

Branching Process Analysis of Linear Ranking and Binary Tournament Selection in Genetic Algorithms¹

Mandira Chakraborty and Uday K. Chakraborty

Department of Computer Science & Engineering, Jadavpur University, Calcutta, India

The present paper extends the branching process model [4] to cover linear ranking selection and binary tournament selection in genetic algorithms. Under certain simplifying assumptions, we derive the probability that a given class would vanish from the population by a given generation. No crossover or mutation is considered. Simulation results are presented.

Keywords: Genetic algorithm, branching process, linear ranking selection, binary tournament selection

1. Introduction

Genetic algorithms (GAs) [13, 16] are a class of computational models inspired by natural evolution. These algorithms are most commonly viewed as function optimizers (search heuristics). Application areas of these algorithms (and their many derivatives) include science, engineering, business and many other fields (see, for example, [6, 10, 11, 12, 21]).

Genetic algorithms start with a population of randomly (or heuristically) generated candidate points (trial solutions) in the search space. Each candidate point is coded (following some predetermined encoding scheme) to represent some underlying parameter set. The algorithm operates in a number of iterations, in an attempt to improve upon the trial solutions. In each iteration (an iteration is called a "generation" in the GA parlance), several probabilistic operators are applied to the trial solutions with a view to creating (possibly) better solutions.

The algorithm terminates when either an optimal / near-optimal solution has been found or a specified number of generations have been completed. It is to be noted that the genetic algorithm is a "weak" method, with no guarantee of finding the optimum solution in a particular run.

The outline of the simple genetic algorithm is as follows:

```

t = 0;
initialize population(t);
evaluate candidate points in population(t);
while predetermined termination condition not
satisfied
{
    t = t + 1;
    select population(t) from population(t-1);
    apply recombination and mutation to candi-
    date points in population(t);
    evaluate candidate points in population(t);
}

```

For some recent advances in theoretical research on genetic algorithms, see [5, 7, 8, 9, 18, 19, 20, 22, 23].

Selection is a major component of the genetic algorithm. In all variants of the GA some form of the selection operator must be present. The basic principle of selection is to allocate, in the next generation, more copies to the fit individuals and fewer copies to the poor ones. Selection pressure directly controls the exploitation factor in the "exploitation-versus-exploration"

¹ An earlier version of this paper appeared in the Proceedings of the IEEE International Conference on Information, Communications and Signal Processing, Singapore, September 1997.

tradeoff that is believed to be critically important in the working of the GA. A wide variety of selection algorithms have been proposed in the GA-literature. For an analysis of the effects of various selection schemes, see [8].

Despite the existence of many experimental studies and approximate, deterministic models of linear ranking and binary tournament selection, there is a dearth of stochastic models of these two selection algorithms. The present paper seeks to fill this void. This paper extends the branching process model [4] to linear ranking selection and binary tournament selection.

2. The Branching Process

The branching process starts at time zero with an initial set of individuals which constitute the 0th generation. Each of these individuals, after one unit of time, gives rise to a random number of offspring according to a pre-determined probability distribution. The children of the 0th generation constitute the first generation; their children are the second generation, and so on.

We consider those individuals in the population whose fitness values are worse than or equal to a predetermined value. To investigate the effect of the above-mentioned two types of selection algorithms, we study the growth/decay of individuals of this designated class in the population.

Let $X^{(t)}$ be the random variable representing the number of individuals in the designated class at generation t . For the genetic algorithm, the successive random variables $X^{(0)}, X^{(1)}, X^{(2)}, \dots$ form a Markov chain. Let the non-negative integer-valued discrete random variable Y represent the number of members of the designated class that a single member (of this class) produces in the next generation, and let $\text{Prob}(Y = k) = p_k, k = 0, 1, \dots$ and $\sum_k p_k = 1$. Then the probability generating function of Y is given by

$$G(z) = \sum_{k=0}^{\infty} p_k z^k. \quad (1)$$

We now compute p_k for the worst individual in the population, and use that value as an approximation to the p_k of any individual of the designated class.

3. Linear Ranking

In ranking selection [1, 15], at each generation, the individuals in the population are sorted according to their fitness and each individual is assigned a rank in the sorted population. The worst individual gets the rank 1 while the best receives the rank N ($N =$ population size). The selection probabilities of the individuals x_k ($k = 1, \dots, N$) are given by some function (most commonly, linear) of their rank.

Let $\{x_1^{(t)}, x_2^{(t)}, \dots, x_N^{(t)}\}$ denote the population at generation t . Then in linear ranking selection the probability of selecting individual x_k ($k = 1, 2, \dots, N$) is given by

$$p(x_k^{(t)}) = \frac{1}{N} \left(\min + \frac{(\max - \min)(\text{rank}(x_k^{(t)}) - 1)}{N - 1} \right)$$

where $\max + \min = 2$ and $1 \leq \max \leq 2$. The $\{p(x_k^{(t)})\}$ is a proper probability distribution ($\sum_{k=1}^N p(x_k^{(t)}) = 1$ for each t), and sampling N individuals according to this probability distribution yields the next generation.

The selection probability, p_{sel} , of the worst individual (rank = 1) is then given by

$$p_{sel} = \frac{\min}{N}.$$

We then have

$$p_k = \binom{N}{k} (p_{sel})^k (1 - p_{sel})^{N-k}. \quad (2)$$

When N is large and p_{sel} is small and the product Np_{sel} is of moderate magnitude, we can apply the Poisson approximation:

$$\begin{aligned} p_k &= \frac{e^{-Np_{sel}} (Np_{sel})^k}{k!} \\ &= \frac{e^{-\min} \min^k}{k!}. \end{aligned}$$

From eq. 1 we now have

$$\begin{aligned} G(z) &= e^{-\min} \sum_{k=0}^{\infty} \frac{(\min \cdot z)^k}{k!} \\ &= e^{-\min(1-z)}. \end{aligned}$$

Let the iterates of the probability generating function G be defined by:

$$\begin{aligned} G_0(z) &= z, \\ G_1(z) &= G(z), \\ G_{t+1}(z) &= G(G_t(z)), t = 1, 2, \dots \end{aligned}$$

Let $G^{(t)}(z)$ denote the probability generating function of $X^{(t)}$, $t = 0, 1, \dots$. Letting m represent the number of members of the designated class at the initial generation, we have the following theorem:

Theorem 1:

$$G^{(t)}(z) = [G_t(z)]^m, t = 0, 1, \dots \quad (3)$$

Proof: See [4].

Let $p_{loss}(t)$ denote the probability that the designated class vanishes from the population by generation t .

Theorem 2: We have

$$p_{loss}(t) = e^{-m \cdot \min.(1 - p_{loss}^{1/m}(t-1))}$$

Proof:

$$\begin{aligned} p_{loss}(t) &= [G(G_{t-1}(0))]^m \\ &= [\exp(-\min(1 - G_{t-1}(0)))]^m \\ &= \exp(-m \cdot \min.(1 - G_{t-1}(0))) \\ &= \exp(-m \cdot \min(1 - \{G^{(t-1)}(0)\}^{1/m})) \\ &= \exp(-m \cdot \min(1 - \{p_{loss}(t-1)\}^{1/m})) \end{aligned}$$

4. Binary Tournament

Although there are a number of variants, the basic mechanism of tournament selection [3, 14] consists of

- Randomly choosing — with or without replacement — a predetermined number of individuals from the population, and picking — probabilistically or deterministically — the best from these individuals.
- Repeating the above step N times (N = population size) to fill the next generation.

As in the case of ranking selection, here, too, we compute p_k for the worst individual in the population. Under binary tournament with replacement, in each tournament any one of the following four mutually exclusive and exhaustive events will occur:

1. The worst individual will be picked twice (note that the picking is considered to have been performed with replacement).
2. Neither of the two individuals picked is the worst individual.
3. The first individual picked is the worst individual, but the second is not.
4. The second individual picked is the worst individual, but the first is not.

It is easy to see that

$$\begin{aligned} \text{Prob (event 1)} &= \left(\frac{1}{N}\right) \cdot \left(\frac{1}{N}\right), \\ \text{Prob (event 2)} &= \left(1 - \frac{1}{N}\right) \cdot \left(1 - \frac{1}{N}\right), \\ \text{Prob (event 3)} &= \left(\frac{1}{N}\right) \cdot \left(1 - \frac{1}{N}\right), \\ \text{Prob (event 4)} &= \left(1 - \frac{1}{N}\right) \cdot \left(\frac{1}{N}\right). \end{aligned}$$

If p denotes the probability with which the better of the two individuals is chosen in any single tournament, then the probability that in any single tournament the worst individual is picked is given by

$$p_{sel} = \left(\frac{1}{N}\right)^2 + 2\frac{1}{N} \left(1 - \frac{1}{N}\right) (1 - p), \quad (4)$$

where $0.5 < p \leq 1$.

Proceeding as in the case of linear ranking selection, we can show that for binary tournament selection we have the following theorem:

Theorem 3:

$$p_{loss}(t) = e^{-m \cdot N \cdot p_{sel} \cdot (1 - p_{loss}^{1/m}(t-1))}$$

where p_{sel} is given by eq. 4.

5. Expected Loss Time

Defining

$$f_{loss}(t) = p_{loss}(t) - p_{loss}(t - 1)$$

for $t = 1, 2, \dots$, the expected (mean) loss time is given by

$$\sum_{t=0}^{\infty} f_{loss}(t) \cdot t$$

6. Empirical Results

To obtain empirical support for the model, experiments were conducted by running the genetic algorithm on a simulation problem with only two types of individuals. Each GA run was continued until the better individual filled the entire population. One hundred independent runs — with as many different initial seeds for the pseudo-random number generator — were taken. All of these 100 runs were started with the same initial population. For each run, the generation number at which the worse individual vanished from the population was noted. The loss probability was obtained from the relation

$$p_{loss}(t) = \frac{n(t)}{100},$$

where $n(t)$ = total number of runs (out of 100) in which the worse class vanished at or before generation t .

The theoretical and empirical f_{loss} values are compared in Tables 1 and 2 where the standard deviation and the 95% confidence interval have also been presented (number of samples = 100; sample mean = relative frequency). The empirical probabilities are not statistically significantly different from the corresponding true values at the 5% level of significance (i.e., with 95% confidence). Confidence intervals marked "-" in Tables 1 and 2 correspond to close-to-zero values of f_{loss} for which the product of the number of samples and the probability is less than 5.

Gen.	Theor. Mean	Sample Mean	Sample Std. Dev.	95% Confid. Interval
1	0.007	0		
2	0.132	0.13	0.336	0.064, 0.196
3	0.269	0.31	0.462	0.219, 0.401
4	0.244	0.17	0.376	0.096, 0.244
5	0.159	0.16	0.367	0.088, 0.232
6	0.090	0.10	0.300	0.041, 0.159
7	0.049	0.06	0.237	0.013, 0.107
8	0.025	0.04	0.196	-
9	0.012	0		-
10	0.007	0		-
11	0.003	0.01	0.099	-
12	0.001	0.02	0.140	-
13	0.001	0		-
14	0.001	0		-
15	0.000	0		-
.	.	.		
.	.	.		

Table 1. Theoretical and experimental f_{loss} values in linear ranking selection ($N = 100, max = 1.5, m = 10$).

Gen.	Theor. Mean	Sample Mean	Sample Std. Dev.	95% Confid. Interval
1	0.017	0.02	0.140	-
2	0.241	0.22	0.414	0.139, 0.301
3	0.340	0.40	0.490	0.304, 0.496
4	0.218	0.20	0.400	0.122, 0.278
5	0.106	0.12	0.325	0.056, 0.184
6	0.046	0.02	0.140	-
7	0.019	0.02	0.140	-
8	0.008	0		-
9	0.003	0		-
10	0.001	0		-
11	0.001	0		-
12	0.000	0		-
13	0.000	0		-
.	.	.		
.	.	.		

Table 2. Theoretical and experimental f_{loss} values in binary tournament selection ($N = 100, p = 0.8, m = 10$).

In these cases the normal distribution approximation (to the binomial distribution) underlying the confidence interval calculation process is not applicable [2, 17].

Some representative cases showing theoretical and experimental mean loss times are shown in

Tables 3 and 4. The results are averages of 100 runs.

Parameter Values	Theor. Mean Loss Time	Empir. Mean	Emp. Std. Dev.	95% Conf. Int.
$N = 100$ $m = 10$ $max = 1.5$	4.18	4.45	1.74	4.11, 4.79
$N = 300$ $m = 30$ $max = 1.5$	5.68	5.97	1.53	5.67, 6.27
$N = 300$ $m = 90$ $max = 1.1$	32.72	34.65	11.01	32.49, 36.81

Table 3. Linear ranking selection: comparison of theoretical and empirical mean loss times.

Parameter Values	Theor. Mean Loss Time	Empir. Mean	Emp. Std. Dev.	95% Conf. Int.
$N = 100$ $m = 10$ $p = 0.8$	3.44	3.51	1.28	3.26, 3.76
$N = 300$ $m = 30$ $p = 0.8$	4.57	4.73	1.28	4.48, 4.98
$N = 300$ $m = 90$ $p = 0.6$	18.35	19.70	4.61	18.80, 20.60

Table 4. Binary tournament selection: comparison of theoretical and empirical mean loss times.

7. Comparison

The present analysis allows us to compare and contrast the two selection algorithms. Theorems 2 and 3 describe the selection algorithms' behavior. Tables 5 and 6 show how the selection pressure changes with changes in the algorithm parameter's value. The data show that for $max = 2p$ the two loss times are almost identical. This corroborates a similar observation in [8, p. 161]. In the case of no selection pressure ($p = 0.5$ and $min = max = 1$), p_{sel} becomes $1/N$ for both the algorithms, and the equations in Theorems 2 and 3 become exactly identical. That the loss time corresponding to

the maximum selection pressure in tournament selection (the last row in Table 6) is greater than 1 can be seen from equation 4 where, for $p = 1$, p_{sel} becomes $1/N^2$, not zero. Thus the selection pressure is marginally lower in tournament than in ranking.

min	Mean Loss Time
0.8	9.26
0.6	5.22
0.4	3.40
0.2	2.26
0	1

Table 5. Change of the (theoretical) mean loss time with min in ranking selection ($N = 100, m = 10$).

p	Mean Loss Time
0.6	9.33
0.7	5.27
0.8	3.44
0.9	2.30
1.0	1.09

Table 6. Change of the (theoretical) mean loss time with p in tournament selection ($N = 100, m = 10$).

8. Conclusion

By building on earlier work, this paper has provided a theoretical analysis of two of the most important selection strategies in genetic algorithms. The analysis enables us to compute the probability that, under selection alone, individuals of a given class (i.e., individuals with fitness worse than or equal to a certain value) would be lost from the population by a given generation. We have used the analysis to compare the selection pressures of the two schemes. The results of this paper sharpen our insight into the working of ranking and tournament selection. It should be noted that the present model is an approximate one. This model assumes that $p_{loss}(\infty) = 1$, but strictly speaking, this is less than unity. In other words, for an exact model (e.g., Chakraborty et al. [8]), $\sum_{t=0}^{\infty} f_{loss}(t) < 1$.

Acknowledgments

We are grateful to two anonymous referees for their detailed comments. Thanks also to Dr. R. Manger for his comments and criticism. This work was partially supported by grants from the Jadavpur University Research Fund. The second author is supported by an AICTE Career Award (1996).

References

- [1] BAKER, J.E. (1985): Adaptive selection methods for genetic algorithms, In J.J. Grefenstette (Ed.), *Proc. Internat. Conf. on Genetic Algorithms* (pp. 101-111), Hillsdale, NJ: Lawrence Erlbaum.
- [2] BHATTACHARYYA, G.K. AND JOHNSON, R.A. (1977): *Statistical Concepts and Methods*, New York: John Wiley & Sons.
- [3] BRINDLE, A. (1981): *Genetic algorithms for function optimization*. Doctoral dissertation, University of Alberta, Edmonton.
- [4] CHAKRABORTY, U.K. (1995): A branching process model for genetic algorithms, *Information Processing Letters* **56**(5), 281-292.
- [5] CHAKRABORTY, U.K. (1995): A simpler derivation of schema hazard in genetic algorithms, *Information Processing Letters* **56**(2), 77-78.
- [6] CHAKRABORTY, M. AND CHAKRABORTY, U.K. (1997): Applying genetic algorithm and simulated annealing to a combinatorial optimization problem. *Proceedings of the IEEE International Conference on Information, Communications & Signal Processing*, Singapore, September 1997, 929-933.
- [7] CHAKRABORTY, U.K. AND MÜHLENBEIN, H. (1997): Linkage equilibrium and genetic algorithms, *Proceedings of the 4th IEEE International Conference on Evolutionary Computation*, Indianapolis, USA, April 1997, 25-29.
- [8] CHAKRABORTY, U.K., DEB, K. AND CHAKRABORTY, M. (1996): An analysis of selection algorithms: A Markov chain approach, *Evolutionary Computation* **4**(2), 133-167.
- [9] CHAKRABORTY, U.K. AND DASTIDAR, D.G. (1993): Using reliability analysis to estimate the number of generations to convergence in genetic algorithms, *Information Processing Letters* **46**(4), 199-209.
- [10] DAVIS, L. (ed.) (1991): *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.
- [11] ESHELMAN, L.J. (ed.) (1995): *Proc. 6th International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo.
- [12] FORREST, S. (ed.) (1993): *Proc. 5th International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo.
- [13] GOLDBERG, D.E. (1989): *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.
- [14] GOLDBERG, D.E., KORB, B. AND DEB, K. (1989): Messy genetic algorithms: motivation, analysis, and first results, *Complex Systems* **3**, 493-530.
- [15] GREFENSTETTE, J.J. AND BAKER, J.E. (1989): How genetic algorithms work: a critical look at implicit parallelism, In J.D. Schaffer (Ed.), *Proc. 3rd Internat. Conf. on Genetic Algorithms* (pp. 20-27), San Mateo, CA: Morgan Kaufmann.
- [16] HOLLAND, J.H. (1975): *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.
- [17] MARASCUILO, L.A. AND MCSWEENEY, M. (1977): *Nonparametric and Distribution-Free Methods for the Social Sciences*, Monterey, CA: Brooks/Cole Publishing Company.
- [18] MÜHLENBEIN, H. AND CHAKRABORTY, U.K. (1997): Analysis of the selection-recombination genetic algorithm for the bit-counting function, *Proceedings of the 2nd International Conference on Soft Computing (SOCO-97)*, Nimes (France), September, 1997.
- [19] MÜHLENBEIN, H. AND SCHLIERKAMP-VOOSEN, D. (1994): The science of breeding and its application to the breeder genetic algorithm (BGA), *Evolutionary Computation* **1**(4), 335-360.
- [20] MÜHLENBEIN, H. AND SCHLIERKAMP-VOOSEN, D. (1993): Predictive models for the breeder genetic algorithm, *Evolutionary Computation* **1**(1), 25-49.
- [21] SAHA, D. AND CHAKRABORTY, U.K. (1997): An efficient link enhancement strategy for computer networks using genetic algorithm, *Computer Communications* **20**(9), 798-803.
- [22] WHITLEY, L.D. (ed.) (1993): *Foundations of Genetic Algorithms - 2*, Morgan Kaufmann, San Mateo, CA.
- [23] WHITLEY, L.D. AND VOSE, M.D. (eds.) (1995): *Foundations of Genetic Algorithms - 3*, Morgan Kaufmann, San Mateo, CA.

Received: December, 1997

Revised: November, 1998

Accepted: December, 1998

Contact address:

Uday Chakraborty
 Department of Computer Science & Engineering
 Jadavpur University
 Calcutta 700032
 India
 phone: +91 33 413 0382
 fax: +91 33 473 4266
 e-mail: uday@jadav.ernet.in

UDAY KUMAR CHAKRABORTY is currently a research scientist at GMD (German National Research Center for Information Technology), Sankt Augustin, Germany. Earlier he was a reader in computer science & engineering at Jadavpur University, India. A PhD in computer science, he has published 30 research articles and a book. He is on the editorial board of the Journal of Computing and Information Technology.

MANDIRA CHAKRABORTY has been teaching at Jadavpur University, India since 1991. She has submitted her PhD thesis on genetic algorithms in 1998. She worked as an associate systems engineer at CMC Limited, Calcutta during 1988-90.
