Project number IST-25582

CGL

Computational Geometric Learning

Random Pursuit in Hilbert Space

STREP

Information Society Technologies

Period covered:	November 1, 2012–October 31, 2013
Date of preparation:	November 11, 2013
Date of revision:	November 21, 2013
Start date of project:	November 1, 2010
Duration:	3 years
Project coordinator name:	Joachim Giesen (FSU)
Project coordinator organisation:	Friedrich-Schiller-Universität Jena Jena, Germany

*The project CG Learning acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 255827.
†Institute of Theoretical Computer Science, ETH Zürich, stich@inf.ethz.ch
‡Institute of Theoretical Computer Science, ETH Zürich, gaertner@inf.ethz.ch

Random Pursuit in Hilbert Space*

Technical Report

Sebastian U. Stich† Bernd Gärtner‡

14 November 2013

Abstract

We define and analyze Random Pursuit (RP) – a random walk in a Hilbert space. RP is defined by iterative projections onto randomly selected hyperplanes. This process originates from several applications in derivative-free optimization.

In this report we study convergence in the induced norm. We present our results in an unifying way that allows to apply them in various settings. Especially, we recover the previously known convergence results form two applications.

1 Introduction

Let H^n denote a Hilbert space over $\mathbb{R}$ of dimension n with associated scalar product $\langle \cdot, \cdot \rangle : H^n \times H^n \to \mathbb{R}$ and induced norm $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. We consider the following random walk in H. For an arbitrary starting point $\mathbf{x}_0 \in H^n$, at each step k a uniformly random direction $\mathbf{u}_k \in S^{n-1}$ is chosen and x_k is projected on the hyperplane orthogonal to $\mathbf{u}_k$. We call this algorithm Random Pursuit (RP). In formulas, the process can be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \langle \mathbf{x}_k, \mathbf{u}_k \rangle \mathbf{u}_k \,. \tag{1}$$

Due to the fact that all the steps are projections, it is intuitively clear that the distance to the origin can only decrease in each step and thus $\|x_k\|$ eventually converges to zero.

In this short report, we study the above process (1) and establish a bound on the convergence rate. We will also consider a slight generalization, where we omit the restriction that the random directions $\mathbf{u}_k$ are uniformly distributed, but we allow also for different distributions (allowing even discrete support), as long as the whole space is sufficiently covered.

1.1 Motivation

Process (1) arises in several applications in derivative-free optimization. We now introduce the most apparent one which arises by studying the behaviour of a very simple optimization algorithm on an isotropic quadratic function.

A simple derivative-free scheme that can be used to minimize convex functions is the following algorithm which is called Random Pursuit optimization algorithm (cf. [15]). In every iteration – due to lack of more information about the objective function – a random direction is chosen, and the objective function is minimized along this direction. This method ranges among the simplest possible optimization schemes as it soleley relies on two easy-to-implement primitives: a random direction generator and a one-dimensional line search.

The RP optimization algorithm has been studied in various places in the literature, first it was mentioned by Mutseniyeks and Rastrigin [9]. Convergence analyses on strongly convex functions have been provided by Krutikov [6] and Rappl [10]. Rappl proved linear convergence of RP optimization algorithm without giving exact convergence rates. Krutikov showed linear convergence in the special case where the search directions are given by n linearly independent vectors which are used in cyclic order. Karmanov [4, 5, 16] already conducted an analysis of this algorithm on general convex functions. Stich et al. [15] extended this work by considering also inexact line search and different sampling distributions for the search direction.

In Section 3 we show that applying the RP optimization algorithm to the simple sphere function $f_1(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, \mathbf{x} \rangle$, is equivalent to random process (1). This not only explains why we do refer to (1) as RP too, but it allows relate the existing results in the literature concerning the RP optimization algorithm to the process (1). Especially, the lower bound on the convergence rate provided by Jägersküpper [3] directly extends to to (1).

In the remainder of Section 3 we will also mention a more recent application, that was introduced by Leventhal and Lewis [7]. They proposed a random process to estimate a unknown Hessian matrix of a convex function. Again we show that this random process is a special case of (1).

2 Random Pursuit

We will now formally introduce the random process RP. As mentioned in the introduction, we consider a slight generalization of the prototype (1).

We denote by $S^{n-1} := \{\mathbf{x} \in H^n \mid \|\mathbf{x}\| = 1\}$ the set of unit vectors in H^n and let ν a probability distribution over S^{n-1}. Define

$$\alpha_\nu := \inf_{\substack{\mathbf{x} \in H^n \\ \|\mathbf{x}\|=1}} \mathbb{E}_{\mathbf{u}} \left[\langle \mathbf{x}, \mathbf{u} \rangle^2 \right] , \tag{2}$$

where $\mathbf{u} \sim \nu$. Thus for every $\mathbf{x} \in H^n$:

$$\mathbb{E}_{\mathbf{u}} \left[\langle \mathbf{x}, \mathbf{u} \rangle^2 \right] \geq \alpha_U \|\mathbf{x}\|^2 . \tag{3}$$

A distribution ν with $\alpha_\nu > 0$ is called *rich enough* to serve as a sampling distribution for the directions in the process (1). The random process is then defined as follows:

For $\mathbf{x}_k \in H^n$ the next iterate of the random process is defined as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \langle \mathbf{x}_k, \mathbf{u}_k \rangle \mathbf{u}_k , \tag{4}$$

where $\mathbf{u}_k \sim \nu$ and ν is a rich enough sampling distribution. We denote this random process as Random Pursuit (RP) (staring from $\mathbf{x}_0 \in H^n$ with sampling distribution ν).

Remark 2.1. *Let $\mathbf{x}_k, \mathbf{x}_{k+1}$ and $\mathbf{u}_k$ satisfy* (4)*. Then*

$$\textit{(i)} \quad \langle \mathbf{x}_{k+1}, \mathbf{u}_k \rangle = 0 \qquad\qquad \textit{(ii)} \quad \|\mathbf{x}_{k+1}\| \leq \|\mathbf{x}_k\| . \tag{5}$$

Proof. To show (i), we calculate the scalar product

$$\langle \mathbf{x}_{k+1}, \mathbf{u}_k \rangle = \langle \mathbf{x}_k, \mathbf{u}_k \rangle - \langle \mathbf{x}_k, \mathbf{u}_k \rangle \underbrace{\langle \mathbf{u}_k, \mathbf{u}_k \rangle}_{=1} = 0 . \tag{6}$$

For (ii), we calculate the norm explicitly:

$$\begin{aligned} \|\mathbf{x}_{k+1}\|^2 = \langle \mathbf{x}_{k+1}, \mathbf{x}_{k+1} \rangle &= \langle \mathbf{x}_k, \mathbf{x}_k \rangle - 2 \langle \mathbf{x}_k, \mathbf{u}_k \rangle \langle \mathbf{u}_k, \mathbf{x}_k \rangle + \langle \mathbf{x}_k, \mathbf{u}_k \rangle^2 \\ &= \|\mathbf{x}_k\|^2 - \underbrace{\langle \mathbf{x}_k, \mathbf{u}_k \rangle^2}_{\geq 0} , \end{aligned} \tag{7}$$

where we have again used $\langle \mathbf{u}_k, \mathbf{u}_k \rangle = 1$. ∎

2.1 Sampling Distributions

A simple way to construct rich enough sampling distributions is to consider a subset $U \subset S^{n-1}$ of points on the unit sphere and take the uniform distribution ν_U over the points in U. In this section we give a few examples.

Example 2.2 (Spherical distribution in $\mathbb{R}^n$)**.** *Let ν_S denote the uniform distribution over S^{n-1} in $\mathbb{R}^n$. Then ν_S is rich enough with $\alpha_S = \frac{1}{n}$. [15, Lemma 3.3].* □

Example 2.3 (Standard unit vectors in $\mathbb{R}^n$)**.** *Let $U = \{e_i \mid i = 1, \dots, n\}$ the set of standard unit vectors and ν_Z denote the uniform distribution over U. Then ν_U is rich enough with $\alpha_U = \frac{1}{n}$. [15, Lemma 3.4].* □

Example 2.4 (Ellipsoidal distribution in $\mathbb{R}^n$)**.** *Let $\mathbf{y} \sim \mathcal{N}(0, \Sigma)$ multivariate normal with covariance matrix $\Sigma \in \mathrm{PSD}_n$. Considering normalized samples $\bar{\mathbf{y}} := \frac{\mathbf{y}}{\|\mathbf{y}\|}$ defines a distribution ν_Σ on S^{n-1}. This distribution is rich enough with $\alpha_\Sigma = \frac{1}{\kappa(\Sigma)n}$, where $\kappa(\Sigma) := \frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)}$ denotes the condition number of Σ.*

Proof. We use the elementary estimate $\|\mathbf{y}\|^2 \leq \lambda_{\max}(\Sigma) \|\mathbf{y}\|^2_{\Sigma^{-1}}$ (cf. [12, Lemma 1]), to deduce for any $\mathbf{x} \in \mathbb{R}^n$:

$$\frac{\langle \mathbf{x}, \mathbf{y} \rangle^2}{\|\mathbf{y}\|^2} \geq \frac{\langle \mathbf{x}, \mathbf{y} \rangle^2}{\lambda_{\max}(\Sigma) \|\mathbf{y}\|^2_{\Sigma^{-1}}}, \tag{8}$$

where $\|\mathbf{y}\|^2_{\Sigma^{-1}} = \mathbf{y}^T \Sigma^{-1} \mathbf{y}$. Applying Lemma 4(ii) from [12] to the right hand side yields

$$\mathbb{E}_{\mathbf{y}} \left[\frac{\langle \mathbf{x}, \mathbf{y} \rangle^2}{\|\mathbf{y}\|^2} \right] \geq \frac{\mathrm{Tr}[\mathbf{x}\mathbf{x}^T \Sigma]}{\lambda_{\max}(\Sigma) n} \geq \frac{\lambda_{\min}(\Sigma) \mathrm{Tr}[\mathbf{x}\mathbf{x}^T]}{\lambda_{\max}(\Sigma) n} = \frac{\|\mathbf{x}\|^2}{\kappa(\Sigma) n}, \tag{9}$$

and shows the claim. □

Example 2.5 (Rank one matrices in SYM_n)**.** SYM_n *denotes the space of all $n \times n$ symmetric matrices. This is a Hilbert space with the standard scalar product $\langle A, B \rangle = \mathrm{Tr}[AB^T]$. Let $U = \{X \in \mathrm{SYM}_n \mid X = \mathbf{x}\mathbf{x}^T, \mathbf{x} \in \mathbb{R}^n\}$ symmetric rank one matrices. Let $\mathbf{x} \sim \nu_S$ a random unit vector (see Example 2.2). Then $X = \mathbf{x}\mathbf{x}^T$ defines a probability distribution on U. This sampling distribution is rich enough with $\alpha = \frac{2}{n(n+2)}$. [7, Theorem 2.1].* □

2.2 Convergence

We now prove convergence of RP (4). We will see below that this is an immediate consequence of sampling the search directions form a rich enough distribution.

We have mentioned in the introduction, that the random process RP (4) originates as a generalization of several random processes in different applications (some of them will be discussed later in Section 3). Hence it should also not come as a surprise, that the (short) proof goes along the same lines as the specialized proofs in the respective applications.

This similarity can be most prominently spotted in [7] or [14]. But also Karmanov [4, 5, 16] was essentially following the same idea. By looking very closely, the one realizes that the same argument can be found in [15] and [12]. And very likely in many more places in the literature.

Theorem 2.6. *Let $\mathbf{x}_0 \in H^n$ and $\{\mathbf{x}_k\}_{k \geq 0}$ a sequence of iterates generated by RP (4) with rich enough sampling distribution μ with parameter α. Then*

$$\text{(i)} \quad \mathbb{E}\left[\|\mathbf{x}_{k+1}\|^2 \mid \mathbf{x}_k \right] \leq (1 - \alpha) \|\mathbf{x}_k\|^2, \tag{10}$$

$$\text{(ii)} \quad \mathbb{E}\left[\|\mathbf{x}_k\|^2 \right] \leq (1 - \alpha)^k \|\mathbf{x}_0\|^2, \tag{11}$$

$$\text{(iii)} \quad \|\mathbf{x}_k\| \leq \|\mathbf{x}_0\|^2 \cdot \exp\left[-\sum_{i=0}^{k-1} \frac{\langle \mathbf{x}_i, \mathbf{u}_i \rangle^2}{\|\mathbf{x}_i\|^2} \right], \tag{12}$$

where in (iii) the $\mathbf{u}_k$ denote the realizations that generated $\{\mathbf{x}_k\}_{k\geq 0}$.

Proof. Exactly as in the proof of Remark 2.1 we obtain

$$\mathbb{E}\left[\|\mathbf{x}_{k+1}\|^2 \mid \mathbf{x}_k\right] = \|\mathbf{x}_k\|^2 - \mathbb{E}\left[\langle \mathbf{x}_k, \mathbf{u}_k\rangle^2 \mid \mathbf{x}_k\right] . \tag{13}$$

Now claim (i) follows by the definition of a rich distribution (3) with parameter α. Now we proceed to (ii). By part (i) we can bound

$$\mathbb{E}\left[\|\mathbf{x}_k\|^2 \mid \mathbf{x}_{k-1}\right] \leq (1-\alpha)\|\mathbf{x}_{k-1}\|^2 . \tag{14}$$

What we now have to do formally, is to condition on $\mathbf{x}_{k-2}$ and calculate the expectation of $\mathbb{E}[E[\mathbf{x}_k \mid \mathbf{x}_{k-1}] \mid \mathbf{x}_{k-2}]$. By the tower property of conditional expectations, $\mathbb{E}[E[\mathbf{x}_k \mid \mathbf{x}_{k-1}] \mid \mathbf{x}_{k-2}] = \mathbb{E}[\mathbf{x}_k \mid \mathbf{x}_{k-2}]$. Repeating this procedure for $\mathbf{x}_{k-3}$ up to $\mathbf{x}_0$, we finally obtain $E[\mathbf{x}_{k+1} \mid \mathbf{x}_0] = \mathbb{E}[\mathbf{x}_{k+1}]$ and the statement follows.

Finally, for part (iii), we deduce form Equation (7):

$$\|\mathbf{x}_k\|^2 = \left(1 - \frac{\langle \mathbf{x}_{k-1}, \mathbf{u}_{k-1}\rangle^2}{\|\mathbf{x}_{k-1}\|^2}\right)\|X_{k-1}\|^2 . \tag{15}$$

With $1 - x \leq e^{-x}$ we obtain the final result. □

3 Applications

In this section we present some applications where the random process (4) naturally appears. We show two examples form optimization. The first one is about minimizing a (strongly) convex objective function, the latter is about estimating a Hessian matrix from an unknown convex function.

3.1 Optimization with RP

In this section we consider functions $f \colon H^n \to \mathbb{R}$. The first function we consider is very simple: $f_1 := \frac{1}{2}\langle \mathbf{x}, \mathbf{x}\rangle$, the sphere function. It is often used in the literature as a fist benchmark to test the performance of new (derivative-free) algorithms numerically [1, 8] or analytically [11, 2].

We make the following observation: optimizing f_1 with the Random Pursuit optimization algorithm [9, 4, 5, 16, 15] is equivalent to RP (4) with spherical sampling distribution ν_S (cf. Example 2.2).

Let $\mathbf{x}_k \in H^n$ be given, and let $\mathbf{u}_k \sim \nu_S$. Then $\mathbf{x}_{k+1}$ in the Random Pursuit optimization algorithm is defined as the minimizer of f_1 on the line $\{\mathbf{x} \mid \mathbf{x} = \mathbf{x}_k + t\mathbf{u}_k, t \in \mathbb{R}\}$. We observe

$$\mathbf{x}_{k+1} := \arg\min_{t\in\mathbb{R}} f_1(\mathbf{x}_k + t\mathbf{u}_k) = \arg\min_{t\in\mathbb{R}} t\langle \mathbf{x}_k, \mathbf{u}_k\rangle + \frac{1}{2}t^2 \underbrace{\langle \mathbf{u}_k, \mathbf{u}_k\rangle}_{=1}, \tag{16}$$

and now it is immediate that $\mathbf{x}_{k+1}$ satisfies Equation (4).

Corollary 3.1 (Convergence on Sphere)**.** *Let $\{\mathbf{x}_k\}_{k\geq 0}$ be a sequence of iterates generated by the Random Pursuit optimization algorithm [9, 4, 5, 16, 15] on f_1. Then*

$$\mathbb{E}[f_1(\mathbf{x}_k)] \leq (1-\alpha)^k f_1(\mathbf{x}_0), \tag{17}$$

with $\alpha = \frac{1}{n}$.

This follows form the above observation, Example 2.2 and Theorem 2.6. □

An analogous statement can be derived for the Random Pursuit optimization algorithm when the the steps are only taken along coordinate directions. We can recover the results form Krutikov [6] by using the statement form Example 2.3 and proceeding similarly as in Corollary 3.1.

The result from Corollary 3.1 is optimal. An argument by Jägersküpper [3] shows, that no search algorithm that selects the search directions from ν_S, can exhibit faster expected convergence as (17).

The Theorem can not only be applied to the sphere function. Let $f_2 = \frac{1}{2}\langle C\mathbf{x}, C\mathbf{x}\rangle$ be a quadratic function with $C \in \mathbb{R}^{n\times n}$ invertible.

Corollary 3.2 (Convergence on Ellipsoid)**.** *Let $\{\mathbf{x}_k\}_{k\geq 0}$ be a sequence of iterates generated by the Random Pursuit optimization algorithm [9, 4, 5, 16, 15] on f_2. Then*

$$\mathbb{E}[f_2(\mathbf{x}_k)] \leq (1-\alpha)^k f_2(\mathbf{x}_0)\,, \tag{18}$$

with $\alpha = \frac{1}{\kappa n}$, where $\kappa = \kappa(C^T C)$ denotes the condition number of f_2.

Proof. The mapping $\mathbf{x} \mapsto C\mathbf{x}$ describes a transformation. Applying the inverse transformation $\mathbf{x} \mapsto C^{-1}\mathbf{x}$ maps f_2 to f_1, the sphere function. We also have to transform the sampling distribution. The uniform distribution ν_S can be generated by considering normalized samples $\frac{\mathbf{x}}{\|\mathbf{x}\|}$ of $\mathbf{x} \sim \mathcal{N}(0, I_n)$. Applying the inverse transformation yields $\mathcal{N}(0, C^{-1}C^{-T}) = \mathcal{N}(0, \Sigma^{-1})$. Thus $\nu_{\Sigma^{-1}}$ (cf. Example 2.4 is the corresponding transformed sampling distribution[1]. Now the statement follows from Example 2.4 with $\kappa = \kappa(\Sigma) = \kappa(\Sigma^{-1})$. □

3.2 Metric learning with RP

The *Randomized Hessian Approximation (RHA)* scheme as introduced in [7] is a random walk on the space of symmetric $n\times n$ matrices, depending on a (fixed) positive definite matrix H. For $Y_0 \in \mathbb{R}^{n\times n}$ symmetric, a step of the random walk is defined as:

$$Y_{k+1} := Y_k - \langle Y_k - H, U_k\rangle\, U_k\,, \tag{19}$$

where $U_k = \mathbf{u}_k\mathbf{u}_k^T$ for a uniform random direction $\mathbf{u}_k \sim \nu_S$, the U_k follows the distribution form Example 2.5. We observe, that the error matrices $X_k := Y_k - H$ satisfy Equation (4) with $X_0 = Y_0 - H$. Hence, Theorem 2.6 can also be applied to RHA.

Corollary 3.3 (Random Hessian Approximation)**.** *Let $\{X_k\}_{k\geq 0}$ be a sequence of iterates generated by RHA [7, 12, 14]. Then*

$$\mathbb{E}[\|X_k\|_2] \leq (1-\alpha)^k \|X_0\|^2\,, \tag{20}$$

with $\alpha = \frac{2}{n(n+2)}$. The norm $\|X\| = \sqrt{\langle X, X\rangle}$ is the standard Frobenius norm.

This follows form Theorem 2.6 and Example 2.5.

From the way the RHA scheme was introduced here in Equation (19) and the comments in Section 3.1 on the Random Pursuit optimization algorithm, it is apparent, that RHA is equivalent to optimization of the objective function $f_L := \frac{1}{2}\langle Y_k - H, Y_k - H\rangle = \frac{1}{2}\|Y_k - H\|^2$, the sphere function in SYM_n, the Random Pursuit optimization algorithm.

This connection might not be so obvious from the presentation in [7]. Let us state a few remarks. In [7], the unknown matrix $H \in \mathrm{SYM}_n$ is encoded in a function $g(\mathbf{x}) := \frac{1}{2}\langle H\mathbf{x}, \mathbf{x}\rangle$. The scalar product $\langle H, U_k\rangle$ for $U_k = \mathbf{u}_k^T\mathbf{u}_k$ is then evaluated by tree evaluations of g:

$$\langle H, U_k\rangle = \langle H\mathbf{u}_k, \mathbf{u}_k\rangle = \frac{g(\mathbf{x}-\epsilon\mathbf{u}_k) - 2g(\mathbf{x}) + g(\mathbf{x}+\epsilon\mathbf{u}_k)}{\epsilon^2}\,, \tag{21}$$

where $\epsilon > 0$ and $\mathbf{x}$ arbitrary (note that g is quadratic). We already remarked in [13] that the symmetry in the right hand side is not needed: instead of evaluating g at $\mathbf{x}$ and $\mathbf{x} \pm \epsilon\mathbf{u}_k$, *any* tree points on a line defined by $\mathbf{u}_k$ would do (interpolation formula for quadratic functions).

[1] Due to symmetry of f_1, we could equivalently also consider ν_Σ.

4 Conclusion

In this report we have analyzed a simple random process in a Hilbert space. We have show convergence in the norm, that is, the norm of successive iterates will converge to zero, if the underlying sampling distribution is rich enough.

We gave examples of rich enough sampling distributions and showed applications where these distributions appear in the literature. Our results shows (i) the convergence of the Random Pursuit optimization algorithm on quadratic function, partially covering the results in [15]. And (ii) the convergence of the Randomized Hessian Approximation scheme, partially covering the results from [7].

There are several open questions that were not considered in this report. First, we did not investigate the impact of inexact/approximate steps. That is, if we would allow iterates that do not satisfy Equation (4) exactly, but only up to some fixed accuracy. For the Random Pursuit optimization algorithm it is known [4, 5, 16] that small errors do have little impact on the convergence rate. Hence, it should also be possible to incorporate this feature in the present framework.

Second, our results for the Random Pursuit optimization algorithm (Cor. 3.2) does only hold for quadratic objective functions but not for general convex functions. It is open, whether there is an extension of the framework that would also allow to treat this more general situation, without loosing its current elegance.

In this report, we studied only the random process (4) only with respect to convergence in the norm, but we did not focus on the trajectories. For certain applications, like the Matrix estimation in [7], it would be interesting to also have some quantitative statements about typical trajectories.

References

[1] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaption in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[2] J. Jägersküpper. Analysis of a simple evolutionary algorithm for minimization in euclidean spaces. In J. Baeten, J. Lenstra, J. Parrow, and G. Woeginger, editors, *Automata, Languages and Programming*, volume 2719 of *Lecture Notes in Computer Science*, pages 1068–1079. Springer Berlin Heidelberg, 2003.

[3] J. Jägersküpper. Lower bounds for hit-and-run direct search. In J. Hromkovic, R. Královic, M. Nunkesser, and P. Widmayer, editors, *Stochastic Algorithms: Foundations and Applications*, volume 4665 of *Lecture Notes in Comput. Sci.*, pages 118–129. Springer Berlin, 2007.

[4] V. G. Karmanov. Convergence estimates for iterative minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 14(1):1 – 13, 1974.

[5] V. G. Karmanov. On convergence of a random search method in convex minimization problems. *Theory of Probability and its applications*, 19(4):788–794, 1974. (in Russian).

[6] V. N. Krutikov. On the rate of convergence of the minimization method along vectors in given directional sy. *USSR Comput. Maths. Phys.*, 23(1):154–155, 1983. in russian.

[7] D. Leventhal and A. S. Lewis. Randomized Hessian estimation and directional search. *Optimization*, 60(3):329–345, 2011.

[8] C. L. Müller and I. F. Sbalzarini. Gaussian adaptation revisited - an entropic view on covariance matrix adaptation. In C. Di Chio et al., editor, *EvoApplications*, number 6024 in Lecture Notes in Comput. Sci., pages 432–441, Berlin, 2010. Springer.

[9] V. A. Mutseniyeks and L. A. Rastrigin. Extremal control of continuous multi-parameter systems by the method of random search. *Eng.Cyb.*, 1:82–90, 1964.

[10] G. Rappl. On Linear Convergence of a Class of Random Search Algorithms. *ZAMM Z. Angew. Math. Mech.*, 69(1):37–45, 1989.

[11] M. Schumer and K. Steiglitz. Adaptive step size random search. *Automatic Control, IEEE Transactions on*, 13(3):270–276, 1968.

[12] S. U. Stich and C. L. Müller. On spectral invariance of randomized hessian and covariance matrix adaptation schemes. In C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, editors, *Parallel Problem*

Solving from Nature - PPSN XII, volume 7491 of *Lecture Notes in Computer Science*, pages 448–457. Springer Berlin Heidelberg, 2012.

[13] S. U. Stich, C. L. Müller, and B. Gärtner. Variable Metric Random Pursuit. *in preparation for Math. Prog.*, 2012.

[14] S. U. Stich, C. L. Müller, and B. Gärtner. Matrix-valued Iterative Random Projections. Technical report, ETH Zürich, 2013. Technical Report CGL-TR-87.

[15] S. U. Stich, C. L. Müller, and B. Gärtner. Optimization of convex functions with Random Pursuit. *SIAM Journal on Optimization*, 23(2):1284–1309, 2013.

[16] R. Zieliński and P. Neumann. *Stochastische Verfahren zur Suche nach dem Minimum einer Funktion*. Akademie-Verlag, Berlin, Germany, 1983.