Pergamon

Computers Math. Applic. Vol. 27, No. 9/10, pp. 77–82, 1994
Copyright©1994 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0898-1221/94 $7.00 + 0.00

0898-1221(94)E0051-K

A Sequential Sampling Procedure for Genetic Algorithms

A. N. Aizawa
National Center for Science Information Systems
3-29-1 Otsuka, Bunkyo-ku, Tokyo 112, Japan
akiko@nacsis.ac.jp

B. W. Wah
Coordinated Science Laboratory, University of Illinois at Urbana-Champaign
1308 West Main Street, Urbana, IL 61801, U.S.A.
wah@manip.crhc.uiuc.edu

Abstract—In this paper, we apply sequential decision theory for scheduling tests in genetic algorithms and investigate an efficient sampling procedure for improving its performance. We use a loss function specifically defined for our analysis and obtain sequential decision equations for the optimal procedure. We derive simplified equations so that the procedure can be applied in practice. Finally, we compare the performance of our heuristic sampling procedure with that of the original genetic algorithms.

1. INTRODUCTION

A *genetic algorithm* is an adaptive search technique based on a selection and reproduction mechanism found in the natural evolution process [1–3]. It provides a robust yet efficient search procedure in the absence of domain knowledge for guiding the search. In the past decade, many applications of genetic algorithms have been developed in such areas as system control, parameter optimization, and pattern recognition [4].

Genetic algorithms are characterized by successive *generations* composed of k *individuals* (referred to as *candidates*). Each generation goes through the *selection* (or *testing) phase* and the *reproduction phase*. In the selection phase, candidates in the current generation are evaluated through experiments, and at the end, the normalized expected performance (or *fitness*) is estimated for each candidate. In the reproduction phase, a new generation is created by applying randomized genetic operators, such as crossover and mutation. These operators guide the reproduction process, or search, toward regions with better fitness values. As a result, an accurate method for estimating the fitness values is crucial in the reproduction phase.

Our objective, in this paper is to propose and show the usefulness of an *adaptive sampling* procedure in genetic algorithms using sequential decision theory [5,6]. Since the testing process is often prone to error in the actual environment, a substantial number of samples are required to accurately estimate the fitness value of a candidate. This is not possible when the total time allowed is restricted, and only a limited number of samples can be taken. Existing studies invariably use *fixed size sampling* methods, which take a constant number of samples for each candidate. Since it is likely that poor candidates can be determined with a small number of samples, it is not necessary to sample all candidates to the same extent. In this paper, we study more effective algorithms for allocating a limited number of tests among candidates.

Research supported by National Aeronautics and Space Administration, Contract Number NCC 2-481. Included in the *Proceedings of Fifth International Workshop of the Bellman Continuum*, Waikoloa, Hawaii, January 1993.

2. DEFINITIONS AND ASSUMPTIONS

Figure 1 illustrates the generation and testing phases of a genetic algorithm for a generation of k candidates $(c_1, \ldots, c_k)$. To allow the problem to be tractable, we assume in this paper that:

(a) the time allowed is divided into multiple generations of size T each,
(b) each candidate can be tested once in unit time,
(c) our goal is to find better algorithms for selecting the candidates to be sampled within a generation,
(d) samples from different candidates are independent, and
(e) samples from the same candidate (c_i) are i. i. d. (independent and identically distributed) with a *sample distribution* f_i, that is defined in terms of an unknown location parameter m_i and a measurement error g_i.

$$f_i(x \mid m_i) = m_i + g_i(e). \tag{1}$$

In equation (1), m_i represents the true performance of c_i, where $m_i \in (-\infty, \infty)$, and $g_i(e)$ is a known (or estimated in advance) error function (such as the Gaussian error). The error is independent of the value of m_i, decreases monotonically as the number of samples increases, and converges to zero asymptotically. The *parameter space* Θ is the space of all possible values of m_i's, where $(m_1, \ldots, m_k) \in \Theta$.

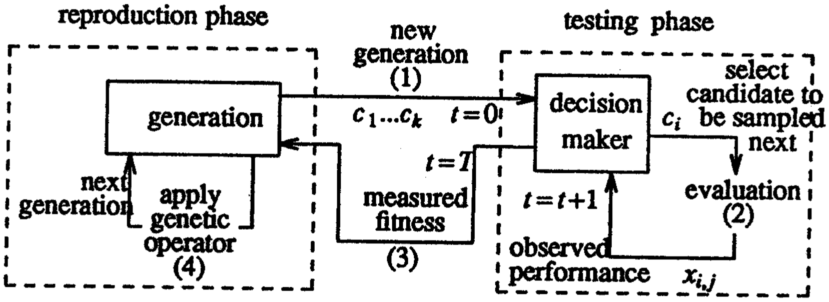

Figure 1. How genetic algorithms work with sequential sampling scheme.

Consider $\mathbf{X}_{t_0}$, the *sample sequence* observed in step $t = t_0$. The possible outcomes of $\mathbf{X}_t$ compose the *sample space* Ξ_t, where $\mathbf{X}_t \in \Xi_t$. Let $\mathbf{x}_{i,t_0}$ be the subsequence of samples in $\mathbf{X}_{t_0}$ associated with c_i. Denoting the *prior distribution* of m_i as $h_i(m)$, the *posterior distribution* $h_i^*(m)$ is, from Bayes theorem,

$$h_i^*(m \mid \mathbf{x}_{i,t_0}) = \frac{p(\mathbf{x}_{i,t_0} \mid m)\, h_i(m)}{\int_{-\infty}^{\infty} p(\mathbf{x}_{i,t_0} \mid m)\, h_i(m)\, dm}, \tag{2}$$

where $p(\mathbf{x}_{i,t_0} \mid m)$ represents the probability density of $\mathbf{x}_{i,t_0}$ being observed when $m_i = m$. Finally, we denote n_{i,t_0} as the number of samples drawn from c_i in step t_0. The allocation rule is then expressed as $\Psi = (\Psi_0, \ldots, \Psi_T)$, with the t^{th} element defined as $\Psi_t = (n_{1,t}, \ldots, n_{k,t})$, where $\sum_i n_{i,t} = t$, and $n_{i,t_1} \leq n_{i,t_2}$, if $t_1 < t_2$.

3. SEQUENTIAL DECISION EQUATIONS

In every step t, the scheduler selects Ψ_{t+1}, the best allocation of samples among candidates that minimizes the expected loss at the end of the current generation. Let $q(\Theta)$ be the largest mean such that $q(\Theta) = \max(m_1, \ldots, m_k)$, and $y(q(\Theta) \mid \mathbf{T})$ be the Bayes estimator of $q(\Theta)$ given observation $\mathbf{X}_T$. For an error function l (such as the squared-error function), the loss function due to sampling can be expressed as

$$L(\Theta, \mathbf{X}_T) = l\,(q(\Theta),\, y(q(\Theta) \mid \mathbf{X}_T)). \tag{3}$$

Equation (3) means that loss is expressed as the estimation error of the best candidate. Although the equation only assumes the best candidate, it actually expresses the loss of the entire generation, as it is necessary to identify the best candidate as soon as possible in order to minimize the loss. Substituting $q(\Theta)$ by $m_{[k]}$ and $y(q(\Theta) \mid \mathbf{X}_T)$ by $\hat{m}_{[k],T}$, we can express the loss simply as $L(\Theta, \mathbf{X}_T) = l(m_{[k]}, \hat{m}_{[k],T})$. ($m_{[k]}$ is the k^{th} order statistics of m.)

The equation for the expected risk is obtained as follows. Assume we have observation $\mathbf{X}_{t_0}$ in step t_0. Denote $P(\mathbf{X})$ as the probability that event $\mathbf{X}$ is obtained in the final step T. For a given parameter set Θ and an allocation rule Ψ, the expected loss (or risk) in step T is expressed as

$$\begin{aligned} R(\Theta, \Psi \mid \mathbf{X}_{t_0})_{t=T} &= \int_{\mathbf{X}_T \in \Xi_T} l(q(\Theta), y(q(\Theta \mid \mathbf{X}_T)) \, dP(\mathbf{X}_T \mid \Theta, \Psi, \mathbf{X}_{t_0}) \\ &= \int_{-\infty}^{\infty} l\left(m_{[k]}, \hat{m}_{[k],T}\right) dP\left(\hat{m}_{[k],T} \mid m_{[k]}, \Psi, \mathbf{x}_{[k],t_0}\right) \\ &\triangleq r\left(m_{[k]}, \Psi \mid \mathbf{x}_{[k],t_0}\right)_{t=T}. \end{aligned} \tag{4}$$

Note that if Θ is given, the risk in equation (4) depends only on samples obtained from the best candidate. Let H_i^* be the c.d.f. of the posterior distribution of m_i. Then $\prod_{j \neq i} H_j^*(m \mid \mathbf{x}_{j,t_0})$ $h_i^*(m \mid \mathbf{x}_{i,t_0})$ represents the probability density that m_i is the largest with value $m_i = m$ after observation $\mathbf{X}_{t_0}$ is obtained. In Bayesian analysis, the expected risk in step T of this generation can be expressed as

$$\hat{R}(\Psi \mid \mathbf{X}_{t_0})_{t=T} = \sum_{i=1}^{k} \int_{-\infty}^{\infty} r(m, \Psi \mid \mathbf{x}_{i,t_0})_{t=T} \prod_{j \neq i} H_j^*(m \mid \mathbf{x}_{j,t_0}) \, h_i^*(m \mid \mathbf{x}_{i,t_0}) \, dm. \tag{5}$$

Equation (5) can be solved optimally by applying backward inference of dynamic programming. Let $\Psi_T^* = \left(n_{1,T}^*, \dots, n_{k,T}^*\right)$ be the optimal allocation that minimizes equation (5) for a given $\mathbf{X}_t$. From Bellman's *Principle of Optimality* [7], the optimal decision (or sampling procedure) in step t is

$$n_{i,t+1} = n_{i,t} + 1, \qquad \text{for one of the } c_i\text{'s such that } n_{i,t} < n_{i,T}^* \tag{6}$$

In other words, we should chose to sample any candidate i if $n_{i,t}$, the number of samples obtained so far, is less than $n_{i,T}^*$, the number of samples prescribed by the optimal decision procedure. The solution is, therefore, straightforward if we can determine the optimal allocation $\left(n_{1,T}^*, \dots, n_{k,T}^*\right)$.

4. SIMPLIFIED EQUATIONS

Although the optimal allocation can be obtained easily for $k = 2$, its computational overhead is too high for it to be practical when $k \geq 3$. In the following, we derive simplified heuristic equations for providing a feasible sampling procedure for larger values of k. The effectiveness of the approximation is confirmed by comparing the simulation result with the optimal case for $k = 2$ (not shown due to space limitation).

When $t = t_0$, $P(\mathbf{X}_t \mid \Theta, \Psi, \mathbf{X}_{t_0})$ in equation (4) is equal to 1 for $\mathbf{X}_t = \mathbf{X}_{t_0}$, and is 0 otherwise; Equation (5) can then be simplified as

$$\begin{aligned} \hat{R}(\Psi \mid \mathbf{X}_{t_0})_{t=t_0} &= \sum_{i=1}^{k} \int_{-\infty}^{\infty} l(m, \hat{m}_{i,t_0}) \prod_{j \neq i} H_j^*(m \mid \mathbf{x}_{j,t_0}) \, h_i^*(m \mid \mathbf{x}_{i,t_0}) \, dm \\ &= \sum_{i} \int_{-\infty}^{\infty} \prod_{j \neq i} H_i^*(m \mid \mathbf{x}_{j,t_0}) \, h_i^*(m \mid \mathbf{x}_{i,t_0}) \, dm \end{aligned}$$

$$\times \frac{\int_{-\infty}^{\infty} l\,(m, \hat{m}_{i,t_0}) \prod_{j\neq i} H_i^*(m \mid \mathbf{x}_{j,t_0})\, h_i^*(m \mid \mathbf{x}_{i,t_0})\, dm}{\int_{-\infty}^{\infty} \prod_{j\neq i} H_i^*(m \mid \mathbf{x}_{j,t_0})\, h_i^*(m \mid \mathbf{x}_{i,t_0})\, dm}$$

$$= \sum_i P_{i,t_0}^*\, E\left[l\,(m_i, \hat{m}_{i,t_0}) \mid m_i \text{ is the largest }\right]. \tag{7}$$

In equation (7), P_{i,t_0}^* denotes the *posterior probability*, the probability that m_i is the largest in step t_0, where

$$P_{i,t_0}^* = \int_{-\infty}^{\infty} \prod_{j\neq i} H_j^*(m \mid \mathbf{x}_{j,t_0})\, h_i^*(m \mid \mathbf{x}_{i,t_0})\, dm.$$

Next, we apply the following heuristic simplifications to equation (7).

(i) $E\left[l\,(m_i, \hat{m}_i) \mid m_i \text{ is the largest }\right] \propto E\left[l\,(m_i, \hat{m}_i)\right]$ (for all i).
(ii) $\frac{\partial P_i^*}{\partial n_j} \sim 0$ (for all i, j).

Heuristics (i) assumes that the expected estimation error, on condition that m_i is the largest, is approximately proportional to the unconditional expected error of m_i. Heuristics (ii) assumes that the effect of changing the sample size is negligible on P_{i,t_0} as compared with the effect on $E\left[l\,(m_i, \hat{m}_i)\right]$. Although these assumptions are difficult to be justified experimentally, they are made in order for our sampling procedure to be usable without a significant overhead. Denoting $E\left[l\,(m_i, \hat{m}_{i,t_0})\right]$ as $\hat{L}_i(n_i)$ and P_{i,t_0}^* as P_i^*, we obtain the following equation with k variables $(n_1, \ldots, n_k)$,

$$\hat{R}(\Psi \mid \mathbf{X}_{t_0})_{t=t_0} \approx \sum_{i=1}^{k} P_i^*\, \hat{L}_i(n_i), \qquad \text{where } \sum_{i=1}^{k} n_i = t_0. \tag{8}$$

Equation (8) is interpreted as follows. The expected risk is a sum of the estimation error of m_i weighed by the posterior probability P_i^*; its value is high if a candidate with a high probability of being the best has a large estimation error, and its effect is negligible when it has the same estimation error and its probability is small.

Next, we will treat the allocation that minimizes equation (8) as the *desired allocation* $\left(n_{1,t_0}^+, \ldots, n_{k,t_0}^+, \sum_i n_{i,t_0}^+ = t_0\right)$, and use the value as a feedback to the scheduler. Applying a Lagrange multiplier to (8), it immediately follows that the following equations hold for the desired allocation

$$P_1^*\, \frac{\partial \hat{L}_1\,(n_{1,t_0}^+)}{\partial n_1} = P_2^*\, \frac{\partial \hat{L}_2\,(n_{2,t_0}^+)}{\partial n_2} = \cdots = P_k^*\, \frac{\partial \hat{L}_k\left(n_{k,t_0}^+\right)}{\partial n_k}. \tag{9}$$

In other words, the desired allocation equalizes each term of (9). Using the difference between the desired and the actual outputs as a feedback, our sampling policy in step t_0 is

$$n_{i,t+1} = n_{i,t} + 1, \tag{10}$$

for the c_i such that

$$P_i^*\, \frac{\partial \hat{L}_i\,(n_{i,t_0}^+)}{\partial n_i} - P_i^*\, \frac{\partial \hat{L}_i(n_{i,t_0})}{\partial n_i} = \max_{1\leq j\leq k} \left[P_j^*\, \frac{\partial \hat{L}_j\,(n_{j,t_0}^+)}{\partial n_j} - P_j^*\, \frac{\partial \hat{L}_j(n_{j,t_0})}{\partial n_j}\right].$$

We can omit the first term on both sides of (10) as equation (9) shows that they are equal. Note that $\frac{\partial \hat{L}_i(n_i)}{\partial n_i}$ is negative for all i.

5. EXAMPLES

In this section, we show the effect of applying our adaptive sampling procedure in a genetic search. Due to the generality of the normal distribution, we assume that the sample and the prior distributions are both normal.

The equations for the case with normal distributions is obtained as follows. Suppose candidate c_i has a sample distribution $N\left(\mu_i, {\sigma_i}^2\right)$ and a prior distribution $N\left(\mu_{0\,i},\, {\sigma_{0\,i}}^2\right)$, where only μ_i is unknown. Assume in the current step, we have $(n_1, \ldots, n_k)$ samples with sample means $\bar{x}_1, \ldots, \bar{x}_k$. From (2), the posterior distribution is also normal $N\left(\mu_i^*,\, \sigma_i^{*2}\right)$, with

$$\mu_i^* = \frac{n_i\,\bar{x}_i + \alpha_i\,\mu_{0\,i}}{n_i + \alpha_i} \quad \text{and} \quad \sigma_i^{*2} = \frac{{\sigma_i}^2}{n_i + \alpha_i},\ \alpha_i = \frac{{\sigma_i}^2}{{\sigma_{0\,i}}^2}.$$

When the squared error is used for l, the Bayes estimator for m_i is equal to m_i^*, and the expected loss is expressed as $\hat{L}_i(n_i) = \sigma_i^{*2}$. The posterior probability and the gradient of the loss can then be derived as

$$P_i^* = \int_{-\infty}^{\infty} \prod_{j \neq i} \Phi\left[\frac{m - \mu_j^*}{\sigma_j^*}\right] d\Phi\left[\frac{m - \mu_i^*}{\sigma_i^*}\right], \quad \frac{\partial \hat{L}_i(n_i)}{\partial n_i} = -\frac{{\sigma_i}^2}{(n_i + \alpha_i)^2}. \tag{11}$$

Applying (11) to equation (10), we can select the candidate to be sampled next.

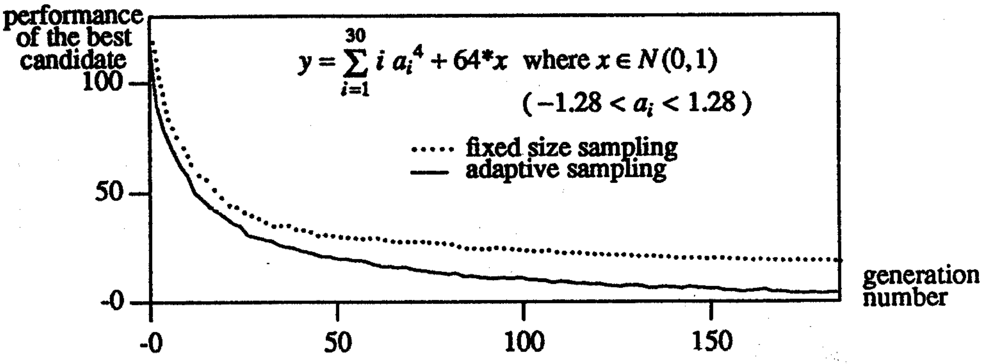

Figure 2. Effect of adaptive sampling.

As an example, we compare the performance of the conventional fixed-size sampling procedure and our adaptive sampling procedure using the test function proposed by DeJong [8] $y = \sum_{1 \le i \le 30} i\, {a_i}^4 + 64\,x$, where $x \in N(0,1)$ and $(-1.28 < a_i < 1.28)$. The problem is to find out the combination $(a_1, \ldots, a_{30})$ that minimizes the expected value of y using a genetic algorithm, assuming that there is no prior knowledge about the relationship between a_i and y. In our simulation study, we assume that the original genetic algorithm has a population size of 30, crossover rate of 0.6, mutation rate of 0.01, scaling window of 1, and overlapping generations [2]. We spent 90 units of the time allowed for presampling in order to estimate the prior distribution and the sample variance (assumed to be common for all candidates). These estimations are updated continuously as sampling proceeds. Figure 2 shows the performance improvement with respect to sampling time averaged over 100 runs, when there are 30 candidates per generation ($k = 30$) and 54 time units per generation ($T = 54$). It shows that adaptive sampling results in better candidates being selected, as compared to the fixed-sampling procedure in traditional genetic algorithms.

6. CONCLUSIONS

Genetic algorithms represent an important class of algorithms for searching a large space of possibilities. Existing sampling procedures, however, test a fixed number of sample for each candidate, and samples good as well as poor candidates to the same extent. These work well when the time allowed is very large. In this paper, we show that, under a fixed amount of time, better candidates can be selected by adaptive sampling based on sequential decision theory. Our results can also be extended to learning in classifier systems with noisy samples.

REFERENCES

1. L.B. Booker, D.E. Goldberg and J.H. Holland, Classifier systems and genetic algorithms, In *Machine Learning: Paradigm and Methods*, (Edited by J. Carbonell), pp. 235–282, MIT Press, (1990).
2. D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, (1989).
3. J.H. Holland, *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press, Ann Arbor, MI, (1975).
4. D.E Goldberg and K. Deb, A gentle introduction to genetic algorithms, Tutorial Notes in *Computer Vision and Pattern Recognition Conf.*, Champaign, IL, (June 1992).
5. T.S. Ferguson, *Mathematical Statistics: A Decision Theoretic Approach*, Academic Press, NY, (1967).
6. B.K. Ghosh and P.K. Sen, Editors, *Handbook of Sequential Analysis*, Marcel Dekker, Inc., NY, (1991).
7. A. Kaufmann and R. Cruon, *Dyanmic Programming*, Academic Press, New York, (1967).
8. K.A. DeJong, Analysis of the behavior of a class of genetic adaptive systems, *Doctorial Dissertation*, Department of Computer and Communication Sciences, Univ. of Michigan, MI, (1975).