RESEARCH ARTICLE

REVISED A novel pseudoderivative-based mutation operator for real-coded adaptive genetic algorithms [v2; ref status: indexed, http://f1000r.es/1td]

Previously titled: Using computation to enhance diagnosis and therapy: a novel mutation operator for real-coded adaptive genetic algorithms

Maxinder S Kanwal, Avinash S Ramesh, Lauren A Huang

Thomas Jefferson High School for Science and Technology, Alexandria, VA 22312, USA

v2 **First published:** 11 Jun 2013, **2**:139 (doi: 10.12688/f1000research.2-139.v1)
Latest published: 19 Nov 2013, **2**:139 (doi: 10.12688/f1000research.2-139.v2)

Abstract

Recent development of large databases, especially those in genetics and proteomics, is pushing the development of novel computational algorithms that implement rapid and accurate search strategies. One successful approach has been to use artificial intelligence and methods, including pattern recognition (e.g. neural networks) and optimization techniques (e.g. genetic algorithms). The focus of this paper is on optimizing the design of genetic algorithms by using an adaptive mutation rate that is derived from comparing the fitness values of successive generations. We propose a novel pseudoderivative-based mutation rate operator designed to allow a genetic algorithm to escape local optima and successfully continue to the global optimum. Once proven successful, this algorithm can be implemented to solve real problems in neurology and bioinformatics. As a first step towards this goal, we tested our algorithm on two 3-dimensional surfaces with multiple local optima, but only one global optimum, as well as on the N-queens problem, an applied problem in which the function that maps the curve is implicit. For all tests, the adaptive mutation rate allowed the genetic algorithm to find the global optimal solution, performing significantly better than other search methods, including genetic algorithms that implement fixed mutation rates.

Open Peer Review

Referee Status: ✓ ✓ ✓

	Invited Referees		
	1	**2**	**3**
REVISED **version 2** published 19 Nov 2013	✓ report	✓ report	✓
version 1 published 11 Jun 2013	? report	✓ report	? report

1 **Christos Ouzounis**, National Center for Research & Technology Greece

2 **Mark Ragan**, University of Queensland Australia

3 **Panayiotis (Takis) Benos**, University of Pittsburgh USA

Discuss this article

Comments (0)

Corresponding author: Maxinder S Kanwal (max.kanwal95@gmail.com)

How to cite this article: Kanwal MS, Ramesh AS and Huang LA. **A novel pseudoderivative-based mutation operator for real-coded adaptive genetic algorithms [v2; ref status: indexed, http://f1000r.es/1td]** *F1000Research* 2013, **2**:139 (doi: 10.12688/f1000research.2-139.v2)

Grant information: The author(s) declared that no grants were involved in supporting this work.

Competing interests: No competing interests were disclosed.

First published: 11 Jun 2013, **2**:139 (doi: 10.12688/f1000research.2-139.v1)
First indexed: 22 Jul 2013, **2**:139 (doi: 10.12688/f1000research.2-139.v1)

REVISED **Changes from Version 1**

The title was modified to avoid giving a false sense of the subject matter of the study. The abstract was modified to focus on the computational findings rather than the applications. Implications mentioned in the Discussion were expanded to elaborate on potential uses for our algorithm in bioinformatics, among other fields. Additional references were added based on the new information.

See referee reports

Introduction

The last few years have seen an exponential increase in the size of databases, especially those in genetics, which catalog the basis of various diseases. Computing power has not kept pace with this rapid increase in bioinformatics. Therefore, it has become critical to develop clever algorithms to reduce the time needed to search such databases and arrive at solutions to the treatment of genetically determined diseases[1]. Genetic algorithms provide new hope to overcome this dilemma. Genetic algorithms attempt to copy the principle of "survival of the fittest", using Darwin's theory of evolution to find a satisfactory solution. In essence, a group of random solutions is created and ranked, after which the best solutions are allowed to "interbreed". During interbreeding, small mutations are allowed to add an element of randomness, and in this way aid the genetic algorithm in finding the optimal solution[2].

The applications of genetic algorithms (GAs) are well known and far ranging[3]. Because of their versatility, GAs have been widely and successfully used to optimize practical problems such as determining a long sequence of base pairs in a genetic database, scheduling drilling operations in a circuit-board factory, and data mining, among others[4]. However, one major problem faced by GAs is premature convergence, in which the algorithm becomes trapped in a local optimum and is unable to find the global optimum[5].

This study focuses on the use of mutations in GAs. Normally, the mutations occur at a constant rate, known as the mutation rate. However, use of a fixed mutation rate can produce sub-optimal solutions[5]. We propose a new, variable mutation rate that uses a pseudoderivative to take into account the time that a GA can be stuck at a certain point. The longer the algorithm has been stuck at a local optimum, the more likely it is that a mutation will occur. This addition of a greater element of randomness to the algorithm allows it to move from the local optimum and look for better solutions. In this paper, we show that the proposed variable mutation rate outperforms fixed mutation rates and other common search algorithms in the effectiveness of their solutions.

Background

Rationale

Many real-life problems can be modeled as continuous, nonlinear optimization problems. Within a given search space S on the optimization function f, a global (absolute) optimum is sought. This may take the form of a global maximum or minimum, depending on the original problem. A typical 3-dimensional global optimization problem follows the form:

$$\text{maximize/minimize } f(x, y), \text{ where } (x, y) \in S \qquad (1)$$

The challenge in solving a global optimization problem is in seeking the global optimum rather than becoming trapped in a local optimum, an issue that will be addressed in more detail later[6]. These optimization problems can be approached with a variety of techniques. One popular technique is the use of GAs, the focus of this study[7].

Genetic algorithms (GAs)

GAs are population-based optimization techniques favored for their properties of self-learning, self-organization, self-adaptation, and implicit parallelism[8]. Based on the principles of Charles Darwin's natural selection and meiotic cell division, GAs involve several components: a population, a measure of fitness, and a method of breeding[2]. The population forms the basis for the GA and is made up of many individuals, often called chromosomes. Over time, "chromosomes" breed with other "chromosomes" to form "children" that make up a new generation in the GA[9].

Chromosomes can be made up of binary strings or of real values. Binary-coded GAs (BCGAs) have chromosomes of 0s and 1s. While BCGAs are the more traditional method and are adequate for small- to moderate-size optimization problems, they fail for high-dimensional problems because they require more computational time and memory. Real-coded GAs (RCGAs) utilize real values that allow for both improved computational time and memory as compared to BCGAs, making the optimization of multi-dimensional and high-precision problems more feasible[5]. Each chromosome (binary-coded or real-coded) has a certain fitness value or f value derived from its binary string or its real values, where the f value of a chromosome refers to its value when inputted into the function described in (1). The calculations for the f value of a chromosome vary by problem. In this respect, each chromosome represents a single solution to the optimization problem[10]. Following the principles of natural selection, chromosomes with higher fitness values give rise to children with high fitness values, so the GA "selects" for fitter chromosomes by giving them a higher probability of breeding and passing on their genes[11].

The breeding stage involves two processes: crossover and mutation. In true meiotic cell division (the process by which a sperm or an egg cell is created), a crossover occurs when two chromosomes pair with each other and exchange portions of their length, forming two hybrid chromosomes. Similarly, chromosomes in a GA exchange values[12]. A simple example of crossover is illustrated in Figure 1.

The second operator of the breeding stage is mutation. Mutation is a naturally-occurring phenomenon that may come into play during any replication of chromosomes. Incorporation of mutation into a GA may be considered an optional step, but has generally been found to increase the diversity of the population[13]. This allows GAs to explore promising areas of the search space[8]. The major advantage of implementing mutations in a GA is in avoiding premature convergence, in which the GA becomes trapped in a local optimum[14]. However, the mutation operator has trade-offs in slowing down the learning process[5]. Mutations have in the past taken the form of point, frame-shift, and translocation mutations, all of which involve

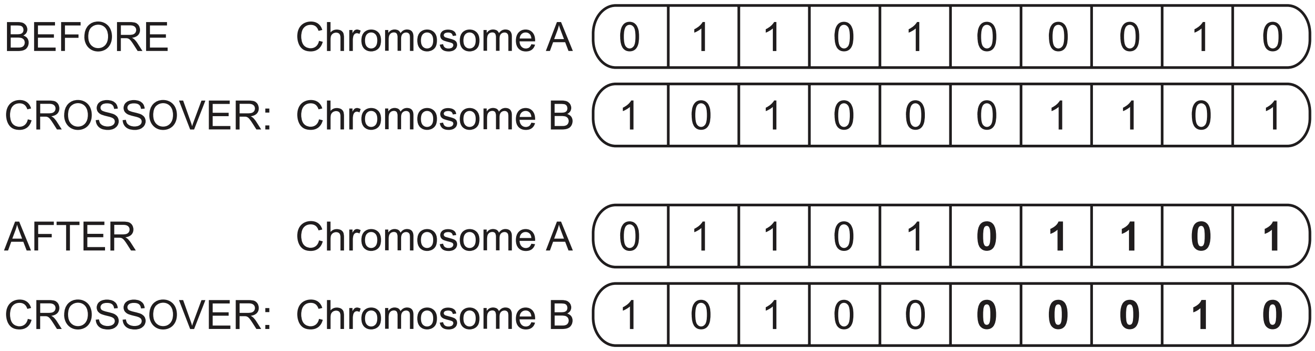
BEFORE CROSSOVER:
Chromosome A 0 1 1 0 1 0 0 0 1 0
Chromosome B 1 0 1 0 0 0 1 1 0 1
AFTER CROSSOVER:
Chromosome A 0 1 1 0 1 0 1 1 0 1
Chromosome B 1 0 1 0 0 0 0 0 1 0

Figure 1. Example of crossover between two binary-coded chromosomes. An example of crossover between a set of two binary-coded chromosomes where half the length of chromosome **A** is exchanged for half the length of chromosome **B**, resulting in a hybrid set of chromosomes.

swapping, switching, or shifting binary bits, in the case of binary-coded chromosomes[15]. Random mutation has also been proposed, in which a gene is replaced with a random value within certain lower and upper bounds[16], and may provide interesting insight into the mechanisms of mutation, since a small mutation in nature may cause a gene to take on a vastly different role[17]. Finally, the rate of mutation, r_m, can play a key role in the effectiveness of a given GA[7].

A variety of modifications have been made to GAs in the breeding stage. A large amount of attention has been devoted to developing improvements in crossover operators, but there have been fewer studies in improving mutation operators, which make up a very promising although largely unexplored branch[5,15]. Several studies have implemented adaptive mutation, or a changing mutation rate[18,19]. Adaptive mutation methods have ranged from individual gene-based mutations to Gaussian mutation operators based on the mean and the standard deviation of the Gaussian distribution[20]. This study proposes a novel adaptive mutation approach and applies it to two tests: the well-known N-queens problem and the maximization of a 3-dimensional function. This study also compares a GA implementing an adaptive mutation operator with other well-known search algorithms, such as the Nelder-Mead method, hill climbing technique, and random search.

Other search algorithms

Numerous other search algorithms besides GAs exist for solving similar optimization problems[21]. A basic method is the random search method whereby solutions are randomly chosen and evaluated for a certain amount of time, and the best solution found during the time span is returned[22]. A more advanced method implements the hill climbing technique. This method begins at a random point and uses a greedy algorithm to move to the best immediate neighboring solution until no neighboring solution is better than the current solution[23]. The final, and most robust, non-GA search algorithm being examined in this study is the Nelder-Mead algorithm. The Nelder-Mead method is a heuristic search that relies on approximating a local optimum for a problem with N variables by using a simplex (a polytope consisting of N + 1 vertices)[24].

The N-queens problem

The N-queens problem is an interesting application for testing the accuracy of genetic algorithms due to the inherent difficulty of optimizing the problem under efficient time and memory constraints[25]. Given an N × N board, find a set of N-queens such that zero pairs of queens are in the same row, column, or diagonal (none of the queens are attacking each other). Although there are several approaches (brute force/recursion, simulated annealing, etc.) to solving this optimal placement problem[26], the N-queens problem was chosen in this study as an early practical test of the proposed new mutation rate operator's accuracy.

Optimizing a 3-dimensional function

Finding the maximum or minimum value of a 3-dimensional surface is a more abstract but more visual problem. A surface with multiple local maxima and minima can simultaneously test a GA's ability to avoid premature convergence and its ability to handle multidimensional optimization problems. Because GAs solve problems with implicit functions of N-dimensions, it is important that the newly proposed mutation operator can deliver reasonable speed and accuracy in the most primal form of the N-dimensional problem, optimizing a space function. While the N-queens problem is adequate for testing the accuracy of a GA, it is not adequate for testing its speed of convergence because there are multiple solutions. Therefore, maximization of a 3-dimensional surface with multiple local maxima but only one absolute maximum was chosen as a practical test of the new mutation operator's speed of convergence, or number of generations needed to converge to the optimal solution. Furthermore, minimization of the Rastrigin function was chosen as a simple method for comparing accuracies of various search algorithms. The Rastrigin function is an optimal function to choose for such a comparison as the surface holds a myriad of local minima, but only one global minimum at $f(0,0) = 0$. The Rastrigin function models a difficult problem to optimize, where a non-robust search algorithm may easily become caught in a local optimum[27].

Methods

Experimental design

The research question this study attempted to answer was: what are the effects of an adaptive mutation rate, based on the derivative of the fitness function with respect to generations, on the efficiency and accuracy of a GA? We performed three experiments to answer this question. The first experiment in this study tested the null hypothesis that given a problem, which is not always solvable (to the global optimum) by a fixed-rate GA, a GA with an adaptive mutation rate (based on the derivative of the fitness function with respect to generations) would find an optimal solution significantly more often than one with a constant mutation rate. To reject the null hypothesis, we will need to show that an adaptive mutation rate can find an optimal solution significantly more often than one with a fixed mutation rate. The second experiment tested the null hypothesis that given a problem, which is always solvable (to the global optimum) by a constant mutation rate GA, there would be no significant difference in the efficiency between the constant mutation rate GA and a GA with an adaptive mutation rate, based on the derivative of the fitness function with respect to generation. The third experiment tested the null hypothesis that a GA with an adaptive mutation rate is not significantly more accurate in converging to a global optimum of a 3-dimensional function as compared to other search algorithms (i.e. Nelder-Mead, hill climbing, random).

The mutation rate function r_m was the independent variable in the first study. Accuracy (how often a GA finds the optimum solution) measured in percent (%) with an error value of ±0.001, and efficiency (how long it takes to converge to the solution), measured in generations, were the dependent variables. The experiment was set up with three levels: a constant r_m of 0% (no mutation, control), a constant r_m of 20% (control), and an adaptive r_m derived from a pseudoderivative and a sigmoid function. The independent variable of the second study was the specific search algorithm being run. The accuracy, measured in percent (%) with an error value of ±0.001, was the dependent variable. The experiment was set up with four levels (search algorithms): a random search, a search implementing the hill climbing technique, a search implementing the Nelder-Mead method, and a search implementing the proposed adaptive GA.

The adaptive mutation rate operator was tested on three problems: the N-queens problem, the maximization of a 3-dimensional surface, and the minimization of a different 3-dimensional surface (the Rastrigin function). The N-queens problem was used specifically to test the accuracy of the new adaptive mutation rate since a GA implementing a constant mutation rate does not have 100% accuracy in the N-queens problem. The maximization of a 3-dimensional surface problem was used to test the efficiency of the new adaptive mutation rate since a GA implementing the 20% constant mutation rate does have 100% accuracy in the 3-dimensional maximization problem. The minimization of the Rastrigin function was used to determine the robustness of the proposed adaptive GA compared to other well-known search algorithms. The Rastrigin function was selected due to a most diverse topography, including a myriad of local optima by which weak search algorithms may be forced to prematurely converge. In the first study, accuracy was defined to be the rate of successful optimization within 100,000 generations to the nearest ±0.001, in order to save computing power. Python 2.5 was used to create and run GAs for the N-queens and 3-dimensional maximization problems. Python 3.0 was used to create and run an adaptive GA, Nelder-Mead, hill climbing, and random search algorithms for the Rastrigin function minimization problem. A vector class was created to aid in simplifying the code for the latter experiment. Both GAs reported mutation rate, convergence, and best f values for later analysis. For both problems, the mutation rate operator function was a sigmoid function fit to the domain x in $[0, \infty)$ and range y in $[0, 1]$. The sigmoid function is defined as:

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{2}$$

with domain x in $(-\infty, \infty)$, range y in $[0, 1]$, and $sigmoid(0) = 0.5$. The function used was thus a fit of the monotonic sigmoid function to the desired domain and range, resulting in

$$r_m = 2 * (sigmoid(x) - 0.5) \tag{3}$$

where x is given by:

$$x = g_c - g_o \tag{4}$$

with g_c representing the current generation and g_o representing the oldest generation of the same best f value. x thus is inversely related to the derivative of the fitness function with respect to generation. This causes an increase in mutation rate when the fitness between generations is stagnant. A theoretical basis and further details of the adaptive mutation rate operator is included in the Theory section under the Discussion.

For the N-queens problem, the specific case of N = 8 was chosen. The fitness function was chosen for this problem to be the number of pairs of queens violating the problem specification. A program was written to run 200 trials of GAs maxed at 100,000 generations or until a solution was found ($f = 0$). The genetic code for each solution in the solution set was a sequence of eight integers, with each index representing a column and each value representing a row. Crossover locations were determined using random integers. All variables were held constant apart from mutation rates, which varied between constant mutation at 0% (no mutation), constant mutation rate of 20%, and the sigmoidally-determined adaptive mutation. For the full code, see Script 1 in the Supplementary materials.

Regarding the maximization of a 3-dimensional function, the proposed sigmoidal mutation operator was again compared to controls of a constant mutation rate of 20% and of no mutation (0%). The function chosen for maximization was

$$z = 3*(1-x)^2 * e^{-x^2-(y+1)^2} - 10*\left(\frac{x}{5} - x^3 - y^5\right) * e^{-x^2-y^2} - \frac{1}{3} * e^{-(x+1)^2-y^2} \tag{5}$$

which represents a function with several local maxima and minima, which can be solved by the 20% mutation rate GA but not the 0% mutation rate GA. This allowed for a test of efficiency of the new sigmoidally-determined adaptation mutation rate GA against the 20% fixed mutation rate GA and a test of accuracy for the sigmoidally-determined adaptation mutation rate GA against the 0% fixed mutation rate (no mutation) GA. A graphical representation of the function is shown as Figure 2. The genetic code for each solution in the solution set for this problem was a sequence of two integers, the x and y coordinates. The chromosomes were implemented as real-coded chromosomes containing the actual coordinates rather than binary strings. For crossover, the x coordinate of one solution was chosen and the y coordinate of another solution was chosen. For mutation, a coordinate was replaced by a random number within the domain of the problem. Once again, all variables, except for mutation rates, were held constant throughout each experiment.

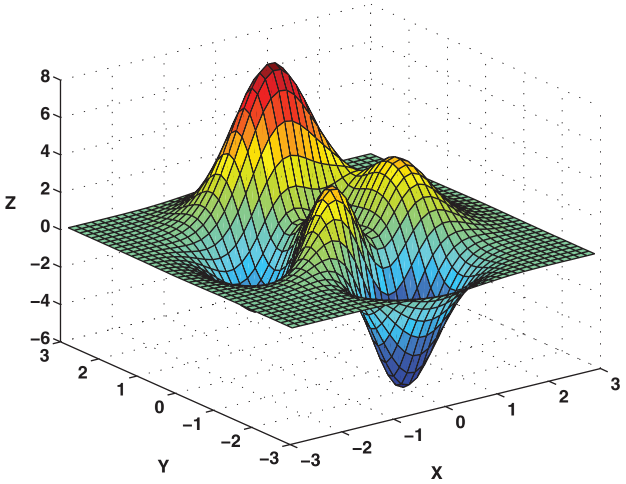

Figure 2. Graphical representation of the surface used for the 3D maximization problem. Visualized in MATLAB 6.5 Using the Command peaks(40). The three maximums of the surface were located at $f = 8.1165$ (global maximum), 3.5507, and 3.4652.

The minimization of the Rastrigin function, given by:

$$z = 20 + x^2 - 10\cos(2\pi x) + y^2 - 10\cos(2\pi y) \qquad (6)$$

was the final test run, with a goal to learn how the proposed adaptive GA compares to non-GA search algorithms. All of the search algorithms were coded to report the run time to converge and the result of the respective convergence. In order for a result to be considered accurate, the returned answer had to be within 0.001 of the true answer. This ensures that the algorithm is converging to the correct minimum and also aids in simulating a difficult optimization problem. Since the random method does not converge, but rather runs for a given time span, the average run time for each GA trial was obtained and used for the run time of the random method search. This allows for direct comparison of the accuracies of the random method and the adaptive GA, as the run time for each method was identical, and thus, held constant. Each algorithm was run for 1,000 trials in order to obtain enough data for a robust analysis. For the full code, see Scripts 2, 3, and 4 in the Supplementary materials. A graphical representation of the Rastrigin function can be found as Figure 3 and Figure 4.

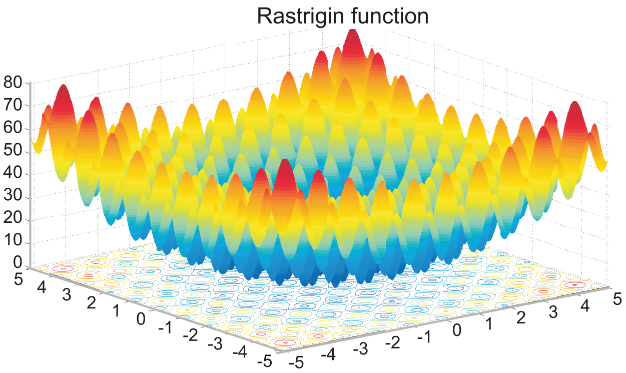

Figure 3. Graphical representation of the Rastrigin function used in the comparison of the Nelder-Mead, Hill Climbing, and Random Search algorithms with the adaptive genetic algorithm. Visualized in MATLAB 6.5: f = @(x,y) 10*2 + x.^2 + y.^2 – 10*cos(2*pi*x) – 10*cos(2*pi*y). The global minimum of the Rastrigin function is $f(0,0) = 0$.

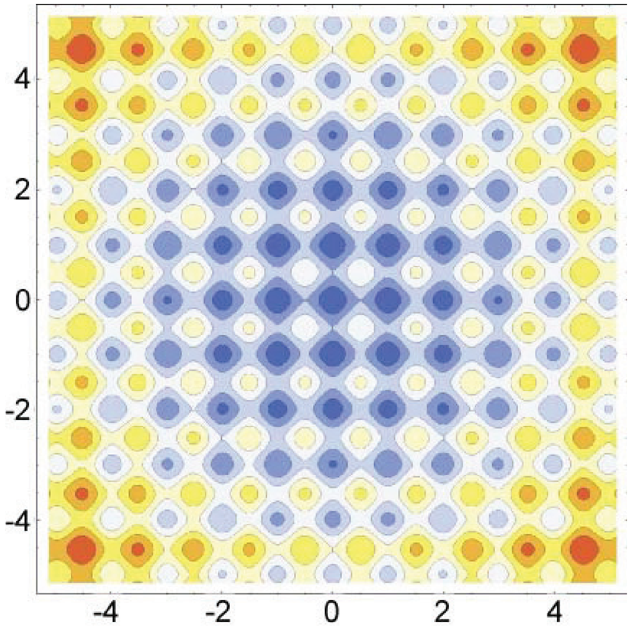

Figure 4. Contour plot of the Rastrigin function. Visualized in Mathematica 8: ContourPlot[10*2 + x^2 + y^2 – 10 Cos[2 Pi x] –10 Cos[2 Pi y], {x, -5.12, 5.12}, {y, -5.12, 5.12}, ImageSize -> 1000, Axes -> False, ColorFunction -> ColorData["TemperatureMap"]].

Results

The results of experimentation are summarized in Table 1, Table 2, Table 5, and Table 7. In the N-queens problem, the proposed sigmoid mutation operator displayed both the fastest convergence and greatest accuracy with convergence after an average of 25,455.4 generations and an accuracy of 95.5%. In the 3-dimensional surface maximization problem, there was no significant difference between the speeds of the successful GAs. Accuracies were tied between 20% constant and adaptive mutation, as both techniques consistently identified the global maximum within the 100,000-generation limit. In the Rastrigin function minimization problem, the proposed sigmoid mutation operator GA performed significantly better than the three other search algorithms in terms of accurately converging to the global minimum. See Figure 5–Figure 7 for a summary of the results and significant differences.

N-queens

In the N-queens problem, a series of unpaired, single-tailed Student's t-tests were used to test the following null hypothesis at the 0.05 level of significance: the accuracy of the adaptive mutation operator will not be significantly higher than that of 20% constant mutation or that of 0% constant mutation. The null hypothesis was rejected for the 20% constant mutation rate ($p = 6.108 \times 10^{-25} < 0.05$) and the 0% constant mutation rate ($p = 2.572 \times 10^{-214} < 0.05$). See Table 3 for a summary of the N-queens statistics.

Accuracies, run times, and statistics for genetic algorithms with various mutation rates solving the N-queens problem

1 Data File

http://dx.doi.org/10.6084/m9.figshare.640022

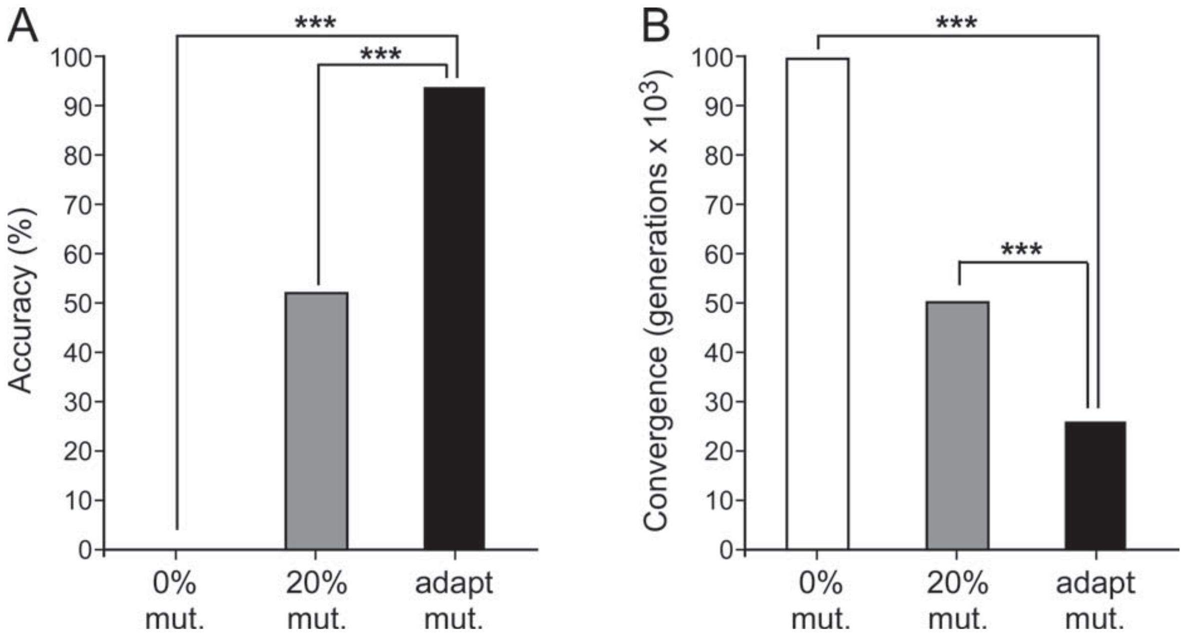

Figure 5. N-queens accuracy and convergence results for constant and adaptive mutation rate genetic algorithms. A. Graph of the N-queens problem accuracies among 0% constant mutation, 20% constant mutation, and adaptive mutation. ***p < 0.001. **B**. Graph of the N-queens problem convergences among 0% constant mutation, 20% constant mutation, and adaptive mutation. ***p < 0.001. mutation (mut.), adaptive (adapt.).

Table 1. Summary of results from solving the N-Queens problem using various mutation rates.

	Constant (0%) mutation	Constant (20%) mutation	Adaptive mutation
Overall accuracy (%)	0	53.5	95.5
Average convergence (generations)	100000	50661.49	25455.4

Table 2. Summary of results from solving the 3-dimensional maximization problem using various mutation rates.

	Constant (0%) mutation	Constant (20%) mutation	Adaptive mutation
Overall accuracy (%)	0	100	100
Average convergence (generations)	100000	67460.99	67779.92

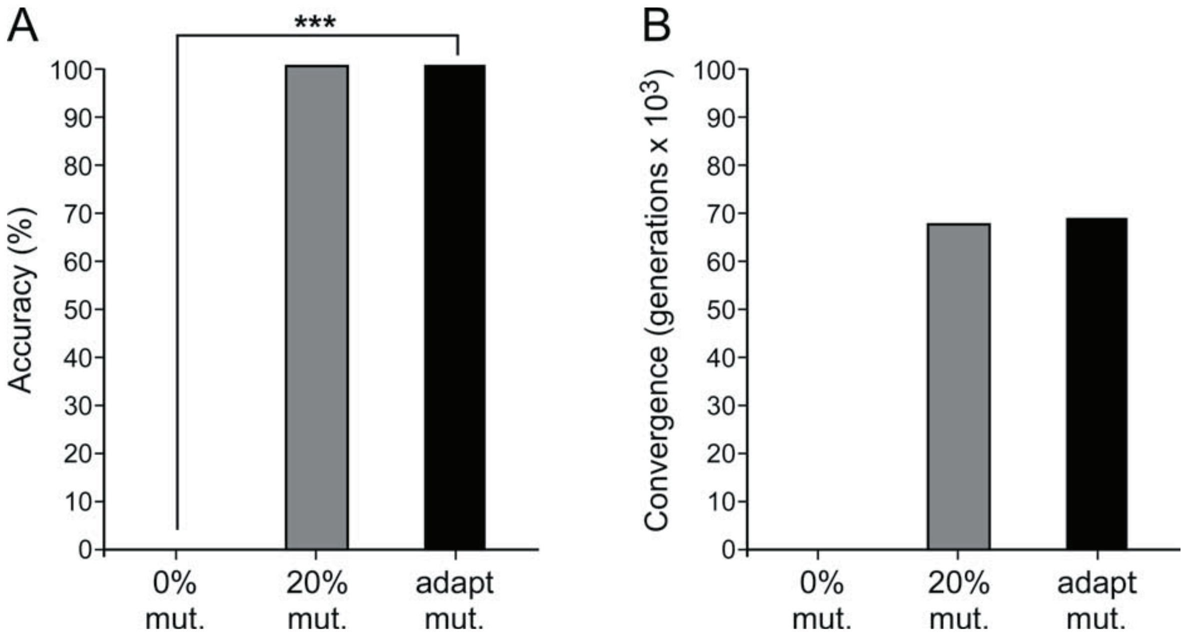

Figure 6. 3D maximization accuracy and convergence results for constant and adaptive mutation rate genetic algorithms. A. This is a graph of the 3D surface maximization problem accuracies among 0% constant mutation, 20% constant mutation, and adaptive mutation. **B**. This is a graph of the 3D surface maximization problem convergences among 20% constant mutation and adaptive mutation. 0% constant mutation was not statistically analyzed because convergence was not achieved. ***p < 0.001. mutation (mut.), adaptive (adapt.).

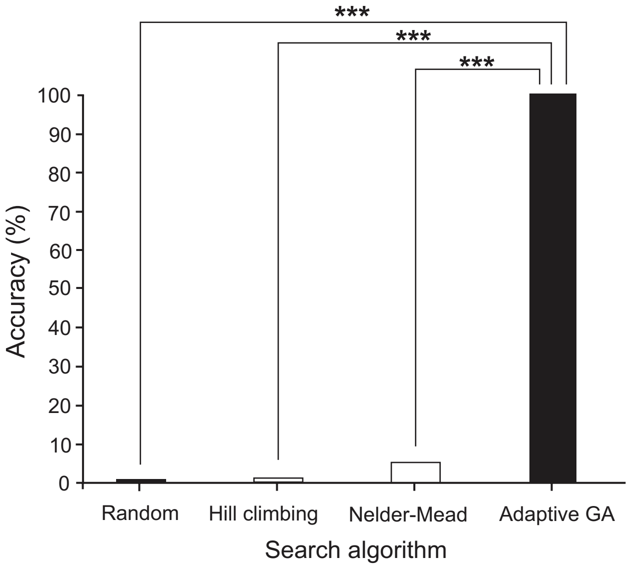

Figure 7. Rastrigin function minimization accuracy results for various search algorithms. Graph of the Rastrigin function minimization problem accuracies among random, Hill climbing, Nelder-Mead, and adaptive GA methods. ***p < 0.001. genetic algorithm (GA).

Table 3. Summary of statistical tests on N-queens results comparing the accuracies and convergence generations associated with adaptive and constant mutation rates.

T-tests	p-value
Constant (0%) mutation vs. adaptive mutation – accuracy	2.572×10^{-214}
Constant (0%) mutation vs. adaptive mutation – convergence	6.2864×10^{-137}
Constant (20%) mutation vs. adaptive mutation – accuracy	6.10806×10^{-25}
Constant (20%) mutation vs. adaptive mutation – convergence	1.86354×10^{-10}

Table 4. Summary of statistical tests on 3-dimensional maximization results comparing the accuracies and convergence generations associated with adaptive and constant mutation rates.

T-tests	p-value
Constant (0%) mutation vs. adaptive mutation – accuracy	0
Constant (0%) mutation vs. adaptive mutation – convergence	1.1566×10^{-56}
Constant (20%) mutation vs. adaptive mutation – accuracy	0.5
Constant (20%) mutation vs. adaptive mutation – convergence	0.44759

Table 5. Summary of results for solving the Rastrigin function minimization problem using various search algorithms.

	Random	Hill climbing	Nelder-Mead	Adaptive genetic algorithm
Accuracy (%)	0.2	0.8	5.7	100
Average run time to correctly converge (s)	0.27405	0.65205	0.01852	0.27405
Average converged minimum value	0.49505	16.97789	4.76441	0
Trials	1000	1000	1000	1000

Table 6. Summary of statistical tests on the Rastrigin function minimization results comparing the accuracies associated with each search algorithm.

T-tests	p-value
Random vs. adaptive genetic algorithm – accuracy	2.53×10^{-223}
Hill climbing vs. adaptive genetic algorithm – accuracy	9.61×10^{-268}
Nelder-Mead vs. adaptive genetic algorithm – accuracy	3.04×10^{-152}

Table 7. Summary of results and statistics comparing run times for accurate and inaccurate convergences using the Hill climbing and Nelder-Mead methods.

	Convergence	Hill climbing	Nelder-Mead
Run time (s)	Accurate	0.65205	0.01852
	Inaccurate	0.59429	0.01830
Trials	Accurate	35	57
	Inaccurate	4965	943
p-value		0.13960	0.57576

Maximizing a 3-dimensional function

In the surface maximization problem, a series of unpaired, single-tailed Student's t-tests were used to test the following null hypothesis at the 0.05 level of significance: the efficiency of the 20% constant mutation rate GA will be significantly higher than that of the adaptive mutation operator GA. The null hypothesis was rejected ($p = 0.448 > 0.05$) for the 20% level. Because the constant 0% mutation rate GA does not find the global maximum, it was compared to the adaptive mutation operator on the basis of null hypothesis 1 at the 0.05 level of significance: the accuracy of the adaptive mutation operator will not be significantly higher than that of 0% constant mutation rate. The null hypothesis was rejected ($p = 0 < 0.05$). See Table 4 for a summary of the surface maximization statistics.

Accuracies, run times, and statistics for genetic algorithms with various mutation rates maximizing a 3D function

1 Data File

http://dx.doi.org/10.6084/m9.figshare.640020

Minimizing the Rastrigin function

In the Rastrigin function minimization problem, a series of unpaired, single-tailed Student's t-tests were used to test the following null hypothesis at the 0.05 level of significance: the accuracy of the adaptive mutation operator GA will not be significantly higher than those of the other search methods (Nelder-Mead, hill climbing, random). The null hypothesis was rejected for random method ($p = 2.53 \times 10^{-223} < 0.05$), for the hill climbing method ($p = 9.61 \times 10^{-268} < 0.05$), and for the Nelder-Mead method ($p = 3.04 \times 10^{-152} < 0.05$). Furthermore, a series of unpaired, single-tailed Student's t-tests were used to test if there were significant differences in the run times of the hill climbing and Nelder-Mead methods when the algorithm accurately converged and when the same algorithm did not accurately converge. Both the hill climbing method ($p = 0.140 > 0.05$) and the Nelder-Mead method ($p = 0.576 > 0.05$) showed no significant differences in run time when accurately converging compared to inaccurately converging. These results point to the possibility that these two methods were only accurate when a fortunate random point was selected to start running the search algorithm from. No extra time being needed to converge correctly versus incorrectly, points towards the idea that no extra computation was needed either. See Table 6 and Table 7 for a summary of the Rastrigin function minimization statistics.

Accuracies, run times, and statistics for search algorithms minimizing the Rastrigin function

1 Data File

http://dx.doi.org/10.6084/m9.figshare.640021

Discussion

Theory

The mutation rate operator presented in this study uses randomness to guide a genetic algorithm (GA) towards the optimal solution. A GA with less randomness leads to faster convergence towards local optima; however, by limiting randomness it limits the search space, which in turns hinders the search for the global optimum. Conversely, a GA with more randomness hinders progress towards local optima, but allows for a wider search space, aiding the search for the global optimum[8,28]. Therefore, randomness should be inversely related to the derivative of the fitness function f. Because f is implicit in most problems, the proposed heuristic attempted to look at the derivative of f with respect to generation, creating a pseudo, inversely related function to the derivative that could be used to calculate an adaptive mutation rate. The heuristic subtracted the first generation at which the most optimal fitness value appeared from the current generation to obtain a result inversely related to the derivative, and thus directly related to the randomness or mutation rate. Therefore any monotonically increasing function with domain bounded by x in $[0, \infty)$ and y in $[0, 1]$ should provide an appropriate amount of randomness to either find a local optimum or increase the search space from the local optima in order to find the global optimum.

Implications

The implications of a new, pseudoderivative-based adaptive mutation rate are considerable. The implementation of this new technique within a genetic algorithm could provide increased accuracy in optimization of all GA problems. For example, the treatment of complex diseases requires the discovery of new drug combinations that are hard to come by solely on the basis of empirical clinical knowledge. Search algorithms, including GAs, have begun to provide promising results in identifying optimal drug combinations, e.g. for destroying human cancer cells as well as for minimizing the physiological decline associated with aging. Moreover, these approaches required only one-third of the number of tests employed in the classic method for the discovery of optimal drug combinations. This approach has greatly reduced the risk and expense of clinical trials[29]. The availability of a more sophisticated class of GAs will also allow search algorithms that are based on GAs to be improved. For example, swarm-based optimization algorithms, such as the Bees Algorithm, can improve the efficiency with which optimal and suboptimal solutions can be discovered within a given search space[30].

Our results agree with results reported in other articles on adaptive mutation accuracy and efficiency[5,31]. The method described in this study also has the added benefit of low computational complexity while still being able to guide the GA out of local optima and towards the global optimum. Studies in adaptive mutation have ranged widely from individual gene-based mutations, to Gaussian operators, to polynomial operators[20]. However, to our knowledge, past adaptive mutation techniques have never used the derivative of f with respect to generation to determine a changing mutation rate. This new method of implementing adaptive mutation may open up entirely new areas for implementing search algorithms based on improved GAs.

Conclusion

GAs are powerful tools that can optimize overwhelmingly complex real-life problems, including speedy diagnoses of complex diseases. Previous GAs faced a trade-off between speed and accuracy, as more random GAs sacrificed speed in exchange for a better chance of optimization or less random ones gave up accuracy for speed. The solution of an adaptive mutation operator based on the derivative of f with respect to generation allows for increased accuracy without the loss of speed. Suggestions for research that would expand upon current findings include determining an optimal monotonically increasing function for the mutation operator, such as a sigmoid, inverse tangent, or scaled linear function, as well as comparing the current GA to other commonly used search methods in current bioinformatics problems.

Author contributions

ASR conceived the study; MSK and ASR designed the experiments; MSK and ASR performed the research; MSK and LAH analyzed the data; and MSK, ASR, and LAH wrote the paper and agreed with the final content.

Competing interests

No competing interests were disclosed.

Grant information

The author(s) declared that no grants were involved in supporting this work.

Acknowledgements

We wish to thank Dr. Shane Torbert and Arjun Ramesh for their support and participation in early discussions about formulating the idea of this study, Jessleen K. Kanwal for assisting with Figure 7 and statistical analyses, and Dr. Jagmeet S. Kanwal for providing guidance in writing and submitting this manuscript.

Supplementary materials

Genetic algorithm scripts

4 Script Files

http://dx.doi.org/10.6084/m9.figshare.640023

References

1. Loh PR, Baym M, Berger B: **Compressive genomics.** *Nat Biotechnol.* 2012; **30**(7): 627–630.
PubMed Abstract | Publisher Full Text

2. Forrest S: **Genetic algorithms: principles of natural selection applied to computation.** *Science.* 1993; **261**(5123): 872–878.
PubMed Abstract | Publisher Full Text

3. Ross P, Corne D: **Applications of genetic algorithms.** *AISB Quaterly on Evolutionary Computation.* 1994; **89**: 23–30.
Reference Source

4. Bingul Z: **Adaptive genetic algorithms applied to dynamic multiobjective problems.** *Appl Soft Comput.* 2007; **7**(3): 791–799.
Publisher Full Text

5. Tang PH, Tseng MH: **Adaptive directed mutation for real-coded genetic algorithms.** *Appl Soft Comput.* 2013; **13**(1): 600–614.
Publisher Full Text

6. Storn R, Price K: **Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces.** *Journal of Global Optimization.* 1997; **11**(4): 341–359.
Publisher Full Text

7. Deep K, Thakur M: **A new mutation operator for real coded genetic algorithms.** *Appl Math and Comput.* 2007; **193**(1): 211–230.
Publisher Full Text

8. Korejo I, Yang S, Li C: **A directed mutation operator for real coded genetic algorithms.** *Evo Applications.* 2010; **6024**: 491–500.
Publisher Full Text

9. Mühlenbein H: **How genetic algorithms really work: I. mutation and hillclimbing.** *Parallel Problem Solving From Nature.* 1992; **2**: 15–26.
Reference Source

10. Zang H, Zhang S, Hapeshi K: **A Review of Nature-Inspired Algorithms.** *J Bionic Eng.* 2010; **7**(Suppl): S232–S237.
Publisher Full Text

11. Rudolph G: **Convergence analysis of canonical genetic algorithms.** *IEEE Trans Neural Netw.* 1994; **5**(1): 96–101.
PubMed Abstract | Publisher Full Text

12. Goldberg DE: **Genetic algorithms in search, optimization, and machine learning**. 1989.
Reference Source

13. Leung Y, Gao Y, Xu ZB: **Degree of population diversity-a perspective on premature convergence in genetic algorithms and its markov chain analysis.** *IEEE Trans Neural Netw.* 1997; **8**(5): 1165–1176.
PubMed Abstract | Publisher Full Text

14. Srinivas M, Patnaik LM: **Adaptive probabilities of crossover and mutation in genetic algorithms.** *IEEE Trans Syst Man Cybem.* 1994; **24**(4): 656–667.
Publisher Full Text

15. De Falco I, Della Cioppa A, Tarantino E: **Mutation-based genetic algorithm: performance evaluation.** *Appl Soft Comput.* 2002; **1**(4): 285–299.
Publisher Full Text

16. Michalewicz Z: **Genetic Algorithms + Data Structures = Evolution Programs**. Springer-Verlag, New York, 1992.
Publisher Full Text

17. Lee HC, Yang CW, Chen CY, *et al.*: **Single point mutation of microRNA may cause butterfly effect on alteration of global gene expression.** *Biochem Biophys Res Commun.* 2011; **404**(4): 1065–1069.
PubMed Abstract | Publisher Full Text

18. Cobb HG: **An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments** (No. NRL-MR-6760) NAVAL RESEARCH LAB WASHINGTON DC. 1990.
Reference Source

19. Bäck T, Schütz M: **Intelligent mutation rate control in canonical genetic algorithms.** *Foundations of Intelligent Systems.* 1996; **1079**: 158–167 Springer Berlin Heidelberg.
Publisher Full Text

20. Uyar Ş, Sariel S, Eryiğit G: **A gene based adaptive mutation strategy for genetic algorithms.** *Genetic and Evolutionary Computation GECCO.* 2004; **3103**: 271–281.
Publisher Full Text

21. Russell SJ, Norvig P, Davis E, *et al.*: **Artificial intelligence: a modern approach (Vol. 2)**. Englewood Cliffs: Prentice hall. 2010.

22. Solis FJ, Wets RJ: **Minimization by random search techniques.** *Mathematics of Operations Research.* 1981; **6**(1): 19–30.
Publisher Full Text

23. Skalak DB: **Prototype and feature selection by sampling and random mutation hill climbing algorithms**. In *Proceedings of the eleventh international conference on machine learning.* 1994; pp. 293–301.
Reference Source

24. Nelder JA, Mead R: **A simplex method for function minimization.** *Comput J.* 1965; **7**(4): 308–313.
Publisher Full Text

25. Stone HS, Stone JM: **Efficient search techniques–an empirical study of the N-Queens problem.** *IBM Journal of Research and Development.* 1987; **31**(4): 464–474.
Publisher Full Text

26. Bell J, Stevens B: **A survey of known results and research areas for *n*-queens.** *Discrete Math.* 2009; **309**(1): 1–31.
Publisher Full Text

27. Karaboga D, Basturk B: **A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm.** *J Global Optim.* 2007; **39**(3): 459–471.
Publisher Full Text

28. Mitchell M, Holland JH, Forrest S: **When will a genetic algorithm outperform hill climbing?** *Adv Neural Inf Process Syst.* 1993; 51–58.
Reference Source

29. Calzolari D, Bruschi S, Coquin L, *et al.*: **Search algorithms as a framework for the optimization of drug combinations.** *PLoS Comput Biol.* 2008; **4**(12): e1000249.
PubMed Abstract | Publisher Full Text | Free Full Text

30. Pham DT, Ghanbarzadeh A, Koc E, *et al.*: **The bees algorithm–a novel tool for complex optimisation problems.** In Proceedings of the 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006). 2006; pp. 454–459.
Reference Source

31. Yang JM, Kao CY: **Integrating adaptive mutations and family competition into genetic algorithms as function optimizer.** *Soft Computing.* 2000; **4**(2): 89–102.
Publisher Full Text

Open Peer Review

Current Referee Status: ☑ ☑ ☑

Version 2

Referee Report 12 March 2014

doi:10.5256/f1000research.2353.r2563

☑ **Christos Ouzounis**
Computational Genomics Unit and Institute of Agrobiotechnology, National Center for Research & Technology, Thessalonica, Greece

The authors have addressed many of the concerns raised.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.

Referee Report 10 February 2014

doi:10.5256/f1000research.2353.r2562

☑ **Panayiotis (Takis) Benos**
Department of Computational Biology, University of Pittsburgh, Pittsburgh, PA, USA

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.

Referee Report 27 November 2013

doi:10.5256/f1000research.2353.r2561

☑ **Mark Ragan**
Institute for Molecular Bioscience, University of Queensland, Brisbane, Australia

The authors have addressed my concerns, particularly, in regards to the wording of the title.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.

Version 1

Referee Report 22 July 2013

doi:10.5256/f1000research.1167.r1179

Panayiotis (Takis) Benos
Department of Computational Biology, University of Pittsburgh, Pittsburgh, PA, USA

This article primarily focuses on a very interesting computational problem, namely whether the variable-rate mutation operators in a genetic algorithm offers an advantage over the fixed-rate ones. The main idea is that variable-rate operators will help algorithms escape the local optima and identify the global optimum more efficiently. The authors indeed show that this is the case in at least some of the examples they investigated. I think this is a very promising article that will generate a lot of interest in this area.

However, I am a bit unsure about the method’s relevance to biology. There are some biological examples that I can think of where their methodology can make an impact but I fail to see how it can be used for “diagnosis” and “therapy” as the title states. I wish they had provided some more examples towards that direction.

Finally, I would like to bring up a point that is frequently overseen in computational studies. Although in many non-biological problems finding the optimal solution is the ultimate goal, many biological problems are better served by finding a set of optimal+ sub-optimal solutions. Examples include protein and RNA folding, and microRNA and transcription factor binding sites. Perhaps the authors would consider extending their methodology towards that direction in the future.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Competing Interests: No competing interests were disclosed.

Author Response 19 Nov 2013

Maxinder Kanwal, Thomas Jefferson High School for Science and Technology, USA

We agree with the reviewer’s point. Our main objective was to show the advantage variable-rate mutation operators in genetic algorithms had over fixed-rate ones when applied correctly. While some of our immediate visions for the algorithm were for applying it towards solving problems in bioinformatics, we realize that the algorithm is more of a general search method that can be applied to a number of problems not just within biology. In order to more accurately portray the content of this article, the title and abstract have been modified to avoid emphasizing a minor aspect of the article. In addition, the Implications section of the Discussion have been expanded on to more comprehensively explain how this algorithm can be applied in the medical field as an example of one of its many applications.

In regards to your comment, you bring up a very good point. We will certainly look into modifying this algorithm to make it a technique that better serves attempting to solve problems in biology by having it search for a set of optimal and suboptimal solutions.

Competing Interests: No competing interests were disclosed.

Referee Report 15 July 2013

doi:10.5256/f1000research.1167.r1062

☑ **Mark Ragan**
Institute for Molecular Bioscience, University of Queensland, Brisbane, Australia

Genetic algorithms offer a biologically motivated approach to nonlinear optimisation. In the so-called breeding stage of a genetic algorithm, the diversity of candidate solutions is increased by operations analogous to crossover between chromosomes, and point mutations. If parameterised appropriately, the mutation operator helps the genetic algorithm explore the fitness landscape, in particular by making it less likely to become trapped in a local optimum. Mutation operators are typically of fixed rate, although there is some prior art in the use of variable-rate operators. Here Kanwal, Ramesh and Huang, senior students at Thomas Jefferson High School for Science and Technology in Alexandria, Virginia, develop some theory on the form of adaptive-rate mutation operators, and investigate the performance of a new heuristic motivated by this theory. They report that this heuristic delivers much-improved accuracy in finding the global optimum, at no cost in speed, in two reasonably difficult optimisation scenarios, and describe bounds on the mutation function.

The research is clearly of publishable quality. The content explained clearly and in sufficient detail to allow the work to be replicated, and the conclusions are balanced and justified on the basis of the results presented. The writing style is exemplary. However, the manuscript does not deliver against the “diagnosis and therapy” promised in the title and abstract: indeed, the authors rightly allude to much broader contexts. In the absence of a specific case study in the biological application domain, the title might focus solely on the algorithmic contribution. It would be interesting to compare the performance of genetic algorithms implementing their new mutation operator with, for example, simulated annealing or multi-chain MCMC, although this would presumably require a broader comparison framework than was within-scope for the work reported here.

Figures 3 and 4 are very similar to those in Wikipedia (http://en.wikipedia.org/wiki/Rastrigin_function, accessed 13 July 2013); does this arise from use of a generic parameterisation?

I would prefer “optima” rather than “optimums”. There is a misspelling in the legend of Figure 3.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.

Author Response 19 Nov 2013

Maxinder Kanwal, Thomas Jefferson High School for Science and Technology, USA

We agree with the reviewer's point. Our main objective was to show the advantage variable-rate mutation operators in genetic algorithms had over fixed-rate ones when applied correctly. While some of our immediate visions for the algorithm were for applying it towards solving problems in bioinformatics, we realize that the algorithm is more of a general search method that can be applied to a number of problems not just within biology. In order to more accurately portray the content of this article, the title and abstract have been modified to avoid emphasizing a minor aspect of the article. In addition, the Implications section of the Discussion have been expanded on to more comprehensively explain how this algorithm can be applied in the medical field as an example of one of its many applications.

In regards to Figures 3 and 4 being similar to those on Wikipedia, the reason for this is because the code given below the images was almost the same code used to generate our figures in the paper with only slight modifications to make the figures more readable. We were aware of the Wikipedia images before generating our own; we simply liked the parameters set in the Wikipedia code, and so we generated our images using that code (along with some additional style changes).

The mistakes pointed out have been fixed as well.

Competing Interests: No competing interests were disclosed.

Referee Report 12 July 2013

doi:10.5256/f1000research.1167.r1057

?

Christos Ouzounis

Computational Genomics Unit and Institute of Agrobiotechnology, National Center for Research & Technology, Thessalonica, Greece

Genetic algorithms (GAs) represent an adaptive computation technique with applications to various optimization problems, borrowing ideas from biological systems – with solution populations and operators that mimic DNA mutation and selection. As the authors of this report mention, GAs can suffer from issues of convergence to local optima, as all optimization algorithms do. It is proposed that a variable mutation rate (as opposed to the more classical approach of a constant mutation rate) can contribute towards better optimization performance, away from local optima: the longer the GA remains at a local optimum, the higher the mutation rate can become. This is the key idea that is addressed in this work.

The study is interesting as it focuses on the mutation operator which has not been researched sufficiently, compared to the crossover operator which has been studied quite extensively. The proposed solution - namely a variable mutation rate as an independent variable, at least in one case - is compared to other optimization techniques, such as random search / hill climbing / Nelder-Mead. Experiments are performed for well-established, highly controlled optimization problems, such as the N-Queens problem, maximization of a 3D surface and the Rastrigin function. Performance in terms of both accuracy of solution and efficiency of resource consumption was monitored for all experiments. Interestingly, while adaptive mutation has a two-fold performance in both terms above for the N-Queens problem, there is no significant difference for the 3D surface problem.

This is interesting work, but relevance to computational biology and real-world solutions is somewhat peripheral. I am sure the authors are investigating applications of their approach to larger problems, as alluded in their title - which unfortunately is somewhat misleading: without the terms ‘diagnosis’ and ‘therapy’, this would have been a fine, technical report on an exciting subject. Hopefully, we will see some further development towards that direction in the near future.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Competing Interests: No competing interests were disclosed.

Author Response 19 Nov 2013

Maxinder Kanwal, Thomas Jefferson High School for Science and Technology, USA

We agree with the reviewer’s point. Our main objective was to show the advantage variable-rate mutation operators in genetic algorithms had over fixed-rate ones when applied correctly. While some of our immediate visions for the algorithm were for applying it towards solving problems in bioinformatics, we realize that the algorithm is more of a general search method that can be applied to a number of problems not just within biology. In order to more accurately portray the content of this article, the title and abstract have been modified to avoid emphasizing a minor aspect of the article. In addition, the Implications section of the Discussion have been expanded on to more comprehensively explain how this algorithm can be applied in the medical field as an example of one of its many applications.

Competing Interests: No competing interests were disclosed.