction are given in Tables I, II, and III. The statistical significance of these results was analyzed using the Wilcoxian signed rank test, and all cases were found to be statistically significantly different at a significance level of 0.05. Note that the value of θ can range from $[0, \pi]$, and the expected value of θ for a randomly directed vector is $\pi/2$, or 1.57. Thus even for the sinusoidal and Griewank functions with many multiple minima, all of the search directions were better on average than random.

Table I gives the results for two monotonic functions. Even for monotonic functions, the analytic gradient may be misleading if the curvature varies over the range of the function. For example, in Rosenbrock's function (Figure 2) the analytic gradient only correlates loosely with the true direction to the optima. Table I shows that in these convex function cases, the regional gradient is a reasonable approximation of the analytical gradient, and the point-difference and two hyperplane fits provide similar approximations.

One expects the regional gradient to be the most useful when the test function has a strong global trend with weak local fluctuations, such as the Griewank function (Figure 2). We expect regional information to be less useful when the local trends are are just as strong as the global trends, such as in the sinusoidal function (shown in Figure 1.) However, even in such cases, we believe that the regional gradient will provide more useful information than the analytic gradient, which is may to point one toward a local minima. In fact, this hypothesis is confirmed by the results in Table II, which shows the regional gradient to be an improved indicator of the optimal direction compared to the analytic gradient for the sinusoidal and Griewank functions. The improvement is larger for the Griewank function, as expected due to its strong functional trend.

The regional gradient estimates perform similarly to each other, but relatively poorly, on the sinusoidal function. On the 2D Griewank function the point-difference is significantly worse than the least-squares hyperplane estimates. For the

TABLE II

ANGLE θ VALUES IN RADIANS FOR MULTI-MINIMA TEST FUNCTIONS

	2D Sinusoidal	2D Griewank	10D Griewank
analytic gradient	0.294	1.339	0.094
regional gradient	0.211	0.014	0.034
pseudoinverse	1.078	0.373	0.900
ridge	1.038	0.333	0.573
point-difference	1.180	0.608	0.957

TABLE III

ANGLE θ VALUES IN RADIANS FOR DIFFERENT RESOLUTIONS IN THE 2D GRIEWANK TEST FUNCTION

	Resolution as % of domain		
	0.2	0.5	0.9
analytic gradient	1.298	1.332	1.339
regional gradient	0.008	0.022	0.014
pseudoinverse	0.289	0.483	0.373
ridge	0.277	0.395	0.333
point-difference	0.639	0.642	0.608

Griewank function calculated over a ten-dimensional hypercube, all the regional gradient estimates are again relatively poor. Throughout, the ridge regression estimate is shown to be slightly more effective than the pseudoinverse or point-difference in two-dimensions, and much more effective for the ten-dimensional case. Estimation error is either due to estimation variance (which describes how variable the estimate is when the sample points change) or estimation bias (which describes how wrong the average estimate is when the average is over many random draws of sample points) [6]. Given that there are only $D+1$ points in D dimensions used for the gradient approximations, estimation variance will generally be a greater problem than estimation bias, and the ridge regression's estimation variance reduction is what causes it to be the best estimator. Similarly, the pseudoinverse consistently performs better than the point-difference. This difference is because the point-difference has the highest estimation-variance due to its use of the minimum $f(x_i)$, which can vary greatly for different draws of $D+1$ points.

Next, we present data on how the size of the region used to estimate the gradient impacts the metric θ. Table III compares θ values for the 2D Griewank function where the resolution of the gradient was limited by setting the maximum distance between any two points in the test set was limited to 0.2, 0.5, and 0.9 of the function domain for each dimension. Note that the analytic gradient does not depend on the region size, and so the analytic gradient row gives an indication of the variance of the results. The results show that, for the Griewank, changing the resolution has a large impact on the power of the high-fidelity regional gradient (β^*). The impact on the three $D+1$ point regional gradient estimates is smaller, perhaps because their fidelity to the true regional gradient is limited already. In the context of a practical global optimization algorithm, the regional size used to estimate gradients will vary, either due to chance or by design. Since some region sizes may result in more useful search directions, a well-designed region size variation may be quite advantageous.

IV. REGIONAL GRADIENTS IN THE CONTEXT OF EVOLUTIONARY ALGORITHMS

Gradient estimation is only one part of the overall behavior of a global optimization algorithm. The algorithms considered here are population-based, which means that the methods for choosing and evolving the population of operating points may have a profound impact on the overall performance of the search. The population distribution directly impacts the regions over which gradients are estimated, thereby dictating how the current set of operating points is updated. At the same time the population must maintain a record of the most promising points, and allow some hill-climbing in the search.

In this section a controlled comparison is made between different gradient estimates in the context of FIPS and MEGA. The results indicate how important the quality of the gradient estimate is to the final performance of the algorithms. In these tests each algorithm was allowed to run for 10,000 function evaluations so that different configurations and algorithms could be compared.

A. Regional Gradients and FIPS

The FIPS update equations result in a velocity (v_i), which includes both the direction and the step size for a given step. To control the FIPS step-size across the different gradient estimates, the original velocity magnitude is used for all the gradient estimates. The original FIPS point difference estimate can be viewed as a regional gradient for a region centered on the midpoint between x_i and P from equation (8), so the other gradient estimates are calculated for the same region. The analytical gradient is also evaluated at that midpoint.

We expect the original FIPS point-difference to perform well, but hypothesize an improvement when better gradient estimates are used. The results in Figure 3 show that with the exception of the Rosenbrock function and some dimensions of the sinusoidal function, the analytical and regional gradient outperform all other options. Using a random direction for the gradient performs the worst in all cases, and gives an indication of the importance of gradient information to the population-dynamics. We hypothesize that well-designed population evolution and parameter choices may compensate for poor gradient information. Since FIPS was designed to work with a point-difference gradient, the step-size and parameter settings may be suboptimal for the other gradient estimates and actually result in a misdirected search.

While the experiments of Section III showed that the ridge regression could produce better gradient estimates than the pseudoinverse or point difference, when coupled with the FIPS algorithm the ridge estimate ϱ lags slightly behind the pseudoinverse estimate ρ. Possible reasons are that the estimation bias of the ridge estimate ends up slowing convergence, or that the reduced estimation variance might actually hurt the progress of finding the optima by reducing

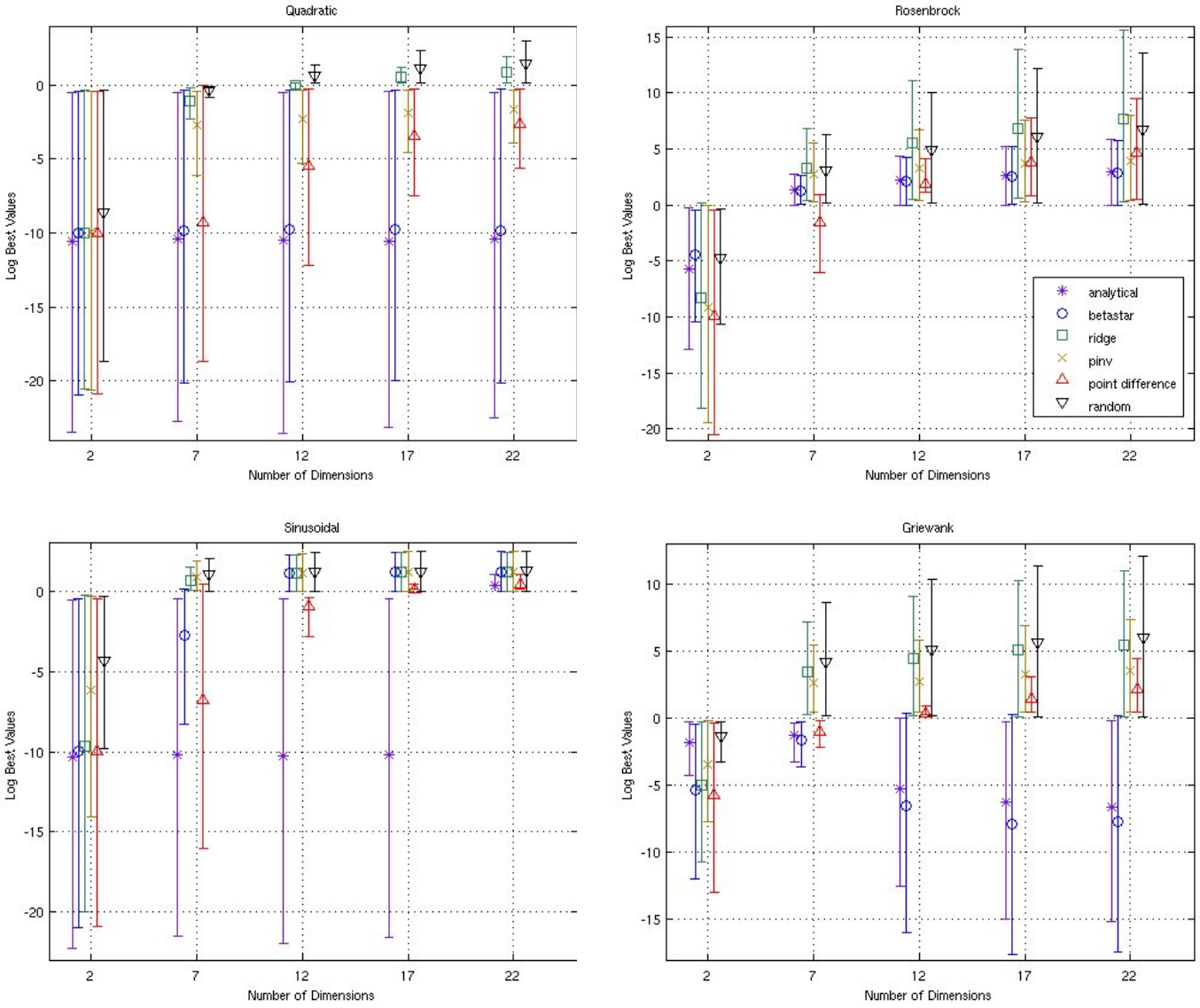

Fig. 3. The FIPS results for each of the four test functions. Clockwise from upper left: quadratic, Rosenbrock, Griewank, and sinusoidal. The figures show the average minimum value found for 100 algorithm runs, on a log scale. The error bars mark the 25th and 75th percentile of the minimum values. The different gradient calculation techniques are the analytical gradient, the estimated regional gradient from equation (2) (betastar), the ridge regression gradient estimate from equation (12) (ridge), the pseudoinverse gradient estimate from equation (11) (pinv), the FIPS gradient estimate from equation (13) (point difference), and a randomly generated search direction (random).

the randomness of the search. A better scheme might be to adapt the regularization parameter λ based on the step-size, the size of the region spanned by the sample points, or on the size of the errors in the least-squares hyperplane fit. In this case the extra parameter becomes a liability and ridge regression becomes less attractive. Optimizing the FIPS parameters for the pseudoinverse or ridge estimate would likely improve performance.

Lastly, a difference between the FIPS point-difference and the other estimates is the randomness. It is thought that the PSO algorithms work by driving particles toward, but not directly to, the nearby minimum. This process of overshoot ensures that the algorithm does not get stuck in local minima. Since earlier analysis showed that the point difference estimate was not as good as other techniques for pointing directly to the global minima, the randomized misdirection appears to help FIPS avoid local minima. In fact, the FIPS point-difference shows a generally larger improvement over the pseudoinverse and ridge regression estimates for the two functions with multiple minima than it does the two convex functions.

B. Regional Gradients and MEGA

The MEGA algorithm is designed to explicitly use regional gradient estimates. Therefore we expect that the higher fidelity gradient estimates will improve the overall performance. For the MEGA experiments, the point-difference estimate is given in (13). The step size is the maximum size

of the cluster in any dimension ($\max_d(x_{\max}[d] - x_{\min}[d])$), and the gradient estimation techniques are used only to determine the search direction.

The results from the MEGA algorithm are shown in Figure 4. For all the functions but Griewank, the analytical gradient outperforms the other gradients, followed by the regional gradient, and then the pseudoinverse gradient. In some of the dimensions of the sinusoidal test function the regional gradient estimate performs poorly, suggesting that this estimate offers a misleading search direction. Similar behavior was seen in the FIPS results. One reason for this result may be that the regional gradient is being estimated at the wrong resolutions.

In the Griewank test function - the one most suited to the use of a regional gradient - the regional gradient and the pseudoinverse estimate perform the best. For the MEGA results, the random descent direction and point-difference estimated gradient do relatively poorly. As with FIPS, the ridge regression does not do as well as the pseudoinverse; we have discussed possible reasons for this in Section IV-A.

V. Conclusions and Further Discussion

The primary question investigated in this paper is how regional gradient estimates affect the performance of global optimization algorithms. The results presented in Section III show that a regional gradient can serve as a pointer to the global minimum in a multimodal function. In Section IV it is shown that gradient-based steps do improve the performance of two population-based search algorithms over random steps. The functional trend is strongest in the Griewank function, and for both algorithms the regional gradient proves advantageous over the analytical gradient for this function.

The random search direction experiments in Section III-B indicated how useful using regional gradient approximations is. There is generally a large difference between the analytic gradient results and the random results, showing that the search direction is important. However, in Section IV we were surprised at how much overlap there is between the minimum values obtained with the random search direction and the gradient-approximation search directions. This may imply that the quality of the gradient estimates is too poor to offer a large advantage. It may also be that the population evolution alone can move the search close to the goal, or that randomness in the search direction is useful in its own right for escaping local minima.

The results in Figures 2 and 3 show algorithm performance and allow us to consider the theoretical performance of the FIPS and MEGA algorithms: given a perfect analytic gradient, how well can these population-evolution architectures perform? The FIPS algorithm does well with the analytic gradient for the quadratic and sinusoidal test functions, but performs better with the regional gradients for the Rosenbrock and Griewank functions. The results for the MEGA algorithm are similar, but the analytic gradient also helps in the Rosenbrock case. We control for the gradient direction, so the difference between the two algorithms is the population evolution mechanics.

This paper shows that regional gradient information is useful for optimization, so the population dynamics of an algorithm must support the development of that information. The population dynamics determine which points are available for estimating gradients, and those points determine the regional span of the gradient. More important for MEGA than FIPS is that $D + 1$ linearly independent points are available for each estimate. Neither MEGA nor FIPS exert much explicit control over the evolution of the regional sizes or maintaining populations that form a linearly independent span of the space. In the trust region methods presented by Powell [13] some of the optimization steps are designed to minimize the function and some are designed to ensure that the simplex used for calculated gradients is not degenerative. Algorithms such as FIPS and MEGA might similarly benefit from occasional steps to ensure that the population is well-distributed in the search space.

One challenge for global optimization algorithms is the definition of the size of each search step. Current methods for determining step size vary, often including a random element, or adapting according to whether the previous step returned a promising solution. In both FIPS and MEGA the step-size is a function of the size of the region represented by the regional gradient estimate, such that the step-size makes appropriate use of the regional gradient estimate. Further advances in global optimization may be possible by considering the step size and region size to be linked.

Gradient-based optimization algorithms can be shown to converge for functions that only have one minima [11], but proving convergence for global optimization is difficult. One approach may be to consider functions that have smooth regional gradients at some resolution, and prove convergence at that resolution.

This paper considered at the impact of gradient estimation in directing searches for global optima. A regional gradient was introduced as a tool for analyzing and comparing different types of gradient estimates. A series of results were presented for performance on a variety of test functions that show that the quality of a gradient estimate has an effect on the performance of each of two population-based search methods. This suggests that one avenue for improving existing global search algorithms is to examine the way a gradient direction is estimated. Further suggestions for using the regional gradient as a tool in global optimization are presented in the last section of this paper.

References

[1] http://www.swarmintelligence.org/tutorials.php.

[2] C. Audet and J.E. Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2003.

[3] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. on Evolutionary Computation*, 6(1):58–72, 2002.

[4] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Sixth International Symposium on Micro Machine and Human Science*, pages 39–43. IEEE, October 1995.

[5] F. Glover and M. Laguna. *Handbook of Applied Optimization*, chapter 3.6.7: Tabu Search, pages 194–208. Oxford University Press, New York, NY, 2002.

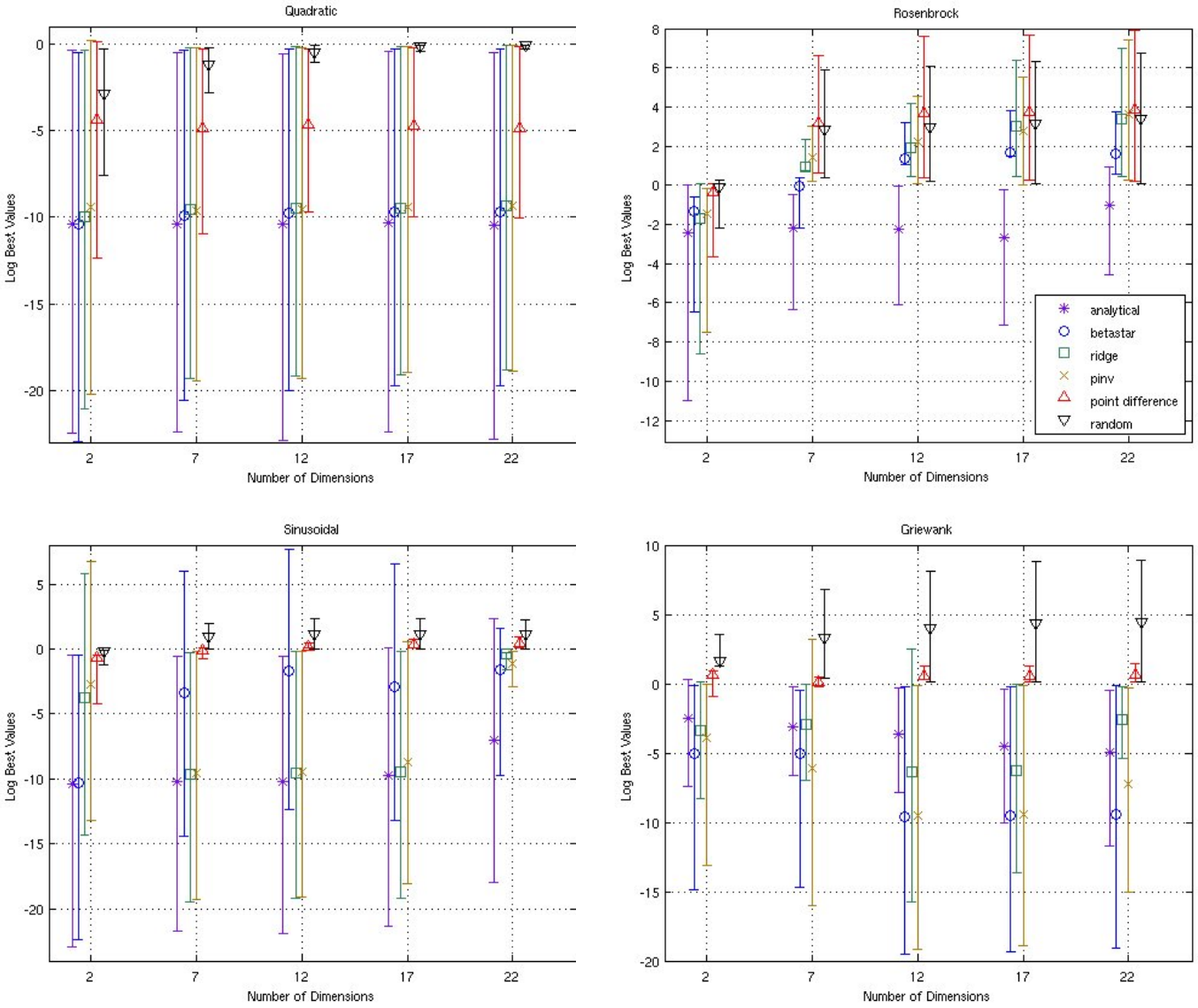

Fig. 4. The MEGA results for each of the four test functions. Clockwise from upper left: quadratic, Rosenbrock, Griewank, and sinusoidal. The figures show the average minimum value found for 100 algorithm runs, on a log scale. The error bars mark the 25th and 75th percentile of the minimum values. The different gradient calculation techniques are the analytical gradient, the estimated regional gradient from equation (2) (betastar), the ridge regression gradient estimate from equation (12) (ridge), the pseudoinverse gradient estimate from equation (11) (pinv), the point difference gradient estimate from equation (13) (point difference), and a randomly generated search direction (random).

[6] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2001.

[7] M. Hazen. *Search Strategies for Global Optimization*. PhD thesis, Univ. of Washington, 2008.

[8] M. Hazen and M.R. Gupta. A multiresolutional estimated gradient architecture for global optimization. pages 3013–3020. IEEE Congress on Evolutionary Computation, July 2006.

[9] A. E. Hoerl and R. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[10] R. Mendes, J. Kennedy, and J. Neves. The fully informed particle swarm: simpler, maybe better. *IEEE Trans. on Evolutionary Computation*, 8(3):204–210, 2004.

[11] D.A. Pierre. *Optimization Theory With Applications*. Dover Publications, Inc., New York, NY, 1986.

[12] O. Polgar, M. Fried, and I. Barsony. A combined topographical search strategy with ellipsometric application. *Journal of Global Optimization*, 19:383–401, 2001.

[13] M.J.D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, 7:287–336, 1998.

[14] J.C. Spall. *Introduction to Stochastic Search and Optimization*. Wiley-Interscience, Hoboken, NJ, 2003.

[15] J.C. Spall, S.D. Hill, and D.R. Stark. *Theoretical Framework for Comparing Several Stochastic Optimization Approaches*, chapter 3. Springer, Berlin, 2006.

[16] K.F.C. Yiu, Y. Liu, and K.L. Teo. A hybrid descent method for global optimization. *Journal of Global Optimization*, 28:229–238, 2004.

[17] Z.B. Zabinsky. *Stochastic Adaptive Search for Global Optimization*. Kluwer Academic Publishers, Norwell, MA, 2003.