The Barycenter Method for Direct Optimization: an Overview

Felipe Pait, *Senior Member, IEEE*

***Abstract*— The goal of this paper is to present the recently developed barycenter method for direct optimization, whose properties make it particularly useful in control applications.**

I. INTRODUCTION

The recently developed barycenter method for direct optimization has properties that make it particularly promising in control applications. These properties include:

1) The barycenter method is a form of derivative–free optimization: it aims at finding extremal points of a function whose mathematical expression is not precisely known. In Section II we give an overview of derivative–free optimization. We believe that the discussion, although somewhat long–winded, can be helpful as an introduction to the field and as an argument in favor of a direct optimization method which has a grounding in closed–loop controls thinking.
2) The method can be described both in a recursive, algorithmic manner, and in a closed form or batch expression which is suitable for mathematical analysis. These expressions, given in Section III, provide the method with a transparent interpretation, in contrast with the ad–hoc justifications of many other direct search procedures.
3) The method is compatible with other well–established search procedures, and thus can make use of previous understanding of what works in a given situation. On the other hand, if no a priori knowledge is available, a purely random local search strategy may be employed. In this most unfavorable of situations, the barycenter method behaves in a gradient-descent manner. This is the content of Theorem 1, stated in Section IV.
4) Theorem 2 in Section IV is a formula for the variance of the estimate when a random search is employed, and Theorem 3 contains an important property of the complex version of the barycenter method. It is our belief that the complex version, which is yet to be more fully tested, can behave in a most desirable way for a variety fo applications, including to controls.
5) The barycenter method is intrinsically applicable to non–differentiable functions and robust to measurement noise, as shown in Theorem 4, the final result stated in Section IV.

The proofs of the results stated in Section IV are in the full paper, available in the arXiv [7]. Simulations are presented and interpreted in Section V. Results, applications, and directions for further work are discussed in Section VI.

Escola Politécnica da Universidade de São Paulo, SP, Brazil; e-mail: pait@usp.br.

Although the work discussed in this paper is essentially on static, nonlinear mathematical optimization theory, rather than referring to dynamical control systems, we judge that the method's novelty and its potential for applications in controls related problems is such that a presentation at the CDC is desirable.

II. DIRECT OPTIMIZATION IN CONTEXT

The barycenter method, which will be expressed by formula (1) in III, is suitable for direct optimization, once scorned [13], now a respectable research area, both for its scientific challenges and practical applications. Also known as *derivative–free optimization*, it deals with the search for extrema of a given function, employing only the values of the function and not its mathematical expression. A recent book [3] serves as a good entry point to the literature. When the 1st and perhaps also 2nd derivatives of the function are available, use of gradients and Hessians leads to steepest–descent and Newton–like search algorithms. Often however derivatives are costly or impossible to compute. The challenge in direct optimization is to obtain algorithms with comparable performance, without knowledge of the derivatives.

The barycenter method has been employed successfully to tune filters in system identification [12], see also [11]. The filter parameters are additional structure parameters that need to be chosen before the model parameters themselves can be optimized, and may be said to perform a role similar to that of hyper–parameters in machine learning. With that experience in mind, the use of the barycenter method to tune hyper–parameters seems worth exploring. A continuous–time version of the barycenter algorithm was analyzed in [8]. Some aspects of the method and its applications were presented at SIAM conferences [9], [10].

The aim of this section is to discuss derivative–free optimization in the context of the perhaps more familiar methods of nonlinear programming, and the barycenter method in the context of derivative–free optimization, always with a point of view that is appropriate for the kinds of problems that may appear in control theory and applications.

A. Nonlinear optimization

Nonlinear programming methods usually start from the assumption that a mathematical expression for the function being minimized exists. This is often not the case. In many problems, the physical or other scientific principles behind the problem suggest general properties of the function f, but do not supply exact values. Perhaps determining the function would require extensive modeling work that is costly or impossible to perform. Perhaps the function changes over

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.

time with the conditions of operation of some machinery, or its precise values are subject to some uncertainty or corrupted by measurement noise. In this case we may need to avail ourselves of some method for direct, also called derivative-free optimization.

What we have is an oracle: an experiment, or perhaps a computer simulation, which will supply the value of $f(x_i)$ at a point x_i whenever questioned. Each oracle query has a cost: performing the experiment will need human labor, a simulation will take computer time. Also perhaps the act of trying out a certain parameter guess will have an effect on the process being studied, which may be undesirable if the value of f is high. We have in mind particularly the case of real-time decisions. For example suppose we are trying to optimize the parameters of a feedback process control device. In the time during which we tried out a "bad" controller, the performance of the process was poor. Now we will have learned to avoid that particular set of parameters, at the cost of allowing for poor control during a certain interval. The criteria by which we judge a direct optimization method may be different from the usual criteria for numerical optimization, which focus exclusively on computational complexity and demands.

B. Derivative–free optimization

One way to approach the problem would be to estimate the function f using a sequence of oracle test queries. Whether this is promising depends crucially on our previous knowledge about the shape of the function and its properties. If the function can be described with a small number of parameters, and those can be computed on the basis of a limited number of polled values, then such a method may be quite effective. After a mathematical expression is derived, optimization may proceed very much along the lines described previously for known functions.

If however our prior knowledge of the function is limited, and it needs to be expressed in terms of a general–purpose functional approximation method, then the amount of data necessary to obtain a reliable functional expression can become excessive. More often than not, the number of experiments or simulations needed to obtain a functional approximation will become larger than what is necessary to simply find a minimum. Among the most successful methods that try to combine functional approximation with search for optimizers is Bayesian learning, which is described in an extensive literature.

If one wants to avoid the extra work of estimating the complete function f using oracle queries, one approach is to emulate the nonlinear programming methods, using the function values to estimate derivatives. Derivatives are however notoriously difficult to measure using the differences between the values of the function in neighboring points. If the points are too far, then what is being estimated is not the value of the derivative; if too close, then the difference is dominated by noise or numerical errors.

With this in mind, many algorithms for derivative-free optimization have been studied in the literature. Those include methods based on biological or evolutionary analogies for the exploration of the search space; methods using hypercubes or simplexes to bound regions where the minima may be contained; search in the direction of coordinates or using other patterns; trust–region methods using local linear or quadratic approximations to the functions; and others.

C. Methods for derivative–free optimization

Let's revisit some usual direct optimization techniques to try to understand where the barycenter method fits. Recall that the majority of the literature in mathematical optimization deals with problems where the goal to be optimized is formulated as a well–defined mathematical function, of which we can compute derivatives and second derivatives, and perform other useful operations. It is often not the case that the expressions are available. Then we are forced to look for minima (or maxima) using measurements of the value of the goal. Derivative–free optimization often proceeds along the following lines:

- Imitate conventional nonlinear programming methods, approximating derivatives using measured values.
- Estimate the function itself using sampled measurements, and then use various methods on the estimated function as if we were certain that the estimate was correct.
- Specify a recursive algorithm for generating a sequence of guesses of where the minimum might be located, and endeavor to show that the procedure converges to an acceptable answer to the minimization question.

Let us consider each of these alternatives. It need not be argued to an audience familiar with automatic control that finite–differences approximations to derivatives dont often result in useful techniques. Computing derivatives experimentally is not computationally or algorithmically difficult; it is simply useless. The realities of measurement noise and procedural errors will make any approximation prone to errors and in need of extensive filtering, which requires repetitive sampling, and increases the number of measurements needed.

Functional estimates are often based on explicit parameterizations, which may be global or use a patchwork of local functions defined within trust regions. Local parameterizations may be based on linear, quadratic, sigmoidal, Gaussian, or fuzzy–logic type basis functions. Global parameterizations could use familiar power or sinusoidal series. Some of the most successful methods that consider global parameterizations are the Bayesian learning ones. There also exist so-called nonparametric models, where the parameterizations are given implicitly by the assumptions made in generating the estimates and the algorithm that produces the sequence of probes, the dimension of the parameter space being very large, perhaps uncountably so.

Effectiveness of the procedure depends on how well the model class matches the class of functions being estimated. If we have workable prior knowledge about the general shape or mathematical expression of the function being optimized, then each sample will contribute to the knowledge of its overall behavior, and learning of the function itself can proceed economically. On the other had if the goal function does not match well the prior assumptions embedded into the parameterization, learning its overall shape will require extensive probing across all of its domain. It becomes likely that this procedure will be overwork if our goal is simply to find a minimum value of the function, rather than a precise global model for all of its values.

Search algorithms for derivative–free optimization have been engineered using a variety of different principles. Some use coordinate patterns to pick the search directions, or geometrical constructs such as hypercubes or other polytopes to progressively shrink the portion of the search space under consideration. In the Nelder–Mead method, one of the oldest and best known in direct optimization, the polytopes are triangles, tetrahedra, or higher–dimensional simplexes. Another category of algorithms are inspired by the individual or collective behavior of animals or biological phenomena such as recombination of genetic material, mutation, and evolution.

Many of the techniques make use of randomization, and even (almost completely) random searches have been considered. All of them provide good ideas and intuition to help specify the algorithms, however the narrative of how they operate often sounds like a just–so story. It is often not clear that Nature runs those procedures with the goal of finding minima and maxima, nor does the motivation furnish convincing arguments for why the extrema will be reached by the procedure. Proving convergence is a fruit of labor, and may require exacting assumptions on the goal functions.

Its is partly in response to the considerations above that the barycenter method has been developed. Its analysis follows from the equivalence between the batch and recursive formulations in III. While the recursive version is flexible enough to incorporate any search mechanism that is found to have merits for a particular problem, the batch formula provides a closed expression that serves to analyze the properties of the method, in particular its robustness to noise and non-differentiability.

One of the desirable properties of gradient–based methods is that they incorporate naturally the concept that points where the derivative is large should be skipped over quickly, at least in the case of differentiable goal functions. This is not an easy notion to consider in most of the derivative–free methods. The complex version of the barycenter method has the property that points with large values of the derivative are given a low weighting. If they cannot be avoided without explicitly taking derivatives, at least the method can be set up so that tests at high-derivative points will not lead to a big waste of resources during the course of the search.

D. The barycenter method

The barycenter method has a very straightforward rationale. Search points are given an exponential weight, which is large for points where the function has low values and small where the goal function is small. The search points are combined to produce an estimate for the minimum at their center of mass.

The method has equivalent batch and recursive formulations. The equivalence of the formulations provides an algorithmic approach and facilitates mathematical analysis of the properties.

The barycenter method may be used to combine a sequence of test points independently of how the sequence was generated. It is not in opposition to the search algorithms previously studied in the literature — any of them, or more than one of them, can be used to generate the test points. In the same way that different approaches can be combined, the barycenter method is naturally parallelizable.

If no previous knowledge exists, or if one decides not to use the existing methods, then it is reasonable to use a purely random search to generate the sequence of test points recursively. In this case it can be show that the barycenter method produces a sequence of steps that follows a gradient descent pattern, without the need to compute or estimate gradients. Theorem 1 gives a proof of this statement. Theorem 2 shows that the variance of the step size is reduced with respect to the variance of the randomized search, which is an indication of the methods robustness.

The barycenter method has a complex version which is inspired by Feynman's interpretation of quantum mechanics. The advantage of the complex version is that it avoids and discounts tests made at points where the derivative of the goal function is high. Recall that such points do not fulfill the necessary conditions for minimality, however they cannot be a priori excluded from the search if we are restricted to using derivative–free methods. Using a form of destructive interference between nearby points, the complex version of the barycenter method goes for the second best option, which consists in giving lower weight to measurements made at high derivative points, as shown in Theorem 3.

Another advantage is that the method is by construction tolerant to noise, or measurement and numerical errors. Theorem 4 provides approximations for the mean bias and variance of the estimate of the minimum introduced by the presence of noise. The analysis of the methods behavior in the presence of noise is facilitated by the simple expressions for its recursive and batch versions, and by their straightforward equivalence. Besides easing the mathematical analysis, the simplicity of the method makes coding and implementation more transparent.

III. RECURSIVE AND BATCH EXPRESSIONS

The barycenter method consists of searching for a minimizer of a function $f : \mathcal{X} \to \mathbb{R}$ using the formula

$$\hat{x}_n = \frac{\sum_{i=1}^{n} x_i \mathrm{e}^{-\nu f(x_i)}}{\sum_{i=1}^{n} \mathrm{e}^{-\nu f(x_i)}}, \tag{1}$$

which expresses the center of mass of n test points $x_i \in \mathcal{X} \subset \mathbb{R}^{n_x}$ weighted according to the exponential of the value of the function at each point. In this formula $\nu \in \mathbb{R}$ is a positive constant. The point $\hat{x}_n$ will be in the convex hull of the $\{x_i\}$, and we shall assume that the n_x–dimensional search space

$\mathcal{X}$ is convex. The rationale behind it is that points where f is large receive low weight in comparison with those for which f is small.

The equation above is equivalent to the recursive formulas

$$m_n = m_{n-1} + \mathrm{e}^{-\nu f(x_n)} \tag{2}$$

$$\hat{x}_n = \frac{1}{m_n}\left(m_{n-1}\hat{x}_{n-1} + \mathrm{e}^{-\nu f(x_n)} x_n\right). \tag{3}$$

Here $m_0 = 0, \hat{x}_0$ is arbitrary, and x_n is the sequence of test values.

From the point of view of recursive search strategies, it can be useful to pick the sequence of test points x_n as the sum of the barycenter $\hat{x}_{n-1}$ of the previous points and a "curiosity" or exploration term z_n:

$$x_n = \hat{x}_{n-1} + z_n. \tag{4}$$

Then (3) reads

$$\hat{x}_n - \hat{x}_{n-1} = \frac{\mathrm{e}^{-\nu f(x_n)}}{m_{n-1} + \mathrm{e}^{-\nu f(x_n)}} z_n. \tag{5}$$

IV. STATEMENT OF RESULTS

In this section we state 4 results concerning the barycenter method. The proofs and more detailed discussions are available in the arXiv [7].

Theorem 1: If z_n has a Gaussian distribution, the expected value of $\Delta\hat{x}_n = \hat{x}_n - \hat{x}_{n-1}$ is proportional to the average value of the gradient of $f(\hat{x}_{n-1} + z_n)$ in the support of the distribution of z.

The theorem shows that roughly speaking a random search performed in conjunction with the barycenter algorithm follows the direction of the negative average gradient of the function to be minimized, the weighted average being taken over the domain where the search is performed. For a given ν, the step size is essentially given by the variance of z.

Moreover, near a minimum of a locally convex function, the variance of the adjustment step grows less than linearly with the variance of the curiosity; the higher the Hessian and the larger ν is, the smaller the variance. This is a desirable property of the method, because it indicates that the barycenter moves around less than the test points. To state a formula for the variance, define $F_n(z) = \frac{\mathrm{e}^{-\nu f(\hat{x}_{n-1}+z)}}{m_{n-1}+\mathrm{e}^{-\nu f(\hat{x}_{n-1}+z)}}$, and $\bar{F}_n(z) = \frac{m_{n-1}}{m_{n-1}+\mathrm{e}^{-\nu f(\hat{x}_{n-1}+z)}} F_n$, so that $\frac{\partial F}{\partial z} = -\nu \bar{F}\frac{\partial f}{\partial z}$.

Theorem 2: Under the conditions of Theorem 1 and assuming that the variance of z is small, the variance of $\Delta\hat{x}_n$ for $\bar{z} = 0$ near a critical point of $f(x)$ where $\nabla f = 0$ is approximately

$$\mathrm{Var}(\Delta\hat{x}) \approx \Sigma E[F^2] - 2\nu\Sigma^T E\left[F\bar{F}\nabla^2 f\right]\Sigma. \tag{6}$$

The *complex barycenter* is defined term–by–term by the same formula as the barycenter in (1), with a complex exponent ν:

$$\eta_n^\alpha = \frac{\sum_{i=1}^n x_i^\alpha \mathrm{e}^{-\nu f(x_i)}}{\sum_{i=1}^n \mathrm{e}^{-\nu f(x_i)}}, \tag{7}$$

but now our estimate of the extremum point is

$$\hat{x}_n^\alpha = |\eta_n^\alpha|. \tag{8}$$

In these formulas all $x_i \geq 0$. The algorithm is suggested by Feynman's interpretation of quantum electrodynamics [4], [5] and by the stationary phase approximation [2], [6] used in the asymptotic analysis of integrals.

Theorem 3: The expected contribution of measurements made outside of any region where $\nabla f \approx 0$ is discounted by one factor, proportional to ∇f and to the ratio between the complex magnitude of ν and its real part, for each dimension of the search space.

This destructive interference, so to speak, between repeated measurements near points which are *not* candidates for minimizers is the justification for employing complex values of ν.

Oftentimes each measurement of the function f at point x_i is corrupted by noise or experimental errors. In this case we still would like to minimize f, but now using oracle answers $f(x_i) + w_i$. For the purpose of analyzing the effect of noise on the results of the barycenter method, we consider the sequence x_i as given and w_i as an ergodic random process. A more elaborate analysis of the effect of noise on the sequence $\{x_i\}$ itself, which would depend on the recursive search algorithm used to generate the oracle queries, isn't done here.

Define the nominal or "noise–free" values $\bar{m} = \sum_{i=1}^n \mathrm{e}^{-\nu f(x_i)}$ and $\bar{\eta} = \sum_{i=1}^n x_i \mathrm{e}^{-\nu f(x_i)}/\bar{m}$, and also the scalar quantity $\bar{\bar{m}} = \sum_{i=1}^n \mathrm{e}^{-2\nu f(x_i)}$, the vector quantity $\bar{\bar{\eta}} = \sum_{i=1}^n x_i \mathrm{e}^{-2\nu f(x_i)}/\bar{\bar{m}}$, and the matrix quantity $\breve{\eta} = \sum_{i=1}^n x_i x_i^T \mathrm{e}^{-2\nu f(x_i)}/\bar{\bar{m}}$.

Theorem 4: Assuming that σ is small, under the circumstances above the mean and variance of η can be expressed approximately as follows:

$$E[\eta] \approx \bar{\eta} + \frac{\bar{\bar{m}}}{\bar{m}^2}(\bar{\eta} - \bar{\bar{\eta}})\nu^2\sigma^2 \text{ and} \tag{9}$$

$$\mathrm{Var}[\eta] \approx \frac{\bar{\bar{m}}}{\bar{m}^2}(\bar{\eta}\bar{\eta}^T - \bar{\eta}\bar{\bar{\eta}} - \bar{\bar{\eta}}\bar{\eta} + \breve{\eta})\nu^2\sigma^2. \tag{10}$$

What we learn from somewhat involved formulas (9) and (10) is that noise or measurement errors generate a bias and a variance in the barycenter, both proportional in 1st approximation to the variance of the noise. These effects conspire to pull the barycenter away from the looked–after minimum of the function. However one may derive a measure of comfort in that the unwelcome errors tend to zero as the noise becomes smaller. Theorem 4 indicates that there is little reason to fear that the method breaks down under moderate measurement or computing errors.

The situation where the (unknown) function to be optimized is not smooth can be studied as a particular case of the optimization of $f(x) + w$, where $f(\cdot)$ itself is smooth, and $w(x)$ is the difference between the function under consideration and its smooth approximation, plus a noise or error parcel when applicable. In this case the assumption that w_i is uncorrelated with x_i can be objected to. On the other hand any function can be approximated with arbitrary precision by a smooth function, making the analysis reasonable in the practical case when the approximation error is overshadowed by measurement or numerical errors.

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.

V. ILLUSTRATIVE SIMULATIONS

In this section we will choose z with mean $\xi\,(\hat{x}_n - \hat{x}_{n-1})$ and variance σ^2. The idea is that the random term is responsible for the rate of change, or acceleration, of the search process. The factor $0 < \xi < 1$ is chosen to dampen oscillations and prevent instability.

Figure 1 depicts the functional values over number of evaluations of a search for the minimum of the Rosenbrock banana function $f(x) = 100\left(x^2 - y\right)^2 + (1-x)^2$ using the randomized barycenter method. Figure 2 shows the test points in search space and the evolution of the barycenter, starting from the initial guess and randomly sliding down the level curves towards the minimum at $(1,1)$. The randomized barycenter search parameters were chosen $\nu = 4, \xi = 0.6$, and the variance σ starts at 2 and decreases linearly to .4 over the 80 sample tests.

The instance of the search procedure is typical of many tests performed, although of course the results of any nondeterministic search may vary, and could conceivably even fail for a unlucky finite number of samples. We have not made any effort to optimize the search, but the results are roughly speaking comparable to the most effective methods tested in, for example, [3], page 11.

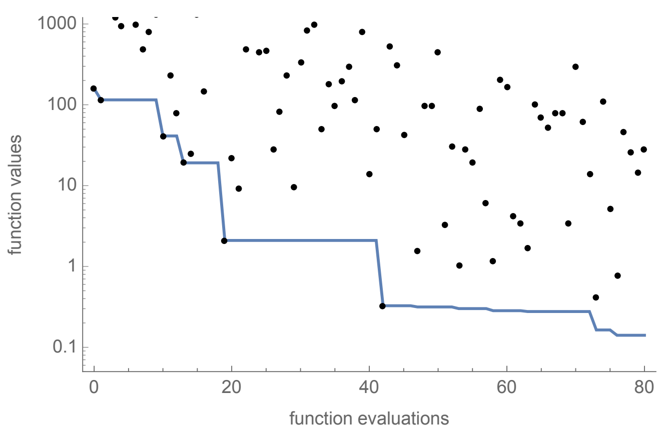

Fig. 1. Points show values of the banana function at the test points x, line shows function values at the estimates $\hat{x}$.

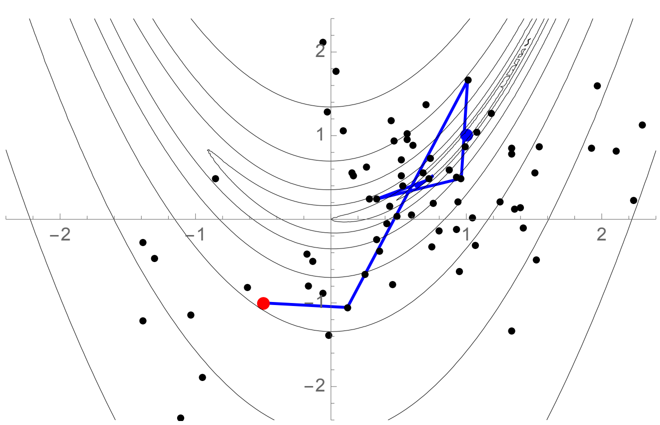

Fig. 2. Points show location of tests, line shows barycenter of previous points across level curves of banana function.

Figure 1 depicts the functional values over number of evaluations of a search for the minimum of the Rosenbrock banana function $f(x) = 100\left(x^2 - y\right)^2 + (1-x)^2$ using the randomized barycenter method. Figure 2 shows the test points in search space and the evolution of the barycenter, starting from the initial guess and randomly sliding down the level curves towards the minimum at $(1,1)$. The randomized barycenter search parameters were chosen $\nu = 4, \xi = 0.6$, and the variance σ starts at 2 and decreases linearly to .4 over the 80 sample tests.

The instance of the search procedure is typical of many tests performed, although of course the results of any nondeterministic search may vary, and could conceivably even fail for a unlucky finite number of samples. We have not made a big effort to optimize the search, but the results are roughly speaking comparable to the most effective methods tested in, for example, [3], page 11.

Another test, shown analogously in Figures 3 and 4, concerns the perturbed quadratic function

$$10x^2\left(1 + \frac{75}{100}\frac{\cos(70x)}{12}\right) + \frac{\cos(100x)^2}{24} + 2y^2\left(1 + \frac{75}{100}\frac{\cos(70y)}{12}\right) + \frac{\cos(100y)^2}{24} + 4xy,$$

also mentioned in [3]. Although minimization of the function with high–frequency sinusoidal perturbations poses challenges to some derivative–free algorithms, the barycenter method performs comparably to the most effective of them. The parameters for the test that involved 100 function value estimations were $\nu = 1, \xi = 8/10$, and variance σ starting at 24/10 and decreasing with 966/1000 at each sample test.

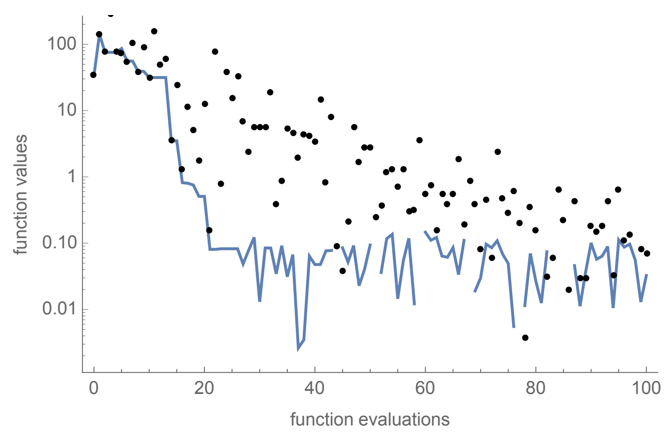

Fig. 3. Values of the perturbed quadratic function at x and $\hat{x}$. Values missing from log plot are negative.

Figures 5 and 6 illustrate the barycenter search for the minimum of the "canoe" function $(1 - e^{||x||^2})\max(||x - c||^2, ||x-d||^2)$, with $c = -d = [30, 40]^T$. This function was introduced as a benchmark to test mesh adaptive direct search algorithms in page 209 of [1], as it might present a challenge to derivative–based optimization methods (both 1st and 2nd order) and also to generalized pattern search

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.

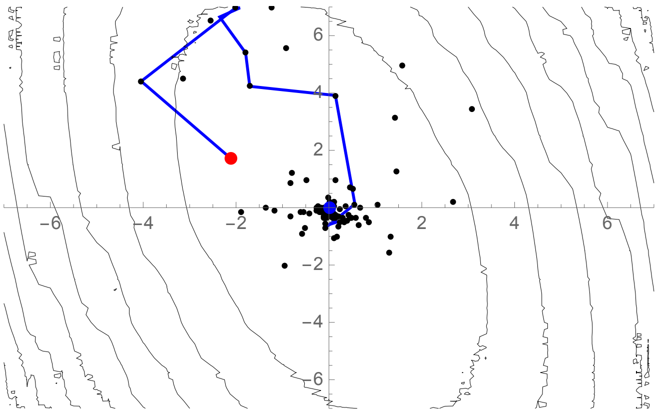

Fig. 4. Evolution of search for minimum of perturbed quadratic function.

(GPS) because of lack of differentiability. The derivative–free method under consideration is applicable to non–differentiable functions, and gives satisfactory results with parameters $\nu = 8/10, \xi = 9/10$, and variance σ starts at 1 and decreases with $(982/1000)^n$ over 300 sample tests.

In the tests, the barycenter method parameters were chosen by the author with basis on aesthetic and didactic considerations, without the benefit of graduate student descent. Notice that some searches are nonmonotonic. A reliably monotonic search would be a sure indicator of lack of robustness to model noise or variability. Simulations, performed with exact specified values in Mathematica™, were very fast.

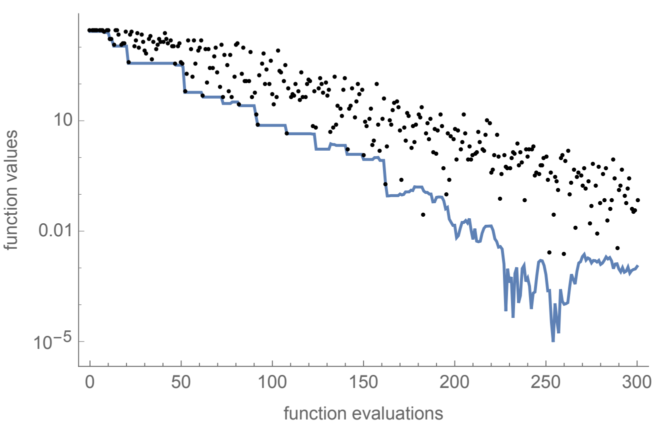

Fig. 5. Canoe function values at test points and estimates.

VI. CONCLUSIONS AND FUTURE WORKS

If one thing has become clear to the reader of this paper, it is that the author likes to write about the barycenter method. But we have reached the page limit and must reluctantly bid farewell with a brief conclusion. The theoretical properties of the method, stated in Section IV and demonstrated in [7], have been illustrated via simulations. The method shows itself to be robust and have acceptable performance in 2–dimensional benchmark problems. The barycenter method is easy to modify. Its performance could be optimized by using anything besides the purely random search we employed,

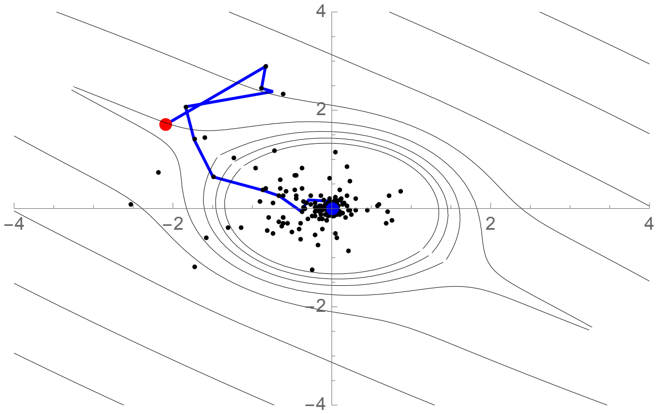

Fig. 6. Evolution of tests across level curves of canoe function.

its properties being guaranteed by the equivalence between the recursive and batch formulas in Section III. It is also naturally parallelizable. Tests of a parallel version, as well as of the complex version presented in equations (7)– (8) with properties given in Theorem 3, remain to be done. The method is already in use for practical applications along the lines discussed in [12] and [11].

VII. ACKNOWLEDGMENTS

The author is grateful for the encouragement and questioning of his students. This work is made possible by readers like you.

REFERENCES

[1] C. Audet and J. J. E. Dennis, "Mesh adaptive direct search algorithms for constrained optimization," *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 188–217, 2006. [Online]. Available: https://doi.org/10.1137/040603371

[2] N. Bleistein and R. Handelsman, *Asymptotic Expansions of Integrals*. Dover Publications, 1975.

[3] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative–free optimization*, ser. MPS/SIAM Series on Optimization. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2009, vol. 8.

[4] R. Feynman, *QED: the Strange Theory of Light and Matter*. Princeton, 1985.

[5] R. Feynman and A. Hibbs, *Quantum Mechanics and Path Integrals: Emended Edition*. Dover, 2012.

[6] L. Hörmander, *The Analysis of Linear Partial Differential Operators I: Distribution Theory and Fourier Analysis*, ser. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2012.

[7] F. M. Pait, "The Barycenter Method for Direct Optimization," *ArXiv e-prints*, Jan. 2018. [Online]. Available: https://arxiv.org/abs/1801.10533

[8] F. Pait, "Reading Wiener in Rio," in *IEEE Conference on Norbert Wiener in the 21st Century*. IEEE, 2014, pp. 1–4.

[9] F. Pait and D. Colón, "A barycenter method for direct optimization," in *SIAM Conference on Optimization*, San Diego, CA, USA, May 2014.

[10] F. Pait and R. Romano, "Direct adaptive control, direct optimization," in *SIAM Annual Meeting*, Boston, Massachusetts, USA, July 2016.

[11] R. A. Romano and F. Pait, "Direct filter tuning and optimization in multivariable identification," in *Proceedings of the 53rd IEEE Conference on Decision and Control*, Los Angeles, USA, Dec 2014, pp. 1798–1803.

[12] ——, "Matchable-observable linear models and direct filter tuning: An approach to multivariable identification," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2180–2193, May 2017.

[13] M. Wright, "Direct search methods: Once scorned, now respectable," in *Numerical Analysis 1995: Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis*, D. F. Griffiths and G. A. Watson, Eds. Addison Wesley Longman, 1995, pp. 191–208.

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.