PLOS ONE

RESEARCH ARTICLE

Optimization by Adaptive Stochastic Descent

Cliff C. Kerr[1,2,3]*, Salvador Dura-Bernal[4], Tomasz G. Smolinski[5], George L. Chadderdon[2], David P. Wilson[2,3]

1 Complex Systems Group, School of Physics, University of Sydney, Sydney, NSW, Australia, **2** Optima Consortium for Decision Science, Melbourne, VIC, Australia, **3** Centre for Population Health, Burnet Institute, Melbourne, VIC, Australia, **4** Department of Physiology and Pharmacology, SUNY Downstate Medical Center, Brooklyn, NY, United States of America, **5** Department of Computer and Information Sciences, Delaware State University, Dover, DE, United States of America

* cliff@thekerrlab.com

Citation: Kerr CC, Dura-Bernal S, Smolinski TG, Chadderdon GL, Wilson DP (2018) Optimization by Adaptive Stochastic Descent. PLoS ONE 13(3): e0192944. https://doi.org/10.1371/journal.pone.0192944

Editor: Lars Kaderali, Universitatsmedizin Greifswald, GERMANY

Received: May 21, 2017

Accepted: January 31, 2018

Published: March 16, 2018

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: C.C.K. was supported by the Australian Research Council (ARC) Discovery Early Career Researcher Award DE140101375. C.C.K. and S.D.B. were supported by the Defense Advanced Research Projects Agency (DARPA) Contract N66001-10-C-2008. C.C.K., G.L.C., and D.P.W. were supported by World Bank Assignment 1045478. T.G.S. was supported by National Institutes of Health grants NCRR 5P20RR016472-

Abstract

When standard optimization methods fail to find a satisfactory solution for a parameter fitting problem, a tempting recourse is to adjust parameters manually. While tedious, this approach can be surprisingly powerful in terms of achieving optimal or near-optimal solutions. This paper outlines an optimization algorithm, Adaptive Stochastic Descent (ASD), that has been designed to replicate the essential aspects of manual parameter fitting in an automated way. Specifically, ASD uses simple principles to form probabilistic assumptions about (a) which parameters have the greatest effect on the objective function, and (b) optimal step sizes for each parameter. We show that for a certain class of optimization problems (namely, those with a moderate to large number of scalar parameter dimensions, especially if some dimensions are more important than others), ASD is capable of minimizing the objective function with far fewer function evaluations than classic optimization methods, such as the Nelder-Mead nonlinear simplex, Levenberg-Marquardt gradient descent, simulated annealing, and genetic algorithms. As a case study, we show that ASD outperforms standard algorithms when used to determine how resources should be allocated in order to minimize new HIV infections in Swaziland.

Introduction

Consider a human $\mathcal{H}$ who is attempting to minimize a nonlinear objective function, $E = f(\mathbf{x})$, by manually adjusting parameters in the vector $\mathbf{x}$. $\mathcal{H}$ typically begins with a uniform prior regarding which parameters to vary, and chooses step sizes that are a fixed fraction (*e.g.*, 10%) of the initial parameter values. $\mathcal{H}$ will then pseudorandomly choose one or more parameters to adjust. Every time a parameter x_i is found to reduce E, the probability that $\mathcal{H}$ will select x_i in the future increases; conversely, if changes in x_i are not found to improve E, the probability that $\mathcal{H}$ will select x_i decreases (formally, $\mathcal{H}$ forms "hunches" about which parameters are "good"). $\mathcal{H}$ also adaptively adjusts the step size based on the information $\mathcal{H}$ obtains about the curvature of parameter space with respect to each dimension (*e.g.*, if $\Delta E/\Delta x_i \approx$ const. over multiple iterations, $\mathcal{H}$ will increase the step size). Despite its drawbacks, the adaptive nature of manual parameter fitting makes it a remarkably powerful method.

12 and NIGMS 8P20GM103446-12, and the National Science Foundation grants EPSCoR-0814251 and HRD-1242067. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Thus, despite the smörgåsbord of available automated optimization algorithms, manual fitting of parameters remains a familiar bane of researchers (*e.g.*, [1, 2]), especially in cases where evaluations of the objective function are computationally intensive, such as climate models [3], neuronal network models [4–6], or detailed epidemiological models [7]. However, it is difficult to estimate how commonly manual parameter fitting is performed, since authors often do not explicitly mention its use (*e.g.*, [8]).

In many types of optimization problems, it is more important to need only a small number of function evaluations to find a reasonable local minimum than it is to find the global minimum [9, 10]. Indeed, the latter may be ill-defined given the large uncertainties that are often present when models of complex systems are fitted to empirical data, as in the citations listed above.

With the increasing availability of high-performance computers and clusters [11], easily parallelizable optimization methods such as evolutionary algorithms (where different individuals can be run on different cores) and Monte Carlo methods (where different initializations can be run on different cores) have a notable advantage for certain types of problems. The common theme in these algorithms is the ability to use a different random seed for each parallel instance. However, as the size of parameter space increases, the advantage of this approach is diluted: whereas a 3- or even 5-dimensional parameter space may be reasonably densely sampled by a Monte Carlo initialization, a 20- or 100-dimensional space cannot. This is because parameter space grows exponentially with an increasing number of dimensions, whereas parallelization increases sampling rates linearly.

In high-dimensional parameter spaces, it is unlikely that all parameters contribute equally to the objective function. Identifying those that contribute more, thereby allowing computational resources to be focused on them, has the potential to significantly reduce the total number of function evaluations required. Despite humans' limited capacity to implement Bayesian-optimal strategies [12, 13], we speculate that this adaptive approach to both parameter selection and step size is the key reason why manual parameter fitting can be highly effective.

The aim of this paper is to present a random search algorithm, Adaptive Stochastic Descent (ASD), that was inspired by manual parameter fitting and is intended to be a simpler alternative to more complex optimization methods. ASD is most applicable to optimization problems with more than approximately 5 dimensions—*i.e.*, large enough so that performing function evaluations across all dimensions is inefficient. ASD forms the core of the optimization algorithm used in the Optima suite of tools (optimamodel.com), most notably Optima HIV [14], and as such has already been extensively used and validated for calibrating epidemic models and determining optimal resource allocations [15–22]. The algorithm has also been applied to fitting a spiking neuronal network model to electrophysiology data from individual rat brains [23], and has been used in ongoing work calibrating a neural field model to reproduces impulse responses in sleep EEG data [24]. Here we also compare ASD to traditional algorithms using two classic optimization test problems, and provide an extended case study on optimally allocating resources for HIV interventions using a detailed model of Swaziland's HIV epidemic.

ASD is provided under the open-source MIT License. Python and MATLAB versions are available for download from thekerrlab.com/asd or via GitHub at github.com/thekerrlab/asd.

Basic algorithm

Consider an objective function $E = f(\mathbf{x})$, where E is the scalar error (or other quantity) to be minimized (or maximized) and $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ is an n-element vector of parameters. There

are $2n$ possible directions j to step in: an increase or decrease in the value of each parameter. Associated with each parameter x_i are (a) two initial step sizes: $s_j = s_i^+$ or s_i^-, which define the step size in the directions of increasing or decreasing x_i, respectively (*i.e.*, $s_i^+ > 0$ and $s_i^- < 0$); and (b) two initial probabilities: $p_j = p_i^+$ or p_i^-, which define the likelihood of selecting direction j (for a uniform prior, $p_j = 1/2n$—satisfying the requirement that $\sum \mathbf{p} = \sum_{j=1}^{2n} p_j = 1$). Thus, the vectors **s** and **p** have length $2n$.

At each step k, the algorithm maps a random variable $\alpha \in (0, 1)$ onto **p**, thereby choosing a direction $j \in (1...2n)$ and a corresponding parameter $i = \lceil j/2 \rceil \in (1...n)$, where $\lceil \cdot \rceil$ denotes the ceiling operator. The algorithm then evaluates

$$E_k^{\pm} = f(\mathbf{x} + \delta(i)), \tag{1}$$

where $\delta(i)$ is an n-element vector such that $\delta_i = s_j$ and 0 otherwise. Then:

1. If $E_k^{\pm} < E_{k-1}$:
   a. The new parameter value is adopted: $x_i \rightarrow x_i + s_j$;
   b. The error is updated: $E_k \rightarrow E_k^{\pm}$;
   c. s_j is increased: $s_j \rightarrow s_j \cdot s_{inc}$ (where $s_{inc} > 1$);
   d. p_j is increased: $p_j \rightarrow p_j \cdot p_{inc}$ (where $p_{inc} > 1$), and **p** is renormalized such that $\sum \mathbf{p} = 1$.
2. Otherwise:
   a. The parameter vector **x** and error E are not changed;
   b. s_j is decreased: $s_j \rightarrow s_j / s_{dec}$ (where $s_{dec} > 1$);
   c. p_j is decreased: $p_j \rightarrow p_j / p_{dec}$ (where $p_{dec} > 1$), and **p** is renormalized as above.

The algorithm thus has four metaparameters: s_{inc}, s_{dec}, p_{inc}, and p_{dec}. In general, the smoother and more linear the objective function is, the larger the learning rates should be; the choice of $s_{inc} = s_{dec} = p_{inc} = p_{dec} = 2$ has been found to work well for both simple test cases as well as optimizing complex epidemiological models, although values from approximately 1.2 to 3 were found to have broadly similar performance. In addition to these metaparameters, three initial value vectors need to be specified: the initial parameter vector $\mathbf{x_0}$, step sizes **s** (which in general can be initialized as a fixed fraction of the corresponding initial parameter value, *e.g.* 20%, unless the initial value is zero), and probabilities **p** (where typically $p_j = 1/2n$ suffices for an n-parameter problem).

By modifying **s** and **p** after each iteration, the algorithm learns which directions are most effective to step in and by how much (in the sense that it updates its choices of **s** and **p** by their initial states depending on accumulated evidence). This, combined with the stochastic choice of which parameters to modify on each iteration, resembles the way in which humans (imperfectly) perform Bayesian decision-making in situations such as N-armed bandit problems [13].

The criteria for terminating the algorithm can be specified in the same way as for traditional optimization algorithms. The most common choices for termination are when changes in parameter values (*i.e.*, Δx) and/or improvements in the objective function (*i.e.*, ΔE) are below a given absolute or relative threshold (*e.g.*, 10^{-6}) for a given number of iterations (*e.g.*, 50).

Extensions to the algorithm

This section describes several modifications to the basic algorithm that may make it more suitable for a broader range of optimization problems.

To circumvent the problem of local minima, the method may be used with Monte Carlo initialization [25]. In this case, the ASD algorithm is repeated multiple times (typically, $10^1 - 10^3$) with pseudorandom choices of $\mathbf{x_0}$. The use of multiple starting points helps achieve the balance between "exploration and exploitation" (exploring the entire feasible region of parameter space versus exploring the most promising subregions), which is critical for efficient global search [26]. This is the approach used in Optima HIV, where typically up to 10 Monte Carlo initializations are used. When we applied ASD to each of the 54 different Optima HIV models that correspond to the countries comprising 80% of the global burden of HIV [27], we found that a single initialization converged on the global optimum for 38 (70%) of the models, while 10 initializations converged on the global optimum for all but one model (98%).

Another approach for circumventing the problem of local minima is a probabilistic step acceptance process, similar to that used in simulated annealing or the Metropolis-Hastings algorithm [28]. Here, instead of always performing step 2 of the algorithm if the new iteration does not reduce error, step 1 is performed with nonzero acceptance ratio ρ, where ρ is a function of the change in error; *e.g.*, $\rho \propto E_{k-1}/E_k^{\pm}$. Although the parameter set resulting from each iteration can be kept, as in a Metropolis-Hastings algorithm, the value of doing so is limited since the asymptotic distribution of parameter sets is not guaranteed to reach a stationary distribution, due to the adaptive method for choosing which parameters to vary. Instead, it would suffice to keep two parameter sets, the current one and the best one. As a simpler alternative to implementing a Metropolis-Hastings approach, rather than always reducing the step size if the new iteration does not reduce the error, the step size could have a nonzero probability of increasing, potentially allowing the algorithm to escape local minima.

Note that in the limit of infinite iterations, the basic ASD algorithm will not almost surely converge to the global optimum, since the step size will asymptotically converge to zero if the algorithm is in a location of parameter space such that its step size in all dimensions is smaller than the size of the local minimum's basin of attraction. However, the algorithm will almost surely converge to the global optimum if probabilistic step acceptance is implemented (or if step sizes have nonzero probability of increasing when an evaluation does not result in improvement). Formally, multiple initializations do not suffice to almost surely converge unless they are infinite in number. However, in practice, depending on the smoothness and monotonicity of the objective function, multiple initializations typically allow the exploration of global parameter space (and thus convergence) more efficiently than probabilistic step acceptance.

In some cases it may be desirable to allow assumptions about the scale or relative importance of parameters to be incorporated, in which case the assumptions of uniform priors $\mathbf{p}$ and uniform initial step sizes $\mathbf{s}$ can easily be relaxed. However, due to the adaptive nature of the algorithm, even silly initial choices of $\mathbf{p}$ and $\mathbf{s}$ will be corrected, as long as all p_j and s_j are nonzero. In general, choices of s_j or p_j that are too small are more problematic than ones that too large, since the latter will be corrected with each iteration that fails to improve the objective function.

To incorporate additional information about the change in the objective function, rather than updating the probability p_j by a fixed amount after each successful iteration, the change in p_j (Δp_j) can be a function of the change in the objective function E (ΔE), such that a larger ΔE results in a larger Δp_j, as in simultaneous perturbation stochastic approximation [29]. However, since the expected change in E at step k is proportional to both $|E_k - \min(E)|$ and the ratio of the step size to the characteristic scale of each parameter, and since in general neither of these quantities are known, the constant of proportionality between Δp_j and ΔE cannot typically be estimated *a priori*. One can partially circumvent this problem by comparing the

current ΔE to its previous values; however, more weight would need to be given to more recent values, since ΔE tends to decrease as the algorithm converges on a solution.

The assumption of local linearity can be relaxed by varying multiple parameters on a single iteration. However, assuming a separate probability is stored for each parameter combination, this reduces the learning rate; for an n-parameter problem, modifying a single parameter at each iteration results in a learning rate of $1/2n$ on average for each parameter; in the limit where all possible combinations of parameters are considered, the learning rate would be $1/2^{2n}$. While manageable for small numbers of parameters (*e.g.*, $\leq$4), this quickly becomes intractable as the number of parameters grows. Conversely, if multiple parameters are modified simultaneously, the probabilities of all modified parameters could be updated simultaneously; this approach is likely to be most effective in very high-dimensional systems where the function E is nearly flat with respect to many of the dimensions, in which case varying parameters one by one may be time-consuming. The superior performance of simulated annealing compared to ASD for small numbers of function evaluations in the 10-parameter Rosenbrock's valley problem discussed below is likely due to this effect.

Finally, although only loosely inspired by Bayesian principles, the ASD algorithm could potentially be adapted to implement them more rigorously. While a more formal Bayesian approach may be desirable in certain situations, in general it is difficult to determine whether new information should be used to update the existing distribution, or whether the system is in a sufficiently dissimilar part of the parameter space that information from much earlier iterations is no longer relevant. Nonetheless, for certain problems, additional capacity for adaptation may be beneficial. For example, as shown below, the basic implementation of ASD described above performs poorly in cases where the objective function is dominated by nonlinear parameter interactions, as in the classic version of Rosenbrock's valley; for this particular problem, an algorithm that was capable of learning nonlinear parameter combinations would be far more efficient.

Comparison to other optimization methods

Here we compare ASD to four standard optimization methods: the Nelder-Mead nonlinear simplex algorithm [30], Levenberg-Marquardt gradient descent [31], simulated annealing [32], and a genetic algorithm [33]. All methods were implemented in MATLAB 2012b (The MathWorks, Nantick, MA), via the Optimization Toolbox functions "fminsearch", "lsqnonlin", "simulannealbnd", and "ga", respectively. These algorithms are also available in the "optimize" module of the Python package SciPy via "minimize(method = 'Nelder-Mead')", "leastsq()", and "anneal()", respectively (genetic algorithms are not available in SciPy, but are available via other modules). We chose these methods to compare against since, like ASD, they have relatively simple implementations and relatively few metaparameters that need to be specified.

For ASD, we used metaparameters $s_{inc} = p_{inc} = s_{dec} = p_{dec} = 2$, initial step sizes s_j of 20% of the parameter values in $\mathbf{x_0}$ (which are given below; the step size for any parameter with an initial value of 0 is the mean of the other step sizes), and uniform initial probabilities p_j (*i.e.*, $1/2n$ for an n-dimensional problem). MATLAB's default metaparameters were used for the other four algorithms, except that the initial temperature of the simulated annealing algorithm was set to be equal to $10 \cdot \langle|\mathbf{x_0}|\rangle$ following manual exploration of metaparameter space, since the default choice of 100 did not generalize well across problems of different scales. Indeed, one of the major disadvantages of this type of algorithm is its sensitivity to the values of its metaparameters [34].

To test this suite of algorithms, we used original and modified versions of the two classic optimization problems used for illustrating the simplex algorithm [30]:

1. Rosenbrock's parabolic valley (two-dimensional):

$$E = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \tag{2}$$

with the starting point at $\mathbf{x} = (-1.2, 1)$. The optimum is at $\mathbf{x} = (1, 1)$.

2. A modified 10-dimensional version of Rosenbrock's valley, with the functional form as given in Eq 2, but with a 10-element parameter vector $\mathbf{x}$; the remaining 8 parameters do not contribute to the objective function. The starting point is at $\mathbf{x} = (1.5, -1.5, 0, 0 \ldots 0)$. The optimum is at $\mathbf{x} = (1, 1, \omega_1 \ldots \omega_8)$, where $\omega_1 \ldots \omega_8$ can be any real numbers.

3. A 4-dimensional Powell's quartic function, modified to be N-dimensional:

$$E = \sum((\mathbf{x_a} + 10\mathbf{x_b})^2 + 5(\mathbf{x_c} - \mathbf{x_d})^2 + (\mathbf{x_b} - 2\mathbf{x_c})^4 + 10(\mathbf{x_a} - \mathbf{x_d})^4), \tag{3}$$

where $\mathbf{x_q}$ is a vector of length $N/4$ (and note that vector operations are performed pointwise). The starting point is at $\mathbf{x_a} = (\overline{3})$, $\mathbf{x_b} = (\overline{-1})$, $\mathbf{x_c} = (\overline{0})$, and $\mathbf{x_d} = (\overline{1})$, where each component is repeated $N/4$ times. The optimum is at $\mathbf{x} = (\mathbf{x_a}, \mathbf{x_b}, \mathbf{x_c}, \mathbf{x_d}) = (0, 0, 0, \ldots 0)$. For example, if $N = 4$ (as in the original), then $\mathbf{x_0} = (3, -1, 0, 1)$ and $\mathbf{x}_{\text{opt}} = (0, 0, 0, 0)$; if $N = 8$, then $\mathbf{x_0} = (3, 3, -1, -1, 0, 0, 1, 1)$ and $\mathbf{x}_{\text{opt}} = (0, 0, 0, 0, 0, 0, 0, 0)$. Here, we used 4, 12, 20, and 100-dimensional versions of Powell's function.

The results from applying each of these algorithms to each of the three test problems is shown in Fig 1. For the stochastic algorithms (ASD, simulated annealing, and genetic algorithms), the interval shown represents the interquartile range for 40 different random seeds. For most test problems and iterations, these interquartile ranges did not overlap, suggesting that the intrinsic differences between the algorithms are more important than their stochastic components.

As shown in Fig 1, for the two-dimensional optimization problem, the nonlinear simplex method is most efficient, with all other algorithms requiring considerably more function evaluations to obtain the same error. Notably, after the initial descent, ASD was especially *inefficient*, since its assumption of local linearity is violated by the shallow, curved valley (if this assumption were relaxed, as described above, then ASD's performance on this problem would be significantly improved). With the modified 10-dimensional version of Rosenbrock's valley, ASD is the most efficient algorithm over most of the first several hundred function evaluations, as shown in Fig 2 for a single random seed. For small numbers of iterations (<30), for this particular seed, simulated annealing was by far the most efficient algorithm, reducing the error by a remarkable 98% after just 4 function evaluations. However, this algorithm became mired near the point (1.5, 2.4), far from the minimum of (1, 1), and did not significantly reduce the error beyond the first 20 function evaluations. After 50 function evaluations, ASD had reduced the error by a median of 99.9%, compared to 99.7% for simulated annealing, 96% for the Levenberg-Marquardt method, 82% for the nonlinear simplex method, and 0% for the genetic algorithm. Similarly, ASD reduced the error by 99.99% after 70 function evaluations; in comparison, the next best algorithm (the simplex method) required 220 function evaluations to reach the same error level.

During the descent into the shallow curved valley (comprising $\sim$99.9% of the total error), the most efficient algorithms were ASD and simulated annealing; within the valley (the remaining $\sim$0.1% of the total error), the simplex algorithm was by far the most efficient. Hence, these examples illustrate that in optimization problems where some parameters are significantly more important than others, ASD has significant advantages. In contrast, for problems in which all parameters have equal importance, as in the original Rosenbrock's valley problem, other algorithms have superior performance.

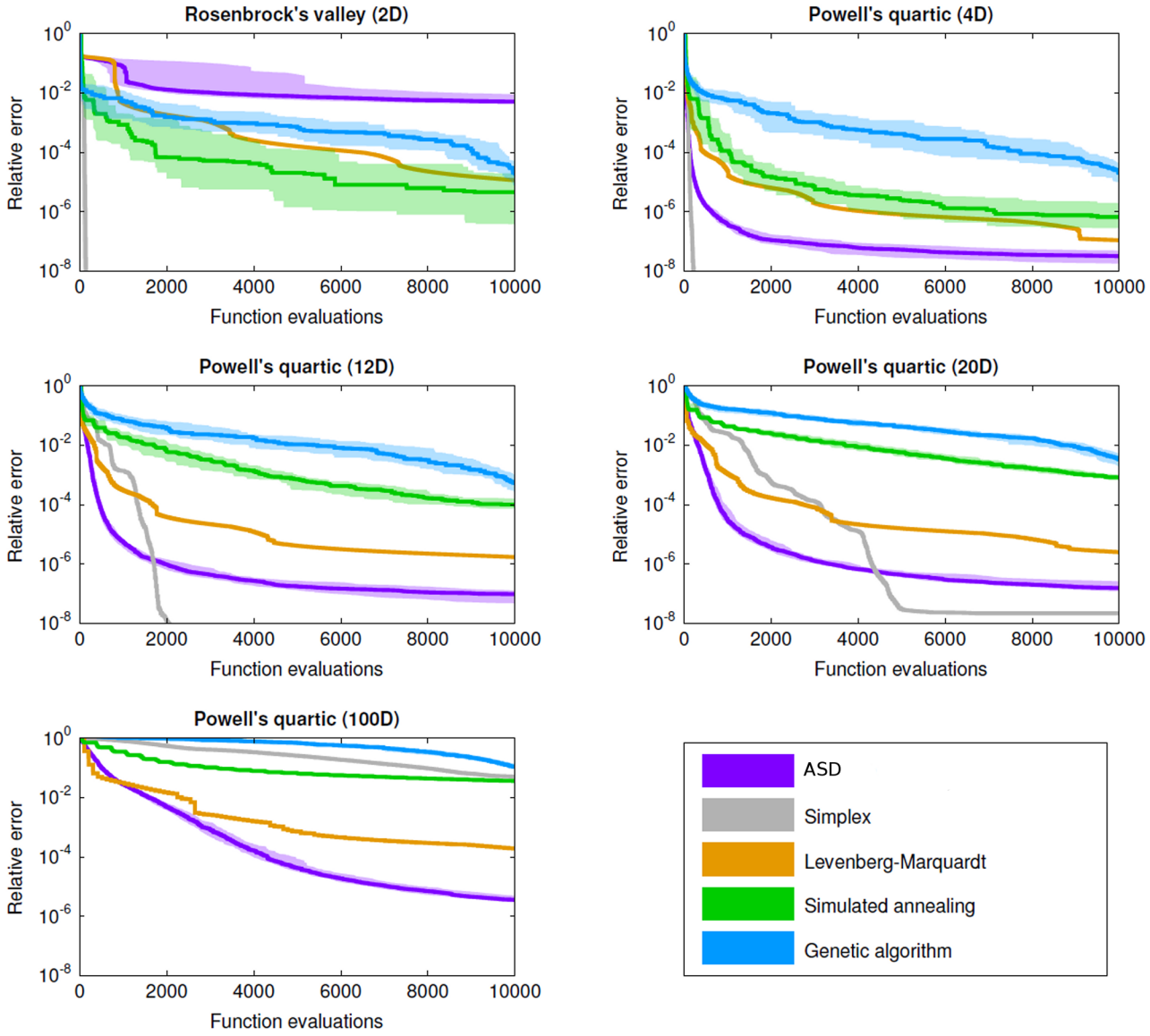

Fig 1. Performance of ASD compared to standard nonlinear optimization algorithms. The four algorithms used are Nelder-Mead nonlinear simplex, Levenberg-Marquardt gradient descent, simulated annealing, and genetic algorithms. The x-axis shows the number of individual function evaluations, while the y-axis shows the error relative to the starting point. Standard methods—especially the simplex method—are most efficient for low-dimensional problems (*e.g.*, Rosenbrock's valley), in many cases ASD is the most efficient algorithm for high-dimensional parameter spaces (*e.g.*, the 100-dimensional version of Powell's quartic function). For the stochastic methods (ASD, simulated annealing, and the genetic algorithm), the shaded regions show the interquartile range for 40 different random seeds.

https://doi.org/10.1371/journal.pone.0192944.g001

For the 4-dimensional Powell's quartic function, the nonlinear simplex method was again the most efficient, followed by ASD. For the 12- and 20-dimensional version, ASD was most efficient for 60–1700 and 250–4400 function evaluations respectively (corresponding to roughly 99.9999% of the total error at the upper limit in each case), after which the simplex method was most efficient. For the 100-dimensional version, the Levenberg-Marquardt method was most efficient for the first 1000 function evaluations (corresponding to 97% of the total error), but ASD was the most efficient algorithm for larger numbers of function evaluations. In practice, algorithms are not run for a fixed number of function evaluations, but rather

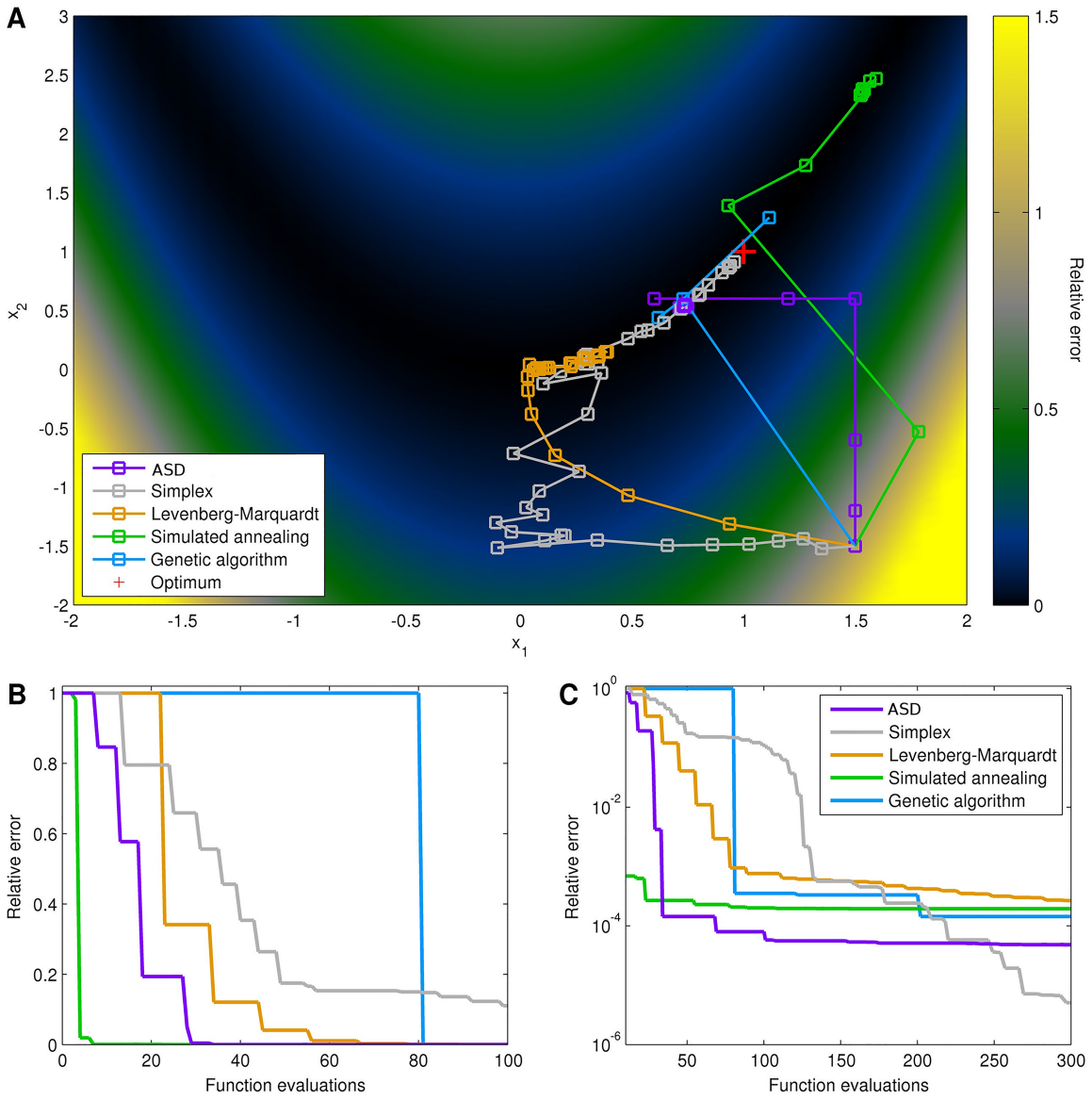

Fig 2. Optimization of the 10-dimensional version of Rosenbrock's valley. (A) Trajectories of each optimization method starting up to 300 function evaluations from the starting point (1.5, −1.5); each iteration is shown with a square, but note that multiple function evaluations may occur at each iteration. Color shows error relative to starting point. Note the locally linear steps of ASD that rapidly adapt in size. (B) Relative error of each method for the first 100 function evaluations, showing the initial stage of the algorithms. (C) Relative error for the first 300 function evaluations, showing the asymptotic stage of the algorithms.

https://doi.org/10.1371/journal.pone.0192944.g002

until they satisfy a given stopping criterion, which is usually defined in terms of the change in the relative or absolute error. Specific choices for these criteria depend on the problem at hand, but for illustrative absolute error tolerances of 99.9% or 99.99%, ASD was the most or equal-most efficient for all cases except the 2D version of Rosenbrock's valley.

The five optimization methods discussed here employ very different parameter update strategies, as shown strikingly in Fig 3. The approach used in ASD is most similar to the Levenberg-Marquardt method, with the exception that the rate of convergence of the former *increases* over time (due to its adaptive step size), whereas for the latter, and for other algorithms, it decreases (as expected from Donsker's theorem [35]). In the example shown here

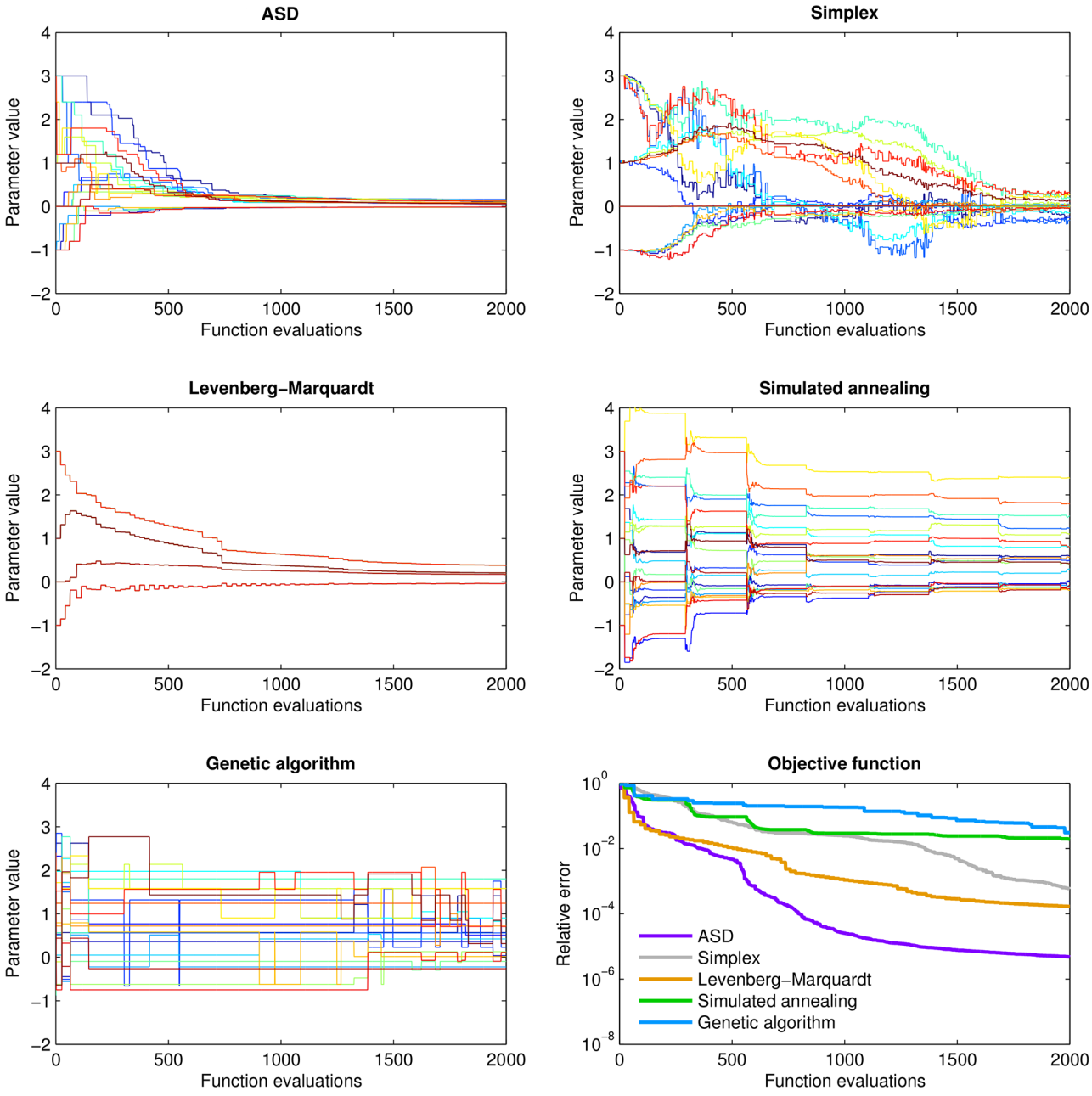

Fig 3. Demonstration of parameter update strategies for each algorithm applied to a 20-dimensional Powell's quartic function. Each plot has 20 lines, showing the value of each parameter after each function evaluation. The optimum is at (0, 0, 0, . . . 0), corresponding to all 20 lines converging to 0. The error relative to the starting point for each method is shown in the bottom right panel. For small numbers of iterations (the adaptive phase of ASD), the Levenberg-Marquardt method reduces error most quickly; for larger numbers of iterations, ASD achieves 1–4 orders of magnitude smaller error for a given number of iterations than the other methods. (Note: since the genetic algorithm does not use a single initial point, individuals were instead initialized using a uniform random distribution in the range [−1, 3]. The Levenberg-Marquardt algorithm operates on the 20-dimension Powell's function identically to the 4-dimensional version, with the exception that each iteration requires 5 times as many function evaluations.)

https://doi.org/10.1371/journal.pone.0192944.g003

(a 20-dimensional Powell's quartic function), the Levenberg-Marquardt method has the lowest error for 250 or fewer iterations; for large numbers of iterations, ASD has by far the lowest error—indeed, for 2000 or more iterations, it has nearly 2 orders of magnitude less error than the Levenberg-Marquardt method, and 4 orders of magnitude less error than nonlinear

simplex, simulated annealing, and genetic algorithms. The superior performance of ASD compared to the other methods is surprising since, unlike in Fig 2, in this problem all parameters are of roughly equal importance, so the adaptive probability **p** is unlikely to significantly contribute to the efficiency of the optimization. Thus, even in cases where ASD's only advantage is its adaptive step size, it is still capable of outperforming traditional algorithms.

Optimizing HIV resource allocations

In contrast to the foregoing theoretical discussion of error minimization for analytical functions, here we describe the practical application that ASD was designed for: finding the allocation of resources across different HIV prevention and treatment programs that minimizes new infections [36]. To do this, we used the Optima HIV model (formerly known as Prevtool [15]) to perform the analyses. An overview of this version of the model is presented in S1 Appendix, with further details provided in [14]. Subsequent modifications to the model have been described in [37], and the most recent version of the software can be accessed via hiv.optimamodel.com.

In brief, the model describes HIV transmission and progression in a number of interacting subpopulations (14 in this case), including female sex workers, men who have sex with men, and general males and females in different age groups. The model incorporates parameters describing the sexual behavior, injecting behavior, HIV testing and treatment rates, and sexual and injecting partnerships of each population, as well as basic clinical parameters such as HIV transmissibility and disease progression rates. The model was based on to behavioral and surveillance data provided by the Swaziland Ministry of Health and UNAIDS. Further details are provided in [38]. In addition to empirical estimates of the model parameters, the model was calibrated to match surveillance data on HIV prevalence, diagnoses, and numbers of people on treatment. (Although ASD was also used for this calibration, here we instead focus on its use for the budget optimization procedure, since it better illustrates the differences between the methods.)

To optimize the allocation of Swaziland's HIV budget, we assumed that spending on particular HIV programs produces changes in corresponding behavioral parameters or testing and treatment rates (for example, programs targeting female sex workers increase their probability of condom use). The objective being minimized was the number of new infections over the period 2015–2020, subject to the constraint that total funding was held constant for the last year in which full budget details were available (2014). The vector **x** being optimized consisted of the budget allocations across 9 different HIV prevention, testing, and treatment programs. Thus, the optimization problem had a dimensionality of 9 (since the constraint of constant total budget, which would otherwise reduce the dimensionality to 8, is applied post hoc). The initial budgets for different programs varied by over three orders of magnitude: from US$40,000 per year for prevention programs for men who have sex with men to US$45 million per year for antiretroviral treatment. To evaluate the objective function, the budget for each program was first converted to one or more model parameter values via a nonlinear cost-outcome function, which in turn were used in the nonlinear dynamical epidemic model. The cost-outcome functions and epidemic model are described in detail in S1 Appendix. Since the model is relatively computationally intensive, requiring approximately 1–2 s per function evaluation on a standard laptop, large numbers ($>10^3$) of evaluations become wearisome.

This particular optimization problem has three notable aspects. First, despite the complexity and nonlinearity of the model, in almost all cases the objective function decreases monotonically as funding to any of the programs is increased—the only exception being HIV testing and counseling programs, in which case diagnosing more people with early-stage HIV

infections without simultaneously increasing funding for antiretroviral treatment prevents some people with late-stage HIV infections from accessing treatment. Second, the country's current HIV budget allocation, which is used as the initialization for the algorithm, is the product of considerable deliberation among numerous stakeholders and experts who have typically had the goal of allocating funds optimally. Thus, in most cases funds are already reasonably well allocated, and hence the initial starting point is expected to lie relatively close to the global optimum. Third, in situations where this is not the case, optimal solutions in very distant parts of parameter space are unlikely to be feasible given political and logistical constraints. Each of these three factors reduce the probability and/or importance of there being a difference between locally and globally optimal solutions.

As shown in Fig 4A, under current conditions, the model predicts approximately 2500 new infections per year in Swaziland. However, if funding is optimally allocated, as shown in Fig 4B (which consists largely of shifting funds from programs for orphans and vulnerable children towards treatment and male circumcision programs), this can be reduced to approximately 1260 new infections per year. ASD found this allocation after 65 function evaluations, while the next-best algorithm, the Levenberg-Marquardt method, found a nearly identical allocation after 830 function evaluations. None of the other methods reached this level of optimization within 2000 function evaluations; by that point, the genetic algorithm had achieved 99.3% of the reduction in new infections found by ASD and the Levenberg-Marquardt method, the nonlinear simplex algorithm 95%, and the simulated annealing algorithm 90%.

Discussion and summary

This paper presents a simple optimization method inspired by the process of manual parameter fitting that is capable of outperforming traditional algorithms for certain classes of problems. The algorithm is most effective for problems with moderate to large dimensionality ($\geq$5 dimensions), which corresponds to the case in which there are enough parameters that different parameters are likely to have substantially different overall contributions to the objective function. Indeed, the relative uniformity of parameters in the simple test functions used here (in terms of both scale and effectiveness) does not necessarily reflect certain real-world situations in which some—or even most—of the objective function's parameters may have little influence on its value. In such situations, as with the real-world example of HIV budget allocations, ASD is especially effective, as it is able to adapt to those parameters (and those scales) that produce the greatest improvements in the objective function. An example of this is provided in Fig 4, where ASD finds what appears to be the globally optimal solution more than 10 times faster than any other algorithm. In contrast, ASD is less effective for optimization problems where the objective function has large discontinuties or numerous local minima; for such problems, evolutionary algorithms typically provide superior performance [39].

Within the taxonomy of optimization methods, ASD is a stochastic, derivative-free, direct search method (for an excellent review of random search methods for simulation optimization, see [40]). Thus, ASD is similar to adaptive random search algorithms [41–46]. However, these algorithms are adaptive only in terms of step size, not step probability, since typically they step in all dimensions simultaneously (*e.g.*, by sampling points from a hypersphere of radius equal to the current step size), and are thus unable to obtain information about individual dimensions. In addition, they typically require additional function evaluations to calculate the optimal step size, whereas ASD updates step size automatically on each iteration. ASD also has some similarities with tabu search [47], which updates step probability (by forming "taboos"

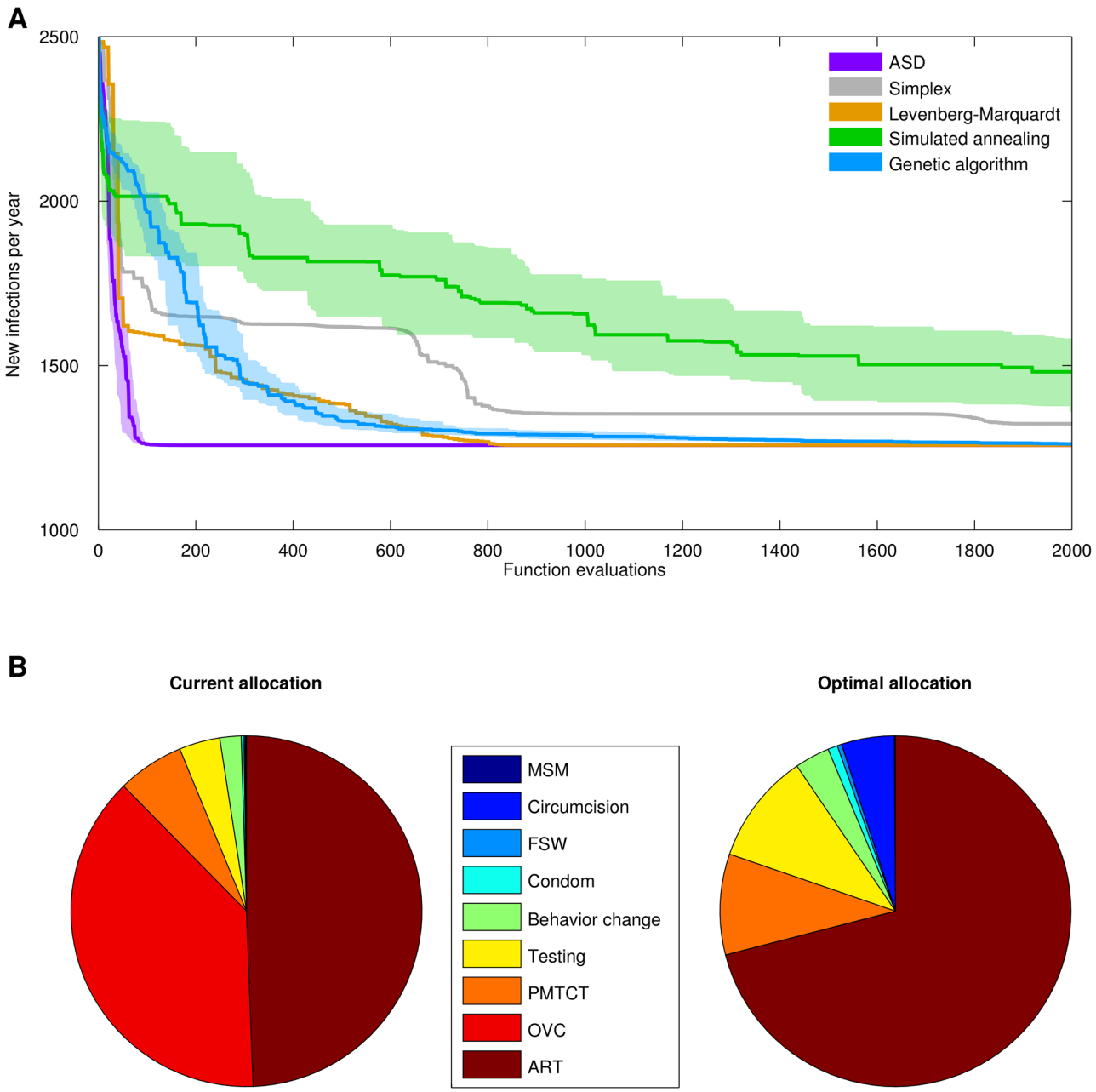

Fig 4. Comparison of optimization methods for a real-world example of HIV resource allocation. (A) Performance of each algorithm for the objective function (y-axis) of minimizing the number of new infections. As above, the shaded regions show the interquartile ranges over 40 different random seeds. (B) Original (left) and optimal (right) budgets. MSM = programs for men who have sex with men; Circumcision = voluntary medical male circumcision; FSW = programs for female sex workers; Comdom = condom promotion programs; Behavior change = social and behavior change communication; Testing = HIV testing and counseling services; PMTCT = prevention of mother-to-child transmission; OVC = programs for orphans and vulnerable children; ART = antiretroviral treatment.

https://doi.org/10.1371/journal.pone.0192944.g004

about stepping in certain directions) but not step size. Thus, ASD is loosely analogous to a combination of the adaptive random search and tabu algorithms.

This study has two main limitations. First, we chose the four algorithms to compare against ASD based on their popularity, as evidenced by their inclusion in MATLAB's Optimization

Toolbox and Python's SciPy module. However, as noted above, many other optimization algorithms exist, some of which significantly outperform these more traditional methods for particular problems—especially those that are non-convex, multi-modal, and/or have many local minima—as shown in the comprehensive review by Rios and Sahinidis [48]. Since ASD was intended as a relatively simple and general-purpose alternative to other traditional optimization algorithms, these more advanced algorithms and the complex (and often relatively specific) problems they have been designed to solve have not been considered in depth. The second limitation of this study is that MATLAB's default values of the metaparameters were used for the simulated annealing and genetic algorithms (except the initial temperature of the simulated annealing, as noted above). Metaparameter tuning would likely increase the performance of these algorithms more than it would for ASD, since these algorithms are not adaptive—but conversely, an advantage of ASD is that it typically does not require any metaparameter tuning, so in that sense the comparison is fair. In this sense, ASD is highly unusual among random search methods in that it can be used "out of the box" with consistent performance across across a wide range of optimization problems for a default set of metaparameters; in contrast, metaparameter tuning is an essential step of using other methods [49].

As noted above, ASD has already been used successfully in the real-world applications of optimizing the allocation of HIV budgets, as well as calibrating various models—of HIV epidemiology, spiking neuronal network activity, and neural field dynamics—to experimental data. In the HIV budget optimization example shown above, standard optimization methods (including the four compared against ASD in this paper) were found to require an unpleasantly large number of function evaluations to obtain acceptable solutions. This led the authors to resort to manual parameter fitting until ASD was developed. It is our hope that this algorithm may be able to free other researchers from similar unpleasantries.

Supporting information

S1 Appendix. THIV epidemic model structure, methods, and data.
(PDF)

Acknowledgments

The authors wish to thank D. Kedziora, R. M. Stuart, J. Francis, W. W. Lytton, M. Killedar, S. Kelly, A. Shattock, D. Pokrajac, and Z. McGrath for their helpful contributions.

Author Contributions

Conceptualization: Cliff C. Kerr.

Investigation: Cliff C. Kerr.

Methodology: Cliff C. Kerr, Tomasz G. Smolinski.

Software: Cliff C. Kerr, Salvador Dura-Bernal.

Supervision: David P. Wilson.

Validation: Salvador Dura-Bernal, George L. Chadderdon.

Writing – original draft: Cliff C. Kerr.

Writing – review & editing: Cliff C. Kerr, Salvador Dura-Bernal, Tomasz G. Smolinski, George L. Chadderdon, David P. Wilson.

References

1. Castillo P, Lozano R, Dzul A. Stabilization of a mini rotorcraft with four rotors. IEEE Control Systems Magazine. 2005; 25(6):45–55. https://doi.org/10.1109/MCS.2005.1550152
2. Brom C, Vyhnánek J, Lukavský J, Waller D, Kadlec R. A computational model of the allocentric and egocentric spatial memory by means of virtual agents, or how simple virtual agents can help to build complex computational models. Cognitive Systems Research. 2012;17–18:1–24.
3. Wilby RL. Uncertainty in water resource model parameters used for climate change impact assessment. Hydrological Processes. 2005; 19(16):3201–3219. https://doi.org/10.1002/hyp.5819
4. Prinz AA, Billimoria CP, Marder E. Alternative to hand-tuning conductance-based models: construction and analysis of databases of model neurons. Journal of Neurophysiology. 2003; 90(6):3998–4015. https://doi.org/10.1152/jn.00641.2003 PMID: 12944532
5. Baker JL, Perez-Rosello T, Migliore M, Barrionuevo G, Ascoli GA. A computer model of unitary responses from associational/commissural and perforant path synapses in hippocampal CA3 pyramidal cells. Journal of Computational Neuroscience. 2011; 31(1):137–158. https://doi.org/10.1007/s10827-010-0304-x PMID: 21191641
6. Kerr CC, Van Albada SJ, Neymotin SA, Chadderdon GL, Robinson P, Lytton WW. Cortical information flow in Parkinson's disease: a composite network/field model. Frontiers in Computational Neuroscience. 2013; 7:1–14. https://doi.org/10.3389/fncom.2013.00039
7. Kwon JA, Anderson J, Kerr CC, Thein HH, Zhang L, Iversen J, et al. Estimating the cost-effectiveness of needle-syringe programs in Australia. AIDS. 2012; 26(17):2201–2210. https://doi.org/10.1097/QAD.0b013e3283578b5d PMID: 22914579
8. Song W, Kerr CC, Lytton WW, Francis JT. Cortical plasticity induced by spike-triggered microstimulation in primate somatosensory cortex. PLOS ONE. 2013; 8(3):e57453. https://doi.org/10.1371/journal.pone.0057453 PMID: 23472086
9. Goodner J, Tsianos GA, Li Y, Loeb GE. BioSearch: A physiologically plausible learning model for the sensorimotor system. In: Proceedings of the Society for Neuroscience Annual Meeting; 2012. 275.22/LL11.
10. Glynn PW, Whitt W. The asymptotic efficiency of simulation estimators. Operations Research. 1992; 40(3):505–520. https://doi.org/10.1287/opre.40.3.505
11. Hilbert M, López P. The world's technological capacity to store, communicate, and compute information. Science. 2011; 332(6025):60–65. https://doi.org/10.1126/science.1200970 PMID: 21310967
12. Charness G, Karni E, Levin D. Individual and group decision making under risk: An experimental study of Bayesian updating and violations of first-order stochastic dominance. Journal of Risk and Uncertainty. 2007; 35(2):129–148. https://doi.org/10.1007/s11166-007-9020-y
13. Steyvers M, Lee MD, Wagenmakers EJ. A Bayesian analysis of human decision-making on bandit problems. Journal of Mathematical Psychology. 2009; 53(3):168–179. https://doi.org/10.1016/j.jmp.2008.11.002
14. Kerr CC, Stuart RM, Gray RT, Shattock A, Fraser N, Benedikt C, et al. Optima: a model for HIV epidemic analysis, program prioritization, and resource optimization. Journal of Acquired Immune Deficiency Syndromes. 2015; 69:365–376. https://doi.org/10.1097/QAI.0000000000000605 PMID: 25803164
15. Eaton JW, Menzies NA, Stover J, Cambiano V, Chindelevitch L, Cori A, et al. Health benefits, costs, and cost-effectiveness of earlier eligibility for adult antiretroviral therapy and expanded treatment coverage: a combined analysis of 12 mathematical models. The Lancet Global Health. 2014; 2(1):e23–e34. https://doi.org/10.1016/S2214-109X(13)70172-4 PMID: 25104632
16. Fraser N, Kerr CC, Harouna Z, Alhousseini Z, Cheikh N, Gray RT, et al. Re-orienting the HIV response in Niger towards sex work interventions: from better evidence to targeted and expanded practice. Journal of Acquired Immune Deficiency Syndromes. 2015; 68:S213–220. https://doi.org/10.1097/QAI.0000000000000456 PMID: 25723987
17. Zhang L, Phanuphak N, Henderson H, Nonenoy S, Srikaew S, Shattock A, et al. Scaling up HIV treatment for MSM in Bangkok: what does it take?—a modelling and costing study. The Lancet HIV. 2015; 2:e200–207.
18. Pham QD, Wilson DP, Kerr CC, Shattock AJ, Do HM, Duong AT, et al. Estimating the cost-effectiveness of HIV prevention programmes in Vietnam, 2006-2010: A modelling study. PLOS ONE. 2015; 10:e0133171. https://doi.org/10.1371/journal.pone.0133171 PMID: 26196290
19. Shattock AJ, Kerr CC, Stuart RM, Masaki E, Fraser N, Benedikt C, et al. In the interests of time: improving HIV allocative efficiency modelling via optimal time-varying allocations. Journal of the International AIDS Society. 2016; 19:20627. https://doi.org/10.7448/IAS.19.1.20627 PMID: 26928810

20. Kelly SL, Shattock AJ, Kerr CC, Stuart RM, Papoyan A, Grigoryan T, et al. Optimizing HIV/AIDS resources in Armenia: increasing ART investment and examining HIV programmes for seasonal migrant labourers. Journal of the International AIDS Society. 2016; 19:20772. https://doi.org/10.7448/IAS.19.1.20772 PMID: 27281790

21. Benedikt C, Kelly SL, Wilson D, Wilson DP, Optima Consortium, et al. Allocative and implementation efficiency in HIV prevention and treatment for people who inject drugs. International Journal of Drug Policy. 2016; 38:73–80. https://doi.org/10.1016/j.drugpo.2016.10.011 PMID: 27883944

22. Shattock AJ, Benedikt C, Bokazhanova A, Đurić P, Petrenko I, Ganina L, et al. Kazakhstan can achieve ambitious HIV targets despite expected donor withdrawal by combining improved ART procurement mechanisms with allocative and implementation efficiencies. PLOS ONE. 2017; 12(2):e0169530. https://doi.org/10.1371/journal.pone.0169530 PMID: 28207809

23. Choi JS, Menzies RJ, Dura-Bernal S, Francis JT, Lytton WW, Kerr CC. Spiking network modeling of neuronal dynamics in individual rats. BMC Neuroscience. 2015; 16(Suppl 1):P122.

24. Zobaer M, Anderson R, Kerr C, Robinson P, Wong K, D'Rozario A. K-complexes, spindles, and ERPs as impulse responses: unification via neural field theory. Biological Cybernetics. 2017; 111(2):149–164. https://doi.org/10.1007/s00422-017-0713-2 PMID: 28251306

25. Metropolis N, Ulam S. The Monte Carlo method. Journal of the American Statistical Association. 1949; 44(247):335–341. https://doi.org/10.1080/01621459.1949.10483310 PMID: 18139350

26. Prudius AA, Andradóttir S. Simulation optimization using balanced explorative and exploitative search. In: Proceedings of the 36th Conference on Winter Simulation; 2004. p. 545–549.

27. Kelly SL, Wilson DP. GBD 2015 and HIV estimates from the Optima model. The Lancet HIV. 2016; 3 (12):e558. https://doi.org/10.1016/S2352-3018(16)30192-8 PMID: 27884372

28. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. Journal of Chemical Physics. 1953; 21(6):1087–1092. https://doi.org/10.1063/1.1699114

29. Spall JC. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control. 1992; 37(3):332–341. https://doi.org/10.1109/9.119632

30. Nelder JA, Mead R. A simplex method for function minimization. Computer Journal. 1965; 7(4):308–313. https://doi.org/10.1093/comjnl/7.4.308

31. Marquardt DW. An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial & Applied Mathematics. 1963; 11(2):431–441. https://doi.org/10.1137/0111030

32. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simmulated annealing. Science. 1983; 220 (4598):671–680. https://doi.org/10.1126/science.220.4598.671 PMID: 17813860

33. Bethke AD. Genetic algorithms as function optimizers. University of Michigan; 1978.

34. Ben-Ameur W. Computing the initial temperature of simulated annealing. Computational Optimization and Applications. 2004; 29(3):369–385. https://doi.org/10.1023/B:COAP.0000044187.23143.bd

35. Donsker MD. Justification and extension of Doob's heuristic approach to the Kolmogorov-Smirnov theorems. The Annals of Mathematical Statistics. 1952;p. 277–281. https://doi.org/10.1214/aoms/1177729445

36. Anderson SJ, Cherutich P, Kilonzo N, Cremin I, Fecht D, Kimanga D, et al. Maximising the effect of combination HIV prevention through prioritisation of the people and places in greatest need: a modelling study. The Lancet. 2014; 384(9939):249–256. https://doi.org/10.1016/S0140-6736(14)61053-9

37. Stuart R, Fraser-Hurt N, Kerr C, Mabusela E, Madi V, Mkhwanazi F, et al. Can the City of Johannesburg end AIDS by 2030? An analysis of the impact of achieving the fast-track targets and what it will take to get there. Journal of the International AIDS Society; in press.

38. Kelly S, Shattock A, Kerr CC, Gama T, Nhlabatsi N, Zagatti G, et al. HIV Mathematical Modelling to Support Swaziland's Development of its HIV Investment Case. The World Bank; 2014.

39. Dura-Bernal S, Neymotin S, Kerr C, Sivagnanam S, Majumdar A, Francis J, et al. Evolutionary algorithm optimization of biological learning parameters in a biomimetic neuroprosthesis. IBM Journal of Research and Development. 2017; 61(2/3):6–1. https://doi.org/10.1147/JRD.2017.2656758 PMID: 29200477

40. Andradóttir S. An overview of simulation optimization via random search. Handbooks in Operations Research and Management Science. 2006; 13:617–631. https://doi.org/10.1016/S0927-0507(06)13020-0

41. Hooke R, Jeeves TA. "Direct search" solution of numerical and statistical problems. Journal of the Association for Computing Machinery. 1961; 8(2):212–229. https://doi.org/10.1145/321062.321069

42. Schumer MA, Steiglitz K. Adaptive step size random search. IEEE Transactions on Automatic Control. 1968; 13(3):270–276. https://doi.org/10.1109/TAC.1968.1098903

43. Widrow B, McCool JM. A comparison of adaptive algorithms based on the methods of steepest descent and random search. IEEE Transactions on Antennas and Propagation. 1976; 24(5):615–637. https://doi.org/10.1109/TAP.1976.1141414

44. Masri S, Bekey G, Safford F. A global optimization algorithm using adaptive random search. Applied Mathematics and Computation. 1980; 7(4):353–375. https://doi.org/10.1016/0096-3003(80)90027-2

45. Zabinsky ZB, Smith RL. Pure adaptive search in global optimization. Mathematical Programming. 1992; 53(1-3):323–338. https://doi.org/10.1007/BF01585710

46. Zabinsky ZB. Stochastic adaptive search for global optimization. vol. 72. Springer Science & Business Media; 2013.

47. Glover F. Tabu search—part I. ORSA Journal on Computing. 1989; 1(3):190–206. https://doi.org/10.1287/ijoc.1.3.190

48. Rios LM, Sahinidis NV. Derivative-free optimization: A review of algorithms and comparison of software implementations. Journal of Global Optimization. 2013; 56(3):1247–1293. https://doi.org/10.1007/s10898-012-9951-y

49. Trelea IC. The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters. 2003; 85(6):317–325. https://doi.org/10.1016/S0020-0190(02)00447-7