Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

OPTIMIZATION VIA GRADIENT ORIENTED POLAR RANDOM SEARCH

Haobin Li
Loo Hay Lee
Ek Peng Chew

National University of Singapore
Department of Industrial and Systems Engineering
1 Engineering Drive 2, SINGAPORE 117576

ABSTRACT

Search algorithms are often used for optimization problems where its mathematical formulation is difficult to be analyzed, e.g., simulation optimization. In literature, search algorithms are either driven by gradient or based on random sampling within specified neighborhood, but both methods have limitation as gradient search can be easily trapped at a local optimum and random sampling loses efficiency by not utilizing local information such as gradient direction that might be available. A combination of the two is believed to overcome both disadvantages. However, the main difficulty is how to incorporate and control randomness in a direction instead of a point. Thus, this paper makes use of a polar coordinate representation in any high dimension to randomly generate directions where the concentration can be explicitly controlled, based on which a brand new Gradient Oriented Polar Random Search (GO-POLARS) is designed and proved to satisfy the conditions for strong local convergence.

1 INTRODUCTION

For many optimization problems aiming to analyze a complex system, quite often the mathematical programming methods cannot be applied, either because the loss function is too complicated to be analyzed or the problem is formulated by a simulation model where the close form of loss function does not occur. In such cases, adaptive search algorithms becomes better choice as only local information is required which can be easily obtained from loss functions or evaluated by simulation models.

One category of well-known search algorithms is driven by gradient information. The oldest method is the steepest descent approach (Debye 1909) which assumes that the gradient is known at each search iterate, so the search always moves towards its opposite direction with a stepsize proportional to its magnitude and a given gain sequence. In the case where the gradient cannot be directly measured, a finite-difference stochastic approximation method (FDSA) is used for estimating the gradient information (Kiefer & Wolfowitz 1952, Blum 1954b). Later, as FDSA is costly in term of number of evaluations when dimension p is high, Spall (1998 & 2003) propose the simultaneous perturbation stochastic approximation (SPSA) that increases the estimation efficiency.

Although it has been shown that under certain conditions, a gradient-driven algorithm converges to a local optimal point almost surely (Spall 2003), the global convergence is difficult to be ensured as the greedy use of gradient information sacrifieces the exploration on the whole solution space. It can be argued that, in FDSA and SPSA the gradient is approximated with certain noise, which unintensionally increases the variety of the search directions. However, since the noise cannot be controlled explicitly, it is difficult to balance search exploration at a desired level.

Another category, often referred as metaheuristics local search, mainly depend on stochastic sampling within carefully designed neighborhood structures. For example, the simulated annealing (SAN)

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

algorithm (Kirkpatrick et al. 1983), Tabu search (Glover 1990), genetic algorithms, the nested partitions method (Shi and Ólafsson 2000), and COMPASS (Hong and Nelson 2006). Compared to gradient-driven algorithms, the neighborhood structure often ensures a better exploration on the search space. But on the other hand, there is certainly some room for improvement in term of search efficiency, as the gradient information which can probably be measured or approximated is not utilized at all.

It is obvious that if the search direction in a gradient-based algorithm can be randomized with desired variation, or the stochstic sampling in a metaheuristic can be oriented by the gradient information, we can design a new seach algorithm that believes to have better performance than both. Although Pogu & Souza de Cursi (1994) proposed a method for random pertubation of the gradient, we noticed that the perturbation within a region surrounding the targeted point cannot control the search direction explicity. For example, when the stepsize is sufficiently large or small, the same amount of perturbation may inccur much difference in the search direction.

For this reason, in Section 2, it is the first time we propose a generalized polar coordinate framework in any p dimension, that provides an explicit way for perturbing a direction. Two random distributions are defined based on it, namely the polar uniform distribution and the polar normal distribution.

Using the proposed framework, in Section 3 we propose a new search algorithm called the Gradient-Oriented Polar Random Search (GO-POLARS). Subsequently, the local convergence property and numerical examples are to be discussed.

2 A POLAR FRAMEWORK

2.1 Generalized Polar Coordinates

For a p-dimensional optimization problem, a Cartesian coordinate system is usually adopted to uniquely identify a solution point in the domain space. In Cartesian system, all coordinates are orthogonal to each other, and a point is denoted by $\mathbf{x} = \left[x_1, \ldots, x_p \right]$ such that x_i refers to its projected position on the i th coordinate.

Cartesian system is a natural way to represent solutions of optimization problems, because in many cases x_i directly refers a decision parameter. However, we observe that for many adaptive or local search algorithms Cartesian representation may not be the best choice as the search is driven by two key factors, namely the direction and the distance. But neither of them is explicitly expressed in a Cartesian system. Thus, we may think of an alternative way to denote the solution, such as polar coordinates.

It should be well known that, a polar coordinate system can be defined on a two-dimensional space in which every point is denoted by its angle with respect to an axis and distance to the origin (Weisstein 2009). Besides, the similar idea can be brought into a three-dimensional case so as to form a sytem called spherical coordinates (Weisstein 2005) or spherical polar coordinates (Walton 1963, Arfken 1985). However, higher dimension cases are seldom discussed in literature. So, as following we propose a generized polar coordinate representation that can be adopted for any high dimensional cases.

Definition 1 *In a p-dimensional polar coordinate system, a point is denoted by $\left[r, \boldsymbol{\theta} \right]$, in which $r \in \left[0, \infty \right)$ and $\boldsymbol{\theta} \in \left[0, 2\pi \right) \times \left[0, \pi \right]^{p-2}$, if its euclidean distance from the origin is r (radial coordinate) and $\boldsymbol{\theta}$ (angular coordinate) refers its direction in the space in the sense that θ_i denotes its angle with respect to the positive direction of the $i+1^{th}$ axis towards the hyperplane spanned by the first i axes.*

To be more specific, the relationship between the Cartesian and polar coordinates in p-dimensional space can be described by Equation (1), (2) and (3).

$$r = \sqrt{\sum_{i=1}^{p} x_i^{\,2}}\ , \tag{1}$$

$$x_1 = r\prod_{j=1}^{p-1}\sin\theta_j\ , \tag{2}$$

$$x_i = r\cos\theta_{i-1}\prod_{j=i}^{p-1}\sin\theta_j \quad \text{for} \quad 2 \le i \le p\,. \tag{3}$$

An illustration for the generalized polar coordinate representation in the two and three dimensional space can be found in Figure 1. We note that the mapping from polar to the Cartesian coordinate system is almost one-to-one, and the degree of freedom remains p in both representation.

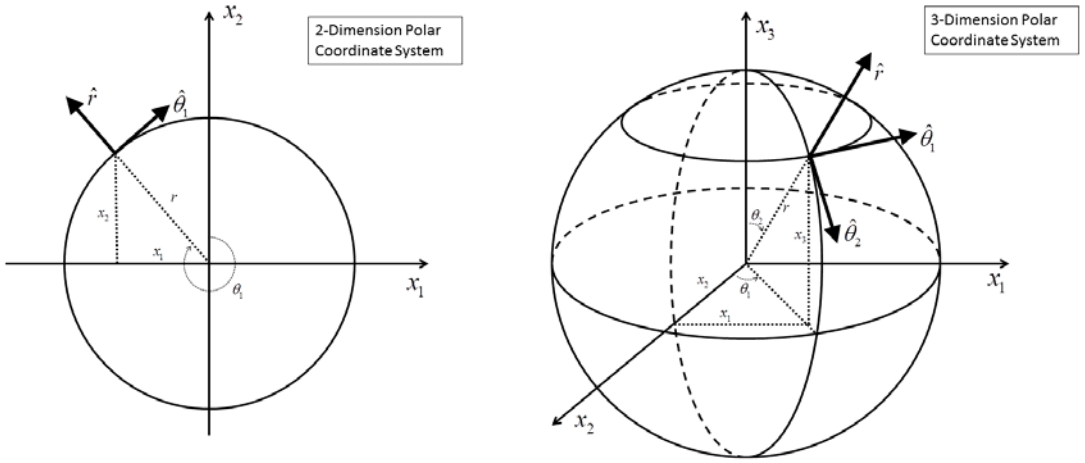

Figure 1: An illustration of generalized polar coordinates (in 2D & 3D).

2.2 Polar Uniform Sampling

With generalized polar coordinates we are able to denote a point in terms of the direction and distance referring to a given position, which provides an advantage for algorithms to explicitly control their search process. But as mentioned in Section 1 where the variation is involved in sampling a direction, random distribution need to be defined before we move to introduce the algorithm. First of all, we look at a uniform case.

In a p-dimensional space, in order to uniformly sample a direction, we simply consider a hyperball with radius r around the origin. For all the points spread in the outermost layer, each of them should have equal opportunity to be sampled as the direction. From mathematical point of view, let $f\left(r,\boldsymbol{\theta}\right)$ be the probability density function, then within an infinitesimal space around the point $\left[r,\boldsymbol{\theta}\right]$, the probability for points to be sampled is

$$f\left(r,\boldsymbol{\theta}\right)\cdot\partial\left(r,\theta_1,\ldots,\theta_{p-1}\right).$$

By consensus of uniformity, this probability should be proportional to the volume of the infinitesimal space $\partial V = \partial\left(x_1,\ldots,x_p\right)$, meaning there exists a function $c\left(r\right)\ge 0$ such that

$$\frac{f(r,\boldsymbol{\theta})\cdot\partial(r,\theta_1,\ldots,\theta_{p-1})}{\partial(x_1,\ldots,x_p)}=c(r).$$

Further notice that the Jacobian determinant (Kaplan 1991) of the mapping from polar to Cartesian coordinate system is expressed as

$$J_{r,\boldsymbol{\theta}}=\frac{\partial(x_1,\ldots,x_p)}{\partial(r,\theta_1,\ldots,\theta_{p-1})}=r^{p-1}\prod_{j=1}^{p-1}\sin^{j-1}\theta_j .$$

So we derive the probability density function for a polar uniform distribution as in (4) and thus have Definition 2. Note that $c(r)$ depends only on r and has to ensure the integral of $f(r,\boldsymbol{\theta})$ on the domain equals to 1.

$$f(r,\boldsymbol{\theta})=c(r)\cdot r^{p-1}\prod_{j=1}^{p-1}\sin^{j-1}\theta_j . \tag{4}$$

Definition 2 *A random point $[r,\boldsymbol{\theta}]$ is said to be from a p-dimensional polar uniform distribution, denoted as $\mathrm{U}^{p}_{\mathrm{polar}}$, if its probability density function is given as in (4).*

When enforce $r=1$, i.e. let $c(r)=0$ if $r\neq 1$, the angular coordinate $\boldsymbol{\theta}$ can be sampled uniformly in the sense that $[1,\boldsymbol{\theta}]\sim \mathrm{U}^{p}_{\mathrm{polar}}$, which is illustrated by Figure 2. Since r is fixed and θ_j is independent from each other, we can decompose the probability density function for each j as

$$f_j(\theta_j)=c_j\sin^{j-1}\theta_j,\quad \text{for } j=1,\ldots,p-1 \tag{5}$$

where $c_1=\dfrac{1}{2\pi}$ and $c_j=\int_0^{\pi}\sin^{j-1}\theta d\theta$ for $j\geq 2$. As there is no close form for c_j, one of the method is to apply numerical approaches, such as acceptance-rejection method or Alias method (Vose 1999, Schwarz 2011) after discretization into small intervals, so that the constant term can be ignored.

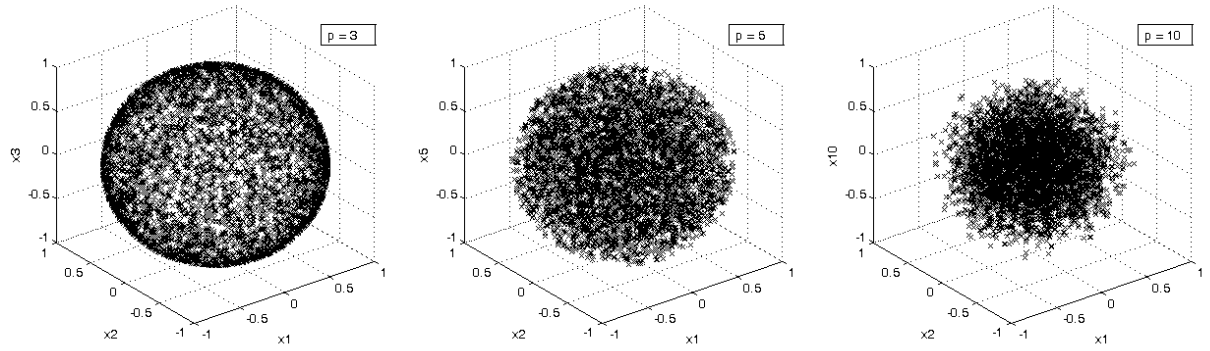

Figure 2: Polar uniform distribution with $r=1$ and $p=3,5,10$.

In practice, some good properties can be observed. Since (5) is independent with dimension p, a point $[1,\boldsymbol{\theta}]\sim \mathrm{U}^{p}_{\mathrm{polar}}$ can be easily extended to $\mathrm{U}^{p+1}_{\mathrm{polar}}$ by adding element θ_p sampled from distribution with density $f_p(\theta_p)=c_p\sin^{p-1}\theta_p$.

Moreover, considering (2) and (3), we notice that the newly added θ_p does not affect the relative values of previous x_i's since all of them are simply scaled by $\sin\theta_p$, whereas the density in (5) only ensures that the new comer x_{p+1} plays harmoniously with the early ones by maintaining the uniformity into the higher dimension. This property is important when we need to add controlled bias into the distribution, which is to be discussed later in 2.3.

Alternatively, we note that a multivariate normal distribution $\mathrm{N}\left(\mathbf{0}, I\sigma^2\right)$ with any σ is a special case for polar uniform distribution when it is converted into the generalized polar coordinates (The proof is omitted due to the page limit). The sampled points can be easily standardized into unit vector by letting $r=1$ before it can be applied in the steps following to be discussed. With this result, we can certainly simplify the random generator for polar uniform distribution.

2.3 Concentrated Polar Distribution

Firstly, we consider a simple case, where we want the sampled direction to be concentrated around a given direction $\tilde{\mathbf{d}}$ that coincides with the positive direction of the p th axis, i.e. $\tilde{\mathbf{d}}=\mathbf{e}_p$. It means that, under the Cartesian representation only x_p has a priority to choose larger value. Thus, using the property discussed in the later part of 2.2, we may have a point $\left[1,\boldsymbol{\theta}\right] \sim \mathrm{U}_{\mathrm{polar}}^{p-1}$, and extend it to p-dimension by adding θ_{p-1} where the distribution can be adjusted from (5), so that x_p has higher chance to take large value without touching the ratio among the others. A typical way is to take the composite density with a truncated normal distribution with mean 0 and variance σ^2, i.e.,

$$f_{p-1}\left(\theta_{p-1}\right)=c'_{p-1}\sin^{p-2}\theta_{p-1}\cdot\phi_{0,\sigma^2}\left(\theta_{p-1}\right) \tag{6}$$

where

$$\phi_{0,\sigma^2}\left(\theta\right)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{\theta^2}{2\sigma^2}} \text{ for } 0\le\theta\le\pi, \quad \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(2\pi-\theta)^2}{2\sigma^2}} \text{ for } \pi<\theta\le 2\pi .$$

Note that $f_{p-1}\left(\theta_{p-1}\right)$ is defined on $\left[0,2\pi\right)$ for $p=2$, and $\left[0,\pi\right]$ for $p>2$. We refer it as the standard polar normal distribution as in Definition 3.

Definition 3 *A random point* $\left[r,\boldsymbol{\theta}\right]$ *is said to be from a* p*-dimensional standard polar normal distribution with variance* σ^2*, denoted as* $\mathrm{N}_{\mathrm{polar}}^{p}\left(\sigma^2\right)$*, if* $\left[r,\theta_1,\ldots,\theta_{p-2}\right] \sim \mathrm{U}_{\mathrm{polar}}^{p-1}$ *and* θ_{p-1} *distributed as in* (6)*.*

The procedure of generating $\left[1,\boldsymbol{\theta}\right] \sim \mathrm{N}_{\mathrm{polar}}^{p}\left(\sigma^2\right)$ is described by Algorithm 1. Set σ to different values, then we have the illustration of sampled points shown by Figure 3. It is clear that similar to normal distribution, small σ will have high density of samples around the given position $\mathbf{e}_p$.

An extreme case can be observed when $\sigma=0$ so that $\phi_{0,\sigma^2}\left(\theta_{p-1}\right)=0$ for all $\theta_{p-1}\neq 0$, thus $\theta_{p-1}=0$ with probability 1, meaning that all sampled direction coincide with $\mathbf{e}_p$ almost for sure. In another way, if $\sigma=\infty$, we have equal value of $\phi_{0,\sigma^2}\left(\theta_{p-1}\right)$ at all θ_{p-1}, thus the term cancelled out from (6). In that case, $\mathrm{N}_{\mathrm{polar}}^{p}\left(\infty\right)\equiv\mathrm{U}_{\mathrm{polar}}^{p}$.

Algorithm 1

Step 1: Let $j = 1$.

Step 2: If $j < p-1$, sample θ_j from its domain with density as in (5), then set $j = j+1$.

Step 3: If $j = p-1$, sample θ_{p-1} from its domain with proportional density as in (6), then stop and report $\boldsymbol{\theta} = \left[\theta_1, \ldots, \theta_{p-1}\right]$; otherwise proceed to Step 3.

Step 4: Go to Step 2.

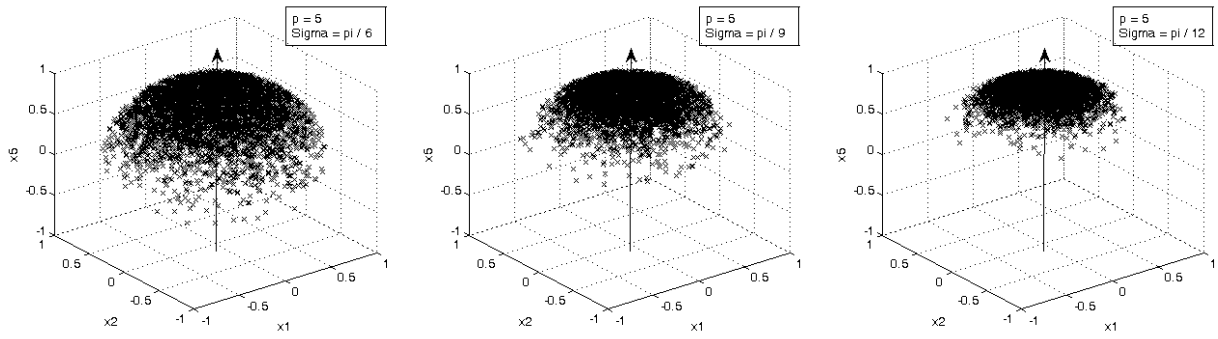

Figure 3: Standard polar normal distribution with $r = 1$, $p = 5$ and $\sigma = \pi/6, \pi/9, \pi/12$.

Denote $\mathbf{d} = \left(\left[1, \boldsymbol{\theta}\right]\right)_{\text{Cart}}$ as the Cartesian conversion of $\left[1, \boldsymbol{\theta}\right]$, we can analyze the expectation of $\mathbf{d}$ (detailed steps are omitted due to the page limit), so as to derive Theorem 1. Later we will use its corollary to prove the local convergence property in 3.2.

Theorem 1 *For a unit vector* $\mathbf{d} \sim \mathrm{N}_{\text{polar}}^{p}\left(\sigma^2\right)$ *given that* $\sigma < \infty$*, we can always finds a scalar* $\gamma_{p,\sigma} \in \left(0,1\right]$ *depends only on* p *and* σ *such that* $\mathrm{E}\left[\mathbf{d}\right] = \gamma_{p,\sigma} \cdot \mathbf{e}_p$.

For the case where the given $\tilde{\mathbf{d}}$ is an arbitrary unit vector, a linear transformation can be applied such that every point obtained by Algorithm 1 is reflected on a hyperline lies in the middle of $\tilde{\mathbf{d}}$ and $\mathbf{e}_p$. In a reverse manner, we have Definition 4 for the polar normal distribution. Figure 4 is an illustration.

Definition 4 *A random point* $\mathbf{d} = \left(\left[r, \boldsymbol{\theta}\right]\right)_{\text{Cart}}$ *is said to be from a* p*-dimensional polar normal distribution centered at unit vector* $\tilde{\mathbf{d}}$ *and variance* σ^2*, denoted as* $\mathrm{N}_{\text{polar}}^{p}\left(\tilde{\mathbf{d}}, \sigma^2\right)$*, if* $\dfrac{2\mathbf{d} \cdot \mathbf{m}^T}{\left\|\mathbf{m}\right\|^2}\mathbf{m} - \mathbf{d} \sim \mathrm{N}_{\text{polar}}^{p}\left(\sigma^2\right)$ *where* $\mathbf{m} = \left(\tilde{\mathbf{d}} + \mathbf{e}_p\right)/2$.

Obviously, as the result of linear transformation, from Theorem 1 we have Corollary 1.

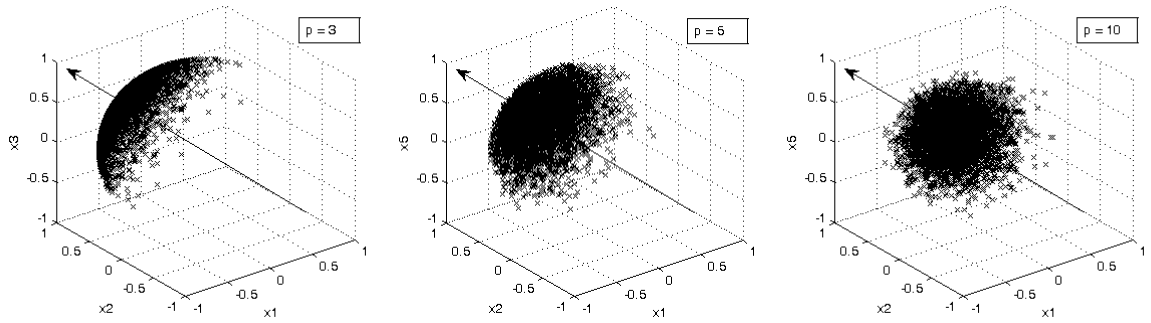

Figure 4: Polar normal distribution with $\tilde{\mathbf{d}} = \sum_{i=2}^{p} \mathbf{e}_i - \mathbf{e}_1$, $\sigma = \pi/9$ and $p = 3, 5, 10$.

Corollary 1 *For a unit vector* $\mathbf{d} \sim \mathrm{N}_{\mathrm{polar}}^{p}\left(\tilde{\mathbf{d}}, \sigma^2\right)$ *given that* $\sigma < \infty$*, we can always finds a scalar* $\gamma_{p,\sigma} \in (0,1]$ *depends only on* p *and* σ *such that* $\mathrm{E}[\mathbf{d}] = \gamma_{p,\sigma} \cdot \dfrac{\tilde{\mathbf{d}}}{\left\|\tilde{\mathbf{d}}\right\|}$.

3 SEARCH ALGORITHM

3.1 The Base Algorithm

The Gradient Oriented Polar Random Search (GO-POLARS) is designed in an adaptive manner. At each iteration, the optimum estimate moves to a random direction with a step size which is guided by the gradient. Specifically, let $\Theta \subseteq \mathbb{R}^p$ be the feasible region, the search algorithm can be described as following:

Algorithm 2

Step 1: (Initialization) Pick an initial guess $\hat{\mathbf{x}}_0 \in \Theta$, and set $k = 0$.

Step 2: Generate a direction $\mathbf{d}_k$ from $\mathrm{N}_{\mathrm{polar}}^{p}\left(\mathbf{g}\left(\hat{\mathbf{x}}_k\right), \sigma_k^2\right)$ in which $\mathbf{g}\left(\hat{\mathbf{x}}_k\right)$ is the gradient at $\hat{\mathbf{x}}_k$.

Step 3: Let $\hat{\mathbf{x}}_{\mathrm{new}} = \hat{\mathbf{x}}_k - b_k \left\|\mathbf{g}\left(\hat{\mathbf{x}}_k\right)\right\| \mathbf{d}_k$. If $\hat{\mathbf{x}}_{\mathrm{new}} \in \Theta$ and $L\left(\hat{\mathbf{x}}_{\mathrm{new}}\right) < L\left(\hat{\mathbf{x}}_k\right)$, set $\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{\mathrm{new}}$, otherwise $\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k$.

Step 4: Set $k = k + 1$. Go to *Step 2*.

Remark. The search procedure can be tuned by controlling the gain sequence b_k and the direction variation sequence σ_k. In 3.2, we will discuss conditions in terms of b_k and σ_k for the algorithm to converge to a local optimum.

3.2 Local Convergence Property

In literature, local convergence property of a stochastic algorithm is often shown by convergence theory of stochastic approximation (SA) (Spall 2003). And we notice that GO-POLARS shares some similarities with SA, such as both have estimates updated adaptively according to the gradient information with certain noise. So in this subsection, we try to relate GO-POLARS to SA and conclude the convergence conditions for the sequence b_k and σ_k.

We start with rewriting Step 3 in Algorithm 2 as an SA type, i.e., $\hat{\mathbf{x}}_{\text{new}} = \hat{\mathbf{x}}_k - a_k Y_k\left(\hat{\mathbf{x}}_k\right)$ where

$$a_k = \gamma_{p,\sigma} \cdot b_k, \tag{7}$$

$$\text{and } Y_k\left(\hat{\mathbf{x}}_k\right) = \frac{\left\|\mathbf{g}\left(\hat{\mathbf{x}}_k\right)\right\|}{\gamma_{p,\sigma}} \cdot \mathbf{d}_k. \tag{8}$$

Note that in a typical SA procedure, a_k is the gain sequence and $\mathbf{Y}_k\left(\hat{\mathbf{x}}_k\right) = \mathbf{g}_k\left(\hat{\mathbf{x}}_k\right) + \mathbf{e}_k\left(\hat{\mathbf{x}}_k\right)$ is an approximation of gradient $\mathbf{g}_k$ with error term $\mathbf{e}_k$. The "statistics" conditions for strong convergence can be drawn as in (9), (10), (11) and (12) (Blum 1954a,b; Nevel'son and Has'minskii 1973).

$$a_k > 0, a_k \to 0, \sum_{k=0}^{\infty} a_k = \infty, \text{and} \sum_{k=0}^{\infty} a_k^2 < \infty, \tag{9}$$

$$\inf_{\eta < \left\|\mathbf{x}-\mathbf{x}^*\right\| < 1/\eta} \left(\mathbf{x}-\mathbf{x}^*\right)^T \mathbf{B}\mathbf{g}\left(\mathbf{x}\right) > 0 \quad \text{for all } 0 < \eta < 1 \tag{10}$$

$$\mathrm{E}\left[\mathbf{e}_k\left(\mathbf{x}\right)\right] = 0 \text{ for all } \mathbf{x} \text{ and } k, \tag{11}$$

$$\mathrm{E}\left[\left\|\mathbf{Y}_k\left(\mathbf{x}\right)\right\|^2\right] \le c\left(1 + \left\|\mathbf{x}\right\|^2\right) \text{ for all } \mathbf{x} \text{ and } k \text{ and some } c > 0. \tag{12}$$

where $\mathbf{B}$ is some symmetric, positive definite matrix.

By (7) and Corollary 1, condition in (9) can be substituted by (13) and (14).

$$\sigma < \infty \tag{13}$$

$$b_k > 0, b_k \to 0, \sum_{k=0}^{\infty} b_k = \infty, \text{and} \sum_{k=0}^{\infty} b_k^2 < \infty \tag{14}$$

While, condition in (11) can be derived from Corollary 1 and (8). Similarly the condition in (12) can also be simplified as in (15) when Corollary 1 and (8) are concerned.

$$\left\|\mathbf{g}\left(\hat{\mathbf{x}}_k\right)\right\|^2 \le c\left(1 + \left\|\mathbf{x}\right\|^2\right) \text{ for all } \mathbf{x} \text{ and } k \text{ and some } c > 0. \tag{15}$$

Thus we have the theorem for local convergence property. Note that the detailed proof is omitted due to the page limit.

Theorem 2 *Given that conditions in (10), (13), (14) and (15) are satisfied, the search iterate* $\hat{\mathbf{x}}_k$ *generated by Algorithm 2 converges to a local optimum almost surely.*

4 NUMERICAL EXAMPLES

In this section, we compare GO-POLARS with several benchmark search algorithms including gradient-based search and metaheuristics local search. Besides, the hybrid of GO-POLARS with advanced stochastic search algorithms such as COMPASS will be illustrated as well.

4.1 A Benchmark Comparison

The Goldstein-Price's function is a two-dimensional global optimization test function as defined in (16). Note that the global minimum occurs at $\mathbf{x}^* = \left(0,-1\right)$ with $L\left(\mathbf{x}^*\right) = 3$, and several local minima occur as well. Set the search domain $\Theta = \mathbb{R}^2$ and assume that the gradient can be calculated at every $\mathbf{x} \in \Theta$, we used the function to compare the performance of GO-POLARS with steepest descent (SD) and simulated annealing method (SAN) as described in Table 1.

$$L(\mathbf{x}) = \left[1 + (x_1 + x_2 + 1)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \cdot \left[30 + (2x_1 - 3x_2)^2 \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right] \quad (16)$$

For fair comparison, we adopt a neighborhood structure setting in SAN that is similar to the GO-POLARS iterate. But instead of choosing direction from a polar normal distribution oriented by the gradient, we let it be generated by a multivariate normal distrbibution that does not involve gradient. However, for comparison consistancy, the magnitude of gradient is used in determining the sample distance. In the experiment, we set $a_k = 0.001/k$ and $\sigma = \pi/3$ for all occasions. Besdies, for SAN, the tempreture T_k is set to be $t(500 - k)$. As the experiement does not show significant difference when t is tuned to be any positive value, we set $t = 1$ for illustration.

Table 1: The overview of settings for testing algorithms.

Algorithms	**Search Iterate (Neighborhood Structure)**	**Condition for Accepting $\hat{\mathbf{x}}_{\text{new}}$**
GO-POLARS	$\hat{\mathbf{x}}_{\text{new}} = \hat{\mathbf{x}}_k - a_k \left\|\mathbf{g}(\hat{\mathbf{x}}_k)\right\| \mathbf{d}_k$ where $\mathbf{d}_k \sim \text{N}_{\text{polar}}^p\left(\mathbf{g}(\hat{\mathbf{x}}_k), \sigma^2\right)$	$L(\hat{\mathbf{x}}_{\text{new}}) < L(\hat{\mathbf{x}}_k)$
Steepest descent (SD)	$\hat{\mathbf{x}}_{\text{new}} = \hat{\mathbf{x}}_k - a_k \cdot \mathbf{g}(\hat{\mathbf{x}}_k)$	
Simulated Annealing (SAN)	$\hat{\mathbf{x}}_{\text{new}} = \hat{\mathbf{x}}_k - a_k \left\|\mathbf{g}(\hat{\mathbf{x}}_k)\right\| \frac{\mathbf{d}_k}{\left\|\mathbf{d}_k\right\|}$ where $\mathbf{d}_k \sim \text{N}(\mathbf{0}, \mathbf{I}_p)$	$L(\hat{\mathbf{x}}_{\text{new}}) < L(\hat{\mathbf{x}}_k)$ or $Z < \exp\left(\left(L(\hat{\mathbf{x}}_k) - L(\hat{\mathbf{x}}_{\text{new}})\right) / T_k\right)$ where $Z \sim \text{U}(0,1)$

The three algorithms can be corelated by starting with a same initial solution $\mathbf{x}_0$ that is randomly selected from $[-2,2]^2$, and run the algorithms until $k = 500$. Repeat the process for 50 replications, we then present the average $L(\hat{\mathbf{x}}_k^*)$ in Figure 5. Note that in each replication, $\hat{\mathbf{x}}_k^*$ denotes the best solution visited upon iteration k.

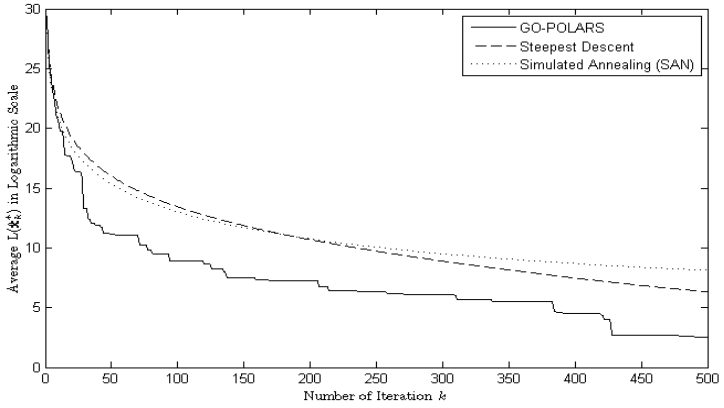

Figure 5: Average $L(\hat{\mathbf{x}}_k^*)$ by different search algorithms.

It is obvious that the average performance of GO-POLARS across replication is superior than both SD and SAN. To analyze the reason, we notice that in SD only single direction is allowed to be sampled, but GO-POLARS ensures that all directions have a positive chance to be selected when $\sigma \neq 0$, which certainly enlarged the pool of candidate solutions. In SAN, the enlarged candidate pool is remianed, however, in order to filter it SAN arbitrarly rejects inferior samples after evaluation using an artificial tempreture paremeter T_k. While in GO-POLARS, solutions on different directions can be filtered in advance with the gradient-oriented random distribution even before any evaluation.

4.2 Hybrid with Stochastic Search

As stated in Section 1, almost all stochastic search algorithms involve random sampling within a specified neighborhood, where it is assumed that gradient information is not available. However, in the cases when gradient can be observed or estimated, we can apply GO-POLARS to help in sampling good solutions more efficiently. In the other hand, if GO-POLARS alone could not obtain desired efficiency, to integrate it with an advanced stochastic search will probably make the achievement.

Assume solutions are to be sampled from a convex set Θ in which $\hat{\mathbf{x}}^* \in \Theta$ is the best known up-todate. We may sample $\hat{\mathbf{x}}_{\text{new}} = \hat{\mathbf{x}}^* - r \cdot \mathbf{d}$ where $\mathbf{d} \sim \mathrm{N}^p_{\text{polar}}\left(\mathbf{g}\left(\hat{\mathbf{x}}^*\right), \sigma^2\right)$ and $r \sim \mathrm{U}\left(0, R\right)$ in which R is the maximum value of r that ensures $\hat{\mathbf{x}}_{\text{new}} \in \Theta$.

We illustrate the concept using COMPASS (Hong and Nelson 2006), which is initially proposed for solving discrete optimization problems, but has been observed performing well also for contineous cases. The main idea of the algorithm is to construct a most-promissing-area after evaluation of all historical samples and in a new iteration retake samples within the area according to a given sampling scheme. For instance, Hong and Nelson (2006) suggest a Revised Mix-D (RMD) method aiming to generate samples almost uniformly. But later it is identified to be less efficient in solving high-dimensional problems, for which the Coordinate Sampling is proposed instead (Hong et al. 2010).

$$L\left(\mathbf{x}\right) = \sum_{i=1}^{p/2}\left[100\left(x_{2i} - x_{2i-1}^2\right)^2 + \left(1 - x_{2i-1}\right)^2\right] \text{ with } \Theta = \left[-4, 4\right]^p \tag{17}$$

We apply the COMPASS on a high-dimension continuous test function as in (17). The function is initially proposed by Rosenbrock (1960) with $p = 2$ and extended by Moré et al. (1981) to higher dimension. Here, we use the setting $p = 10$. Note that it has a unique optimum $L\left(\mathbf{x}^*\right) = 0$ occurring at $\mathbf{x}^* = \left(1, 1, \ldots 1\right)$.

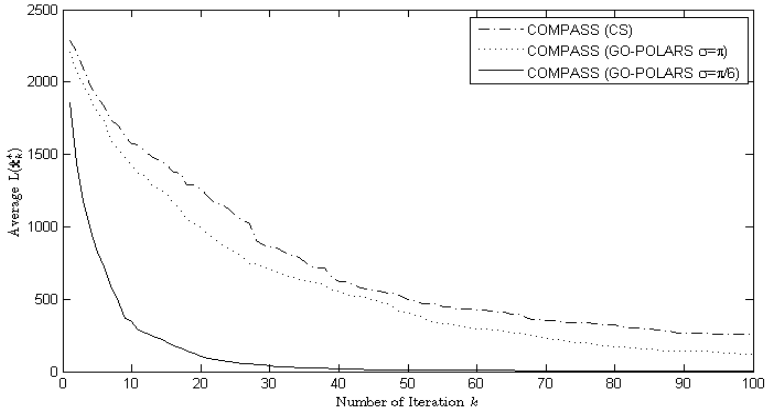

Figure 6: Average $L\left(\hat{\mathbf{x}}_k^*\right)$ by COMPASS with different sampling schemes.

Two sampling schemes are compared in the test, namely the Coordinate Sampling (CS) and GO-POLARS sampling as in (17), for which the σ is set to π and $\pi/6$ respectively. Besides, the batch size of COMPASS, i.e., the number of solutions to be sampled in each iteration, is set to 1.

From the average $L\left(\hat{\mathbf{x}}_k^*\right)$ drawn from 50 replications (Figure 6), we conclude that compared with CS, the hybridized GO-POLARS provides a higher convergent rate and the rate increases as the sampling concentrates to the gradient direction (denoted by smaller σ).

In addition, by a long run study we found it almost impossible for CS converge to the uniqe optimum, simply due to the reason that CS is designed intently for discrete problems while in continuous cases the search could be trapped in the region where solution cannot be improved on any coordinate directions. Thus, for COMPASS to be applied in solving continuous problems, GO-POLARS is one of the only choices.

5 CONCLUSION

In this paper, we reviewed two categories of search algorithms for optimization problems and suggest that incorporating randomness in utilizing gradient information will improve both gradient-based search and metaheuristics local search. A brand new algorithms GO-POLARS is built on this purpose using a generalized polar coordinate representation and associated random distributions. It has been shown that GO-POLARS has the strong local convergence property and works well in numerical examples either independently or hybridizing with sophisticated stochastic algorithms such as COMPASS.

Future study may address the adjustment of σ and analyze how it affects the solutions quality versus search efficiency for different applications. The possibility to hybirdize GO-POLARS with other stochastic search algorithms can also be discussed. Besides, instead of gradient, other directional information based on the nature of respective problems can also be used to orient the polar random distribution. Then a large number of search and sampling algorithms can be developed based on the concept. Overall, with the promissing numerical results and the broad derivatives, we have plenty of reason to believe that GO-POLARS is openning a new era of polar search.

REFERENCES

Arfken , G. (1985). *Spherical Polar Coordinates* (3rd ed.). Orlando, FL: Academic Press.

Blum, J. R. (1954a). Approximation Methods Which Converge with Probability One. *Annals of Mathematical Statistics, 25*, 382-386.

Blum, J. R. (1954b). Multidimensional Stochastic Approximation Methods. *Annals of Mathematical Statistics, 25*, 737-744.

Debye, P. (1909, 12 1). Näherungsformeln für die Zylinderfunktionen für große Werte des Arguments und unbeschränkt veränderliche Werte des Index. *Journal Name: Mathematische Annalen, 67*(4), pp. 535-558.

Glover, F. (1990, July-August). Tabu Search; A Tutorial. *Interfaces, 20*(4), pp. 74-94.

Hong, L. J., & Nelson, B. L. (2006, January-February). Discrete Optimization via Simulation Using COMPASS. *Operations Research, 54*(1), pp. 115-129.

Hong, L. J., Xu, J., & Nelson, B. L. (2010, September 21). Speeding up COMPASS for high-dimensional discrete optimization via simulation. *Operations Research Letters*, pp. 550-555.

Kaplan, W. (1991). *Advanced Calculus* (4th ed. ed.). Redwood City, CA: Addison-Wesley.

Kiefer, J., & Wolfowitz, J. (1952). Stochastic Estimation of the Maximum of a Regression Function. *Annals of Mathematical Statistics, 23*(3), pp. 462-466.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983, 5 13). Optimization by Simulated Annealing. *Science, 220*(4598), pp. 671-680.

Matyas, J. (1965). Random Optimization. *Automation and Remote Control, 26*, 244-251.

Moré, J. J., Garbow, B. S., & Hillstrom, K. E. (1981). Testing Unconstrained Optimization Software. *ACM Transactions on Mathematical Software, 7*(1), 17-41.

Neal, R. M. (2003). Slice Sampling. *Annals of Statistics, 31*(3), 705–767.

Nevel'son, M. B., & Has'minskii, R. Z. (1973, c1976). *Stochastic Approximation and Recursive Estimation.* Providence, RI: American Mathematical Society.

Pogu, M., & Souza de Cursi, J. E. (1994). Global Optimizationby Random Perturbation of the Gradient Method with a Fixed Parameter. *Journal of Global Optimization, 5*, 159 - 180.

Pohlheim, H. (2007, 1 15). *Examples of Objective Functions.* Retrieved 4 10, 2012, from GEATbx - The Genetic and Evolutionary Algorithm Toolbox for Matlab: http://www.geatbx.com/download/GEATbx_ObjFunExpl_v38.pdf

Rosenbrock, H. H. (1960). An Automatic Method for Finding the Greatest or Least Value of a Function. *The Computer Journal, 3*(3), 175-184.

Schwarz, K. (2011, 12 29). *Darts, Dice, and Coins: Sampling from a Discrete Distribution*. Retrieved 3 28, 2012, from KeithSchwarz.com: http://www.keithschwarz.com/darts-dice-coins/

Shi, L., & Ólafsson, S. (2000, May-June). Nested Partitions Method for Global Optimization. *Operations Research, 48*(3).

Spall, J. C. (1998). An Overview of the Simultaneous Perturbation Method for Efficient Optimization. *Johns Hopkins APL Technical Digest, 19*(4), pp. 482-492.

Spall, J. C. (2003). *Introduction to stochastic search and optimization: estimation, simulation, and control.* John Wiley & Sons, Inc.

Vose, M. D. (1999). A Linear Algorithm For Generating Random Numbers With a Given Distribution. *IEEE Transactions on Software Engineering, 17*(9), 972-975.

Walton, J. J. (1963, March). Tensor calculations on computer: appendix. *Communications of the ACM, 10*(3), 183-186.

Weisstein, E. W. (2003, 9 5). *Hypersphere*. Retrieved 3 14, 2012, from MathWorld: http://mathworld.wolfram.com/Hypersphere.html

Weisstein, E. W. (2005, 10 26). *Spherical Coordinates*. Retrieved 3 14, 2012, from MathWorld: http://mathworld.wolfram.com/SphericalCoordinates.html

Weisstein, E. W. (2009, 3 8). *Polar Coordinates*. Retrieved 3 14, 2012, from MathWorld: http://mathworld.wolfram.com/PolarCoordinates.html

AUTHOR BIOGRAPHIES

HAOBIN LI is a Ph.D. candidate in the Department of Industrial and Systems Engineering, National University of Singapore. He received his B.Eng. degree with 1st Class Honors in Industrial and Systems Engineering from National University of Singapore in 2009. His research interests include analytical methods of simulation, and multi-objective optimization. His email address is li_haobin@nus.edu.sg.

LOO HAY LEE is an Associate Professor and Deputy Head in the Department of Industrial and Systems Engineering, National University of Singapore. He received his B.S. (Electrical Engineering) degree from the National Taiwan University in 1992 and his S. M. and Ph.D. degrees in 1994 and 1997 from Harvard University. He is currently a senior member of IEEE, a committee member of ORSS, and a member of INFORMS. His research interests include production planning and control, logistics and vehicle routing, supply chain modeling, simulation-based optimization, and evolutionary computation. His email address is iseleelh@nus.edu.sg.

EK PENG CHEW is an Associate Professor and Deputy Head in the Department of Industrial and Systems Engineering, National University of Singapore. He received his Ph.D. degree from the Georgia Institute of Technology. His research interests include logistics and inventory management, system modeling and simulation, and system optimization. His email address is isecep@nus.edu.sg.