

A Genetic Algorithm with randomly shifted Gray codes and local optimizations based on quadratic approximations of the fitness

Alexandre Mayer
University of Namur
Rue de Bruxelles 61
Namur, Belgium 5000
alexandre.mayer@unamur.be

ABSTRACT

We present a Genetic Algorithm that we developed in order to address computationally expensive optimization problems. In order to accelerate this algorithm, we establish, generation after generation, quadratic approximations of the fitness in the close neighborhood of the best-so-far individual. We then inject in the population an individual that corresponds to the optimum of this approximation. We also introduce a modified mutation operator that acts on randomly-shifted Gray codes. We show that these techniques lead to the global optimum of typical benchmark problems in 5, 10 and 20 dimensions with a probability of success in one run of the order of 95-97% and an average number of fitness evaluations of the order of $400 - 750 \times n$, where n refers to the dimension of the problem.

CCS CONCEPTS

•Computing methodologies → Genetic algorithms; •Theory of computation → Evolutionary algorithms; Bio-inspired optimization; Convex optimization;

KEYWORDS

Genetic algorithms, shifted Gray codes, modified mutation operator, local optimization, quadratic approximation

ACM Reference format:

Alexandre Mayer. 2017. A Genetic Algorithm with randomly shifted Gray codes and local optimizations based on quadratic approximations of the fitness. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017*, 2 pages.

DOI: <http://dx.doi.org/10.1145/3067695.3075968>

1 INTRODUCTION

For optimization problems that are computationally expensive, it is desirable to determine the global optimum ideally by a single run of the genetic algorithm and with a reduced number of fitness evaluations. We can guide the algorithm to promising directions and accelerate the refinement of the final solution by analyzing, generation after generation, the data collected by the algorithm (Memetic Algorithms). Despite this guidance, the ability of the genetic algorithm to make appropriate transitions in the decision variable

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO '17 Companion, Berlin, Germany

© 2017 Copyright held by the owner/author(s). 978-1-4503-4939-0/17/07...\$15.00

DOI: <http://dx.doi.org/10.1145/3067695.3075968>

space can remain artificially difficult if these transitions require too many bit changes. This issue can be addressed by introducing a modified mutation operator that acts on randomly-shifted versions of the Gray code. These ideas are developed in this article.

2 DESCRIPTION OF THE ALGORITHM

We want to determine the global minimum of $f = f(x_1, \dots, x_n)$, where $x_i \in [x_i^{\min}, x_i^{\max}]$ with a discretization step Δx_i . The variables are represented by $x_i = x_i^{\min} + \langle \text{gene } i \rangle \times \Delta x_i$, where $\langle \text{gene } i \rangle \in [0, 2^{n_i} - 1]$ with n_i the number of bits in a gene and $n_{\text{bits}} = \sum_{i=1}^n n_i$ the total number of bits in a DNA. We use the Gray code as reference encoding scheme. A description of our algorithm can be found in [1, 2]. We work here with a population of $n_{\text{pop}}=50$ individuals. We start with random individuals. We evaluate their fitness and sort the population from the best individual to the worst. The worst $n_{\text{rand}} = 0.1 \times n_{\text{pop}} \times (1 - p)$ individuals are replaced by random individuals ($p = |s - 0.5|/0.5$, with s the fraction of bits in the population whose value is identical to the best individual); they are transferred to the next generation. The remaining $N = n_{\text{pop}} - n_{\text{rand}}$ individuals participate to the steps of selection, crossover and mutation. We select N parents by a rank-based roulette wheel selection. For any pair of parents, we define two children for the next generation either (i) by a one-point crossover of the parents' DNA (probability of 70%), or (ii) by a simple replication of the parents. The children obtained by crossover are subjected to a *modified mutation operator*, using $m = 0.95/n_{\text{bits}}$ as mutation rate. We apply at this point a *local optimization* step. We finally evaluate the fitness of all individuals scheduled for the next generation. We sort the population, apply elitism and continue iteratively until a termination criteria is met. We keep a record with all fitness evaluations in order to avoid any duplication of these evaluations.

Local Optimization: We select collected data points \vec{x} for which $\max_i \frac{|x_i - x_{i,\text{ref}}|}{\Delta x_i} \leq W$, where \vec{x}_{ref} refers to the best-so-far solution and W to the width of this selection. We then approximate these data by $f(\vec{x}) = a_0 + \vec{A}_1 \cdot \vec{X} + \frac{1}{2} \vec{X} \cdot \mathbf{A}_2 \vec{X}$, where $\vec{X} = \Delta^{-1}(\vec{x} - \vec{x}_{\text{ref}})$ and $\Delta = \text{diag}[\Delta x_1, \dots, \Delta x_n]$. The coefficients a_0 , \vec{A}_1 and \mathbf{A}_2 are determined by a singular-value-decomposition method [2]. W ensures that the number of data considered is at least twice the number of coefficients to determine. The solution of $\vec{\nabla} f = 0$ is then formally given by $\vec{x}^* = \vec{x}_{\text{ref}} - \Delta \mathbf{A}_2^{-1} \vec{A}_1 = \vec{x}_{\text{ref}} - \Delta \sum_k \frac{\vec{x}_k \cdot \vec{A}_1}{\lambda_k} \vec{x}_k$, where $\mathbf{A}_2 \vec{x}_k = \lambda_k \vec{x}_k$ is the eigensystem of \mathbf{A}_2 . In order to control numerical errors, we use only eigenvalues λ_k for which $|\lambda_k| \geq 10 \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} \lambda_{\text{max}} \epsilon$, where $\epsilon = 10^{-10}$ is the relative accuracy of the fitness. If \vec{x}^* can

Table 1: Test functions used for the benchmarking.

$f_1 = \sum_{i=1}^n x_i^2, x_i \in [-5.12, 5.12], \Delta x_i = 0.01$
$f_2 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2, x_i \in [-65.5, 65.5], \Delta x_i = 0.1$
$f_3 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2],$ $x_i \in [-2.05, 2.05], \Delta x_i = 0.0025$
$f_4 = n(x_1 - 1)^2 + \sum_{i=2}^n (2x_i^2 - x_{i-1})^2, x_i \in [0, 10], \Delta x_i = 0.0025$
$f_5 = -\prod_{i=1}^n \cos(x_i)^2 \exp(-x_i^2/10), x_i \in [-5, 5], \Delta x_i = 0.01$
$f_6 = -\sum_{i=1}^n x_i \sin(\sqrt{ x_i }), x_i \in [-500, 500], \Delta x_i = 1$
$f_7 = \sin^2(\pi w_1) + \sum_{i=1}^{n-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)]$ $+ (w_n - 1)^2 [1 + \sin^2(2\pi w_n)], w_i = 1 + (x_i - 1)/4,$ $x_i \in [-10, 10], \Delta x_i = 0.01$
$f_8 = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)), x_i \in [-5, 5], \Delta x_i = 0.01$
$f_9 = -a \exp\left(-b \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(cx_i)\right) + a + e,$ $a = 20, b = 0.2, c = 2\pi, x_i \in [-32.8, 32.8], \Delta x_i = 0.025$
$f_{10} = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(x_i/\sqrt{i}),$ $x_i \in [-600, 600], \Delta x_i = 0.25$

be accepted, it replaces the last individual scheduled for the next generation. We try otherwise up to 3 times by increasing W .

Modified mutation operator: The displacements generated by mutations depend artificially on the coding considered. In order to allow a wider range of displacements and help the genetic algorithm to escape local optima, we apply mutations on the encoding obtained with a randomly-shifted version of the Gray code [4]. The result is then translated back to the original Gray code (reference encoding). The shift introduced in the Gray code is specific to each gene. It is chosen in the range $[0, 2^{n_i} - 1]$. It is identical for all individuals in the population. It is reset at each generation.

3 RESULTS

We are interested in a class of problems for which $(x_i^{\max} - x_i^{\min})/\Delta x_i \sim 1000$. We also consider that the global optimum is found if the objective function is within a margin Δf_{target} of 10^{-4} compared to the exact solution [1, 2]. Our objective is to determine the global optimum with a high chance of success in one run and with a reduced number of fitness evaluations. In order to test our algorithm in these conditions, we consider typical benchmark problems in 5, 10 and 20 dimensions (see Table 1). We run the algorithm 500 times on each test function in order to measure the probability $P(\Delta f_{\text{target}})$ with which the target Δf_{target} is reached by a single run. We also measure the average number of fitness evaluations $\langle n_{\text{eval}} \rangle$ required to reach this target [3]. We use for this testing a maximum of $30 \times n_{\text{bits}}$ generations. The algorithm is interrupted if (i) there is no improvement of the best fitness in the last $1.5 \times n_{\text{bits}}$ generations, (ii) the mean value of the genetic similarity s over the last $1.5 \times n_{\text{bits}}$ generations is higher than $1 - 3m$, (iii) $s \geq 1 - m$, or (iv) the number of fitness evaluations exceeds $10,000 \times n$. The results are presented in Table 2. The different columns indicate the results obtained when considering/not considering (i) local optimizations based on quadratic approximations of the fitness, and (ii) a modified mutation operator that acts on randomly-shifted Gray codes.

Table 2: Benchmark results. $P(\Delta f_{\text{target}})$ represents the probability to reach the target $\Delta f_{\text{target}} = 10^{-4}$ by a single run. $\langle n_{\text{eval}} \rangle$ is the average number of fitness evaluations required to reach this target. #fct($P \geq 10\%$) is the number of functions for which the target was reached at least once in ten runs.

Local Optimizations		no	no	yes	yes
Gray code shifted		no	yes	no	yes
$n = 5$	$P(\Delta f_{\text{target}})$	60.1%	76.6%	79.9%	95.5%
	$\langle n_{\text{eval}} \rangle$	12,430	10,587	2,150	1,812
	#fct($P \geq 10\%$)	8/10	8/10	all	all
$n = 10$	$P(\Delta f_{\text{target}})$	39.3%	72.4%	65.8%	97.2%
	$\langle n_{\text{eval}} \rangle$	47,895	27,722	6,713	4,344
	#fct($P \geq 10\%$)	5/10	8/10	8/10	all
$n = 20$	$P(\Delta f_{\text{target}})$	25.3%	72.8%	50.6%	96.7%
	$\langle n_{\text{eval}} \rangle$	175,439	67,119	28,983	14,819
	#fct($P \geq 10\%$)	3/10	8/10	6/10	all

4 CONCLUSIONS

This article shows that the performance of genetic algorithms can be boosted by local optimizations based on quadratic approximations of the fitness¹ and by using a modified mutation operator that acts on randomly-shifted Gray codes. The results show that a target accuracy of 10^{-4} on the global optimum of the functions considered can be reached with a probability of success in one run of the order of 95-97%, instead of 60% ($n=5$), 39% ($n=10$) and 25% ($n=20$) when these techniques are not used. These results were tested for statistical significance.² The average number of fitness evaluations required to reach this target accuracy is around $400 - 750 \times n$, which compares well with state-of-the-art algorithms.³

ACKNOWLEDGMENTS

A.M. is funded by the Fund for Scientific Research (F.R.S.-FNRS) of Belgium. He is member of NaXys, Namur Center for Complex Systems, University of Namur, Belgium. This research used resources of the "Plateforme Technologique de Calcul Intensif (PTCI)" (<http://www.ptci.unamur.be>) located at the University of Namur, Belgium, which is supported by the F.R.S.-FNRS under the convention No. 2.5020.11. The PTCI is member of the "Consortium des Equipements de Calcul Intensif (CECI)" (<http://www.ceci-hpc.be>).

REFERENCES

- [1] A. Mayer and A. Bay. 2015. Optimization by a genetic algorithm of the light-extraction efficiency of a GaN light-emitting diode. *Journal of Optics* 17 (2015), 025002.
- [2] A. Mayer, J. Muller, A. Herman, and O. Deparis. 2015. Optimized absorption of solar radiations in nano-structured thin films of crystalline silicon via a genetic algorithm. *Proceedings of SPIE* 9546 (2015), 95461N.
- [3] P. Posik, W. Huyer, and L. Pal. 2012. A comparison of global search algorithms for continuous black box optimisation. *Evolutionary Computation* 20, 4 (2012), 509-541.
- [4] J. Rowe, D. Whitley, L. Barbulescu, and J.-P. Watson. 2004. Properties of Gray and Binary Representations. *Evolutionary Computation* 12, 1 (2004), 47-76.

¹This is a particular example of "surrogate modeling"; see also "response surface methodology" for other approximations.

² $H_0 = "P(\Delta f_{\text{target}})$ not improved" rejected at $\alpha = 10^{-5}$ by a right-tailed z-test.

³CMA-ES on the same problems achieves $P(\Delta f_{\text{target}}) = 60.2\%$ ($n=5$), 60.0% ($n=10$) and 58.7% ($n=20$), with $\langle n_{\text{eval}} \rangle = 2,034$ ($n=5$), $4,570$ ($n=10$) and $10,988$ ($n=20$).