

# An Analysis of the Solution Quality of the Simple Genetic Algorithm

HALDUN AYTUG

ANAND PAUL

Information Systems and Operations Management Department

Warrington College of Business

University of Florida

STZ 351

Gainesville, FL 32611

USA

aytugh@ufl.edu <http://warrington.ufl.edu/isom>

paulaa@ufl.edu <http://warrington.ufl.edu/isom>

## Abstract:

We introduce a probabilistic worst case bound on the solution quality for the simple genetic algorithm. Given a probability and number of iterations a simple genetic algorithm has performed we show how to bound the distance between the best solution found and the optimal. We introduce several concepts needed for the analysis. The ideas are illustrated on several well-known functions used in the analysis of genetic algorithms.

*Keywords – Genetic algorithms, heuristic search, worst-case analysis, error bounds, run-time complexity, Markov Chains*

## 1 Introduction

During the last two decades there has been substantial work on modelling the behaviour of GAs as dynamical systems [13]. Vose and Liepins [12] first developed the idea of punctuated equilibria and derived asymptotic properties for the simple, infinite population GA. Nix and Vose [9] later showed that this behaviour can be modelled for finite populations as a Markov Chain and provided a way to compute transition probabilities.

Since the earlier work by Vose and his co-authors several bounds have been developed ([1], [2], [3], [7], [16]). The general conclusion from this body of work is that in the worst case GA behaves like random search even in the presence of multiple-optima.

There has been some research on developing probabilistic bounds on the quality of the solution obtained by heuristic search on combinatorial problems. For example, Golden and Alt [6] show how to construct a confidence interval for an optimal solution and report successful results for the Traveling Salesman Problem. They use the limiting distribution of the minimum of a sample (i.e., the best solution). Ovacik, Rajagopalan and Uzsoy [11] use several procedures on single machine scheduling problems. Nydick and Weiss [10] use jackknife-based estimators that combine any heuristic based solution with the two best random

solutions. Giddings et al. [5] give an excellent review of such methods.

In this work, we describe how one can make probabilistic statements about the quality of the solution obtained by a GA and provide bounds for a variety of function classes discussed in the literature. Unlike methods described in the previous paragraph our results are distribution free and dependent on the number of iterations of the GA. The main thrust of this work is to present a foundation for the analysis of a worst-case error bound. We illustrate its use on some well-known functions extensively used to test GAs.

## 2 The Simple Genetic Algorithm

Let  $\Omega$  be the set of binary strings of length  $\ell > 1$  over which we wish to maximize a function  $f: \Omega \rightarrow \mathbb{R}^+$  using SGA. Unless otherwise specified we will assume that  $f$  has a unique optimum. We represent  $x \in \Omega$  either as a bit string (e.g.,  $x = 010110$ ) or as the integer it represents (i.e.,  $x = 22$ ).  $\oplus$  and  $\otimes$  represent bitwise “exclusive-or” and “and” operators respectively.  $|x|$  is the number of nonzero bits of string  $x$ .  $x^t$  represents the best solution after  $t$  iterations.  $x^*$  is the optimal solution. We define  $O^x = \{y \mid |y \oplus x| = 1, y \in \Omega\}$  to be the

set of strings that are within one bit of  $x$ .  $O_K^x$  is a class composed of subsets of  $O^x$  of size  $K-1$  each unioned with  $x$ . That is  $O_K^x = \{O \cup \{x\} \mid O \subseteq O^x, |O| = K-1\}$ . Finally,  $O_K$  is the union of all  $O_K^x$ ,  $O_K = \bigcup_{x \in \Omega} O_K^x$ .

It is important to explain the sets  $O_K^x$  and  $O_K$  as they are crucial to the understanding of a key theorem by Aytug and Koehler [2] that will be presented next. Given a string  $x \in \Omega$ , class  $O_K^x$  includes all sets of size  $K$  that include string  $x$  and  $K-1$  other strings that are within one bit of  $x$ . For example for  $K=2$ ,  $\ell=3$  and  $x=1$

$$O_2^1 = \{0, 3, 5\} = \{000, 011, 101\} \quad \text{so}$$

$O_2^1 = \{\{1, 0\}, \{1, 3\}, \{1, 5\}\}$ .  $O_K$  consists of sets that are composed of members of  $O_K^x$  for all  $x$ , (i.e.,

$$O_K = O_K^0 \cup O_K^1 \cup \dots \cup O_K^{2^\ell-1}).$$

We will limit our discussion to simple GA's involving one-point crossover and uniform mutation. After creating a random population SGA operates by applying crossover and mutation at each iteration until stopped [13].

Nix and Vose, [9], show that a GA can be modeled as a Markov Chain and derive the transition probabilities for SGA. Based on this model Aytug and Koehler, [1], derived a worst case bound on the number of iterations needed to find one of  $K$  solutions (for example, one of  $K$  multiple optimal solutions). They show that in the worst case these  $K$  solutions are members of the set  $O^{x^*}$ .

Let  $t_K$  be the number of iterations required to make sure that the SGA has seen one of the  $K$  pre-specified solutions with probability at least  $\alpha$ . Then the following holds:

**Theorem 1 (Theorem 2.1 of Aytug and Koehler, [1])**

$t_K$  is bounded above by

$$t_K \leq \left\lceil \frac{\ln(1-\alpha)}{n \ln \left( 1 - \min \left[ \left( (K-1)\mu^{\ell-1} + (K-2)\mu^\ell \right), \left( K(1-\mu)^{\ell-1} \mu \right) \right] \right) \right\rceil$$

where  $K \leq \ell+1$  when  $\mu \leq 0.5$  and  $2 \leq K \leq \ell$  when  $\mu > 0.5$ .

It is important to emphasize that Aytug and Koehler showed this bound is valid for any  $K$  distinguished strings (which are usually thought of as strings having a fitness value close to the optimal fitness but may be distinguished in some other manner) and

the bound is tight for all  $O_K^x$ . Theorem 1 bounds the number of iterations necessary to visit states that include strings that are one bit off of a baseline string (and consequently have two differing bits). For example, for  $K=2$  and  $\ell=3$  strings 0, 1, 2, and 4 have this property. If 0 is the baseline string all other strings are one bit off of 0.  $t_K$  guarantees (with probability  $\alpha$ ) that the SGA has seen populations that include 0 or 1, and 0 or 2, and 0 or 4. In other words, it guarantees that the GA has seen at least one member of the sets  $\{0,1\}$ ,  $\{0,2\}$ ,  $\{0,4\}$ . It is this behaviour that we will use to derive our results.

### 3 Worst Case Error Bounds

We now develop a probabilistic bound on the difference between the optimal value of the objective and the best-known value after  $t_K$  iterations. Precisely we are interested in finding an  $\varepsilon$  such that

$$P\{f(x^*) - f(x^{t_K}) \leq \varepsilon\} \geq \alpha.$$

This type of analysis is called a post analysis in the sense that it bounds the difference in objective values between the best-known solution and any optimal solution after the search has been conducted.

We now show how the bound developed by Aytug and Koehler [1] can be used to estimate this error when we stop the SGA after  $t_K$  iterations. For the sake of clarity we show our results for  $2 \leq K \leq \ell+1$ . Cases where  $K > \ell+1$  follow a similar line of reasoning (as can be seen using arguments in [1]).

**Theorem 2** With probability  $\alpha$ , if

$$\varepsilon_K = \min_{O \in O_K^{x^*}} \max_{x \in O} \{f(x^*) - f(x)\}, \quad 2 \leq K \leq \ell+1$$

then  $\varepsilon_2 \leq \varepsilon_3 \leq \dots \leq \varepsilon_K$ .

**Proof:**

For a given  $K$  let  $\bar{O}_K = \arg \max_{O \in O_K^{x^*}} \min_{y \in O} \{f(y)\}$ . For simplicity first assume  $\bar{O}_K$  is unique for each  $K$ .

For a given  $K$ ,  $\bar{O}_K$  has the strings with the best  $K-1$  objective values of  $O^{x^*}$  that are within one bit of  $x^*$  plus string  $x^*$ . Then  $\bar{O}_K \subset \bar{O}_{K+1}$  and  $y \in \bar{O}_{K+1} / \bar{O}_K$  is the  $K^{\text{th}}$  best string that is within one bit of  $x^*$ . It follows that,  $\min_{x \in \bar{O}_K} \{f(x)\} \geq f(y)$

and  $\varepsilon_K \leq \varepsilon_{K+1}$ . Repeating this for all  $K > 1$  concludes the proof for the case when  $\bar{O}_K$  is unique.

In the event that  $\bar{O}_K$  is not unique  $\bar{O}_{K+1}$  may not be unique either. However, for each alternative  $\bar{O}_K$  there is a matching  $\bar{O}_{K+1}$  such that  $\bar{O}_K \subset \bar{O}_{K+1}$  still yields the desired result. In this case,  $\varepsilon_K = \varepsilon_{K+1}$  holds for some  $K_1 \leq K \leq K_2$  depending on how many solutions with the same objective function value exist. This completes the proof.  $\square$

Theorem 2 states that the larger the  $K$  the worse the provable difference may get. Note however, Theorem 1 shows that the larger the  $K$  the smaller the number of iterations to guarantee the bound with probability  $\alpha$ .

When  $\bar{O}_K$  is not unique it is sufficient to see a member of only one of these sets to realize the same error bound, which in turn suggests that  $t_K$  of Theorem 1 overestimates the number of iterations needed. Quantifying the exact number of iterations when such information is available is beyond the scope of this paper. We now present one of the main results of this paper.

**Corollary 1:**

With probability  $\alpha$ ,  $f(x^*) - f(x^t) \leq \varepsilon_K$  only if  $t \geq t_K$ .

**Proof:**

By definition  $t_K$  is the number of iterations necessary to see one of  $K$  specified strings in all sets of size  $K$  and  $x^{t_K}$  is the best string obtained. Since Theorem 1 guarantees that at least one member of  $\bar{O}_K$  (defined in Theorem 2) has been seen after  $t_K$  iterations the result follows.  $\square$

To determine the quality of the best string after  $t$  iterations one needs to use Theorem 1 to compute  $K$ . Once  $K$  is known Theorem 2 describes how far the string is from the optimal. Corollary 1 formalizes this procedure.

In this section we will analyze some of the common classes of functions used in the literature to test GAs. We refer the interested reader to Digalakis and Margaritis [4], Mitchell and Forrest [8], and Wegener and Witt [14] among others for a list of such functions. The aim is not to do an exhaustive proof of bounds on all functions studied but rather to illustrate the use of the concept of the worst-case for a GA search. After Corollary 2 we will only prove the results for  $K=2$ , however all results generalize to  $K \leq \ell + 1$  simply by finding the  $K^{\text{th}}$  best string (including the optimal) in the same manner we found the  $2^{\text{nd}}$  best string.

### 3.1 Quadratic Pseudo-Boolean Functions (QPBF)

Wegener and Witt [14] define a quadratic pseudo-boolean function to be

$$f(x) = a_0 + \sum_{i=1}^{\ell} a_i x_i + \sum_{i=1}^{\ell} \sum_{j=i+1}^{\ell} a_{ij} x_i x_j, \quad a_i, a_{ij} \in \mathbb{Z}.$$

Corollary 2 below characterizes the runtime behavior.

**Corollary 2** For all QPBF, the best string,  $x^{t_2}$ , the GA has obtained after  $t_2$  iterations satisfies

$$(1) \quad f(x^*) - f(x^{t_2}) \leq \min_{k < \ell} \left( a_k + \sum_{i=1}^{\ell} a_{ki} x_i \right) \quad \text{with}$$

probability  $\alpha$  where  $x^*$  is the optimal string.

$$(2) \quad \text{Moreover for } K \leq \ell + 1$$

$f(x^*) - f(x^{t_K}) \leq z_{[K-1]}$  where  $z_{[K-1]}$  is the  $K-1^{\text{st}}$  largest  $z$  when sorted in ascending order, where

$$z_k = \left| a_k + \sum_{i=1}^{\ell} a_{ik} x_i \right|$$

**Proof:**

Since we have seen all strings that are within one bit of the optimal, the second best string has one bit with a positive coefficient set to zero or a bit with a negative coefficient set to one. First observe that  $x^{t_2} = x^* \oplus 2^k$ ,  $\exists k \leq \ell$  and

$$\begin{aligned} f(x^*) - f(x^{t_2}) &= \left[ \sum_{i \neq k} a_i x_i^* + a_k x_k^* + \sum_{i \neq k} \sum_{j=i+1 \neq k} a_{ij} x_i^* x_j^* \right. \\ &\quad \left. + x_k^* \sum_i a_{ij} x_i^* - \sum_{i \neq k} a_i x_i^* - a_k (x_k^* \oplus 1) \right. \\ &\quad \left. - \sum_{i \neq k} \sum_{j=i+1 \neq k} a_{ij} x_i^* x_j^* - (x_k^* \oplus 1) \sum_i a_{ij} x_i^* \right] \\ &= a_k (x_k^* - (x_k^* \oplus 1)) + (x_k^* - (x_k^* \oplus 1)) \sum_i a_{ki} x_i^* \\ &= (x_k^* - (x_k^* \oplus 1)) \left( a_k + \sum_i a_{ki} x_i^* \right) \end{aligned}$$

$$\text{then } f(x^*) - f(x^{t_2}) = \min_k \left| a_k + \sum_i a_{ki} x_i^* \right| \text{ since}$$

$$(x_k^* - (x_k^* \oplus 1)) \in \{1, -1\}.$$

(2) This follows from Theorem 2.  $\square$

**Remark:**

One of the simplest functions that is commonly studied is the bitwise linear function which is a special case of the QPBF with all  $a_{ij} = 0$ . Consequently the worst case bound in Corollary 2

reduces to  $f(x^*) - f(x^{t_2}) \leq \min_{k < \ell} (|a_k|)$  for bitwise linear functions.

### 3.2 Linear Separable Functions (LSF),

Wright and Zhao [16] define a linear separable function as follows: Let  $m_i$  be a binary mask (i.e., a string of 0,1) such that  $m_1 \oplus \dots \oplus m_B = 2^\ell - 1$  and  $m_i \otimes m_j = 0, i \neq j$ . A function  $f$  is separable over masks specified by  $m_1, \dots, m_B$  if

$$f(x) = \sum_{i=1}^B f(x \otimes m_i).$$

When  $f(x \otimes m_i) = a_i(x \otimes m_i)$ ,  $a_i \in \mathbb{R}$ ,  $f$  is a linear separable function.

**Corollary 3** Let  $f$  be a linear separable function. Then

$$f(x^*) - f(x^{t_2}) \leq \min_{i, h: m_i \otimes 2^h = 2^h} |a_i 2^h|, i \leq B, h < \ell$$

after  $t_2$  iterations.

**Proof:**

First observe that  $x^{t_2} = x^* \oplus 2^h, 0 \leq h < \ell$ . Then

$$\begin{aligned} f(x^*) - f(x^{t_2}) &= \sum_{i=1}^B (f(x^* \otimes m_i) - f((x^* \oplus 2^h) \otimes m_i)) \\ &= \sum_{i=1}^B (f(x^* \otimes m_i) - f((x^* \otimes m_i) \oplus (m_i \otimes 2^h))) \\ &= \sum_{\substack{i=1 \\ i \neq k}}^B (f(x^* \otimes m_i) - f(x^* \otimes m_i)) + f(x^* \otimes m_k) \\ &\quad - f(x^* \otimes m_k \oplus 2^h) \\ &= f(x^* \otimes m_k) - f(x^* \otimes m_k \oplus 2^h) \end{aligned}$$

for a given  $k$  and  $h$ . Since  $f$  is linearly separable

$$\begin{aligned} f(x^*) - f(x^{t_2}) &= a_k(x^* \otimes m_k) - a_k(x^* \otimes m_k \oplus 2^h) \\ &= \pm a_k 2^h. \end{aligned}$$

Since the SGA was run for  $t_2$  iterations  $x^{t_2}$  can be any of the strings that satisfies  $x^{t_2} = x^* \oplus 2^h, 0 \leq h < \ell$ . This gives the freedom to choose the mask and its coefficient such that  $2^h \otimes m_i = 2^h$  and  $a_i 2^h$  is minimized.  $\square$

**Remark:** One special case of this class of functions is the royal road function of Mitchell and Forrest [8]. The royal road function can be shown to have a worst-case bound of  $|m_i|$  for all  $K \leq \ell + 1$  if all masks are the same size or  $\min_i |m_i|$  if they are

varying size for  $K \leq |m_i|$ . (See Wiles and Bradley [15] for further discussion on these functions.)

## 4 Conclusion

We presented a framework to analyze the solution quality of the SGA. Theorem 2 establishes how the ideas in Theorem 1 (i.e.,  $K$  distinguished strings) can be used to bound error with some probability. We showed that with some pre specified probability we can bound the distance between the current best and optimum. We also analyzed two commonly used functions using Corollary 1.

In essence the results presented here are worst-worst case bounds - worst in terms of the number of iterations and worst in terms of the structure of the  $K$  distinguished strings. Work is under way to determine what happens if we have information about the structure of the neighborhood of the optimal.

### References:

- [1] Aytug H., G.J. Koehler, The Effect of Multiple Optima on the Simple GA Run-time Complexity, *European Journal of Operational Research*, 2007, 178, 1, pp. 27-45..
- [2] Aytug, H. and G. J. Koehler, New Stopping Criteria for Genetic Algorithms, *European Journal of Operational Research*, 126, 2004, pp. 662-674.
- [3] Aytug, H. and G. J. Koehler, Stopping Criteria for Finite Length Genetic Algorithms. *ORSA Journal on Computing*, 8, No. 2, 1996, pp. 183-191.
- [4] Digalakis J.G., K.G. Margaritis, On Benchmarking Functions for Genetic Algorithms, *International Journal of Computer Mathematics*, 2000, pp. 1-27.
- [5] Giddings, A.P., R.L. Rardin, B.W. Schmeiser, R. Uzsoy, Statistical Optimum Estimation Techniques for Combinatorial Optimization Problems: A Review and Critique, *Working Paper. School of Industrial Engineering, Purdue University*, West Lafayette, IN 47907-1287, 2004.
- [6] Golden, B.L., F.B. Alt, Interval Estimation of a Global Optimum for Large Combinatorial Problems, *Naval Research Logistics*, 26, 1979, pp. 69-77.
- [7] Greenhalgh, D. and S. Marshall, Convergence Criteria for Genetic Algorithms, *SIAM Journal on Computing*, 30, 1, 2000, pp.269-282.

- [8] Mitchell M., S. Forrest. 1997. B.2.7.5: Fitness Landscapes: Royal Road Functions. *Handbook of Evolutionary Computation*. Oxford: Oxford University Press.
- [9] Nix, A. and M. D. Vose. 1992. Modeling Genetic Algorithms with Markov Chains. *Annals of Mathematics and Artificial Intelligence*. 5, pp. 79-88.
- [10] Nydick R.L., H.J. Weiss. 1994. An Analytical Evaluation of Optimal Solution Value Estimation Procedures. *Naval Research Logistics*. Vol. 41, pp. 189-202.
- [11] Ovacik I.M., S. Rajagopalan, R. Uzsoy. 2000. Integrating Interval Estimates of Global Optima and Local Search Methods for Combinatorial Optimization Problems. *Journal of Heuristics*. 6 481-500.
- [12] Vose M. D., and G. E. Liepins. 1991. Punctuated Equilibria in Genetic Search, *Complex Systems*, 5, no. 1, 31-44.
- [13] Vose, M.D. 1999. *The Simple Genetic Algorithm*. The MIT Press, Cambridge, MA.
- [14] Wegener I., C. Witt. 2000. On the behavior of the (1+1) Evolutionary Algorithm on Quadratic Pseudo-Boolean Functions. *Technical Report*. ISSN 1433-3325, University of Dortmund, Dept. of Computer Science, Dortmund, Germany.
- [15] Wiles J., T. Bradley. 2003. Mapping the Royal Road and Other Hierarchical Functions. *Evolutionary Computation*. Vol. 11, No.2, pp. 129-149.
- [16] Wright AH., Y. Zhao. 1999. Markov Chain Models of Genetic Algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference*. Orlando, Florida.