A Theoretical Investigation of a Parallel Genetic Algorithm

Chrisila C. Pettey[†] and Michael R. Leuze[‡] *
[†]Department of Computer Science
Box 1679 Station B
Vanderbilt University
Nashville, Tennessee 37235
[‡]Oak Ridge National Laboratory
Mathematical Sciences Section
P.O. Box 2009, Bldg. 9207-A
Oak Ridge, Tennessee 37831-8083

Abstract

In the past few years the limitations of uniprocessor computing systems and the increasing availability of multiprocessors have led to investigations of parallel genetic algorithms. One algorithm in particular, called PGA, consists of a set of communicating sequential genetic algorithms. This paper includes an investigation of the theoretical allocation of trials to schemata by PGA's in general and by a particular PGA (which performs communication in a uniformly random manner) along with an experimental validation of an assumption made in the theoretical investigation.

1 Introduction

In the past few years the limitations of uniprocessor computing systems and the increasing availability of multiprocessors have led to investigations of parallel versions of genetic algorithms (GA's) [Cohoon 87] [Dolan 87] [Grefenstette 81] [Grefenstette 86] [Hilliard 88] [Jog 87] [Oosthuizen 87] [Pettey 87a] [Pettey 87b] [Pettey 88] [Robertson 87] [Sannier 87] [Tanese 87]. One approach (called a PGA [Pettey 87b]) maintains a number of communicating populations and has been the subject of several preliminary investigations [Pettey 87b] [Pettey 88] [Cohoon 87] [Tanese 87]. These investigations have demonstrated that PGA's perform quite well. PGA's have a further advantage in that they are easily implemented on any multiprocessor system since no constraints are placed on the topology of the multiprocessor interconnection network.

Despite good empirical behavior, PGA's have not been analyzed theoretically. The purpose of this paper is to establish a theoretical basis for the behavior of PGA's. In particular, this paper will address the question of how a PGA allocates trials to schemata, an important consideration in the theoretical work of Holland [Holland 75] and DeJong [DeJong 75] for standard sequential GA's.

Section II contains an outline of a PGA algorithm followed by a theoretical description of the algorithm in section III. The paper continues in section IV with a theoretical discussion of one particular PGA and concludes in section V with an experimental validation of assumptions that were made for the PGA of section IV.

2 Outline of the Algorithm

A PGA consists of a set of nodal GA's, each of which is resident on one node of a multiprocessor, maintains a part of the population, and executes the same algorithm as all other nodal GA's in the multiprocessor. The basic nodal GA outline is as follows:

*work supported in part by the Applied Mathematical Sciences Research Program of the office of Energy Research, U.S. Department of Energy

```
nodal GA:
    begin
      initialization;
      evaluation;
      while not done
          begin
            communication and selection;
            recombination;
            evaluation;
          end;
    end.
```

Except for the communication phase, which can occur before or after selection, a nodal GA is identical to a sequential GA. During the communication phase, a nodal GA exchanges information with other nodal GA's. A nodal GA can communicate with any arbitrary set of nodal GA's, including all of the nodal GA's in the multiprocessor. The volume of information exchanged can be small (e.g., the average evaluation of the nodal GA's population, or a single individual) or large (e.g., half of the individuals in the nodal GA's population). Throughout the remainder of this paper, we assume that nodal GA's communicate individuals.

When selection is performed by choosing individuals based on their performance relative to the average performance in the population (an assumption which will be maintained throughout this paper), then the selection phase of a GA has been shown to be efficient [DeJong 75] in that it allocates exponentially increasing (decreasing) trials to schemata based on their performance. Furthermore, for low order, short defining length schemata the recombination phase does not affect the efficiency of the selection phase. A PGA does not manipulate the entire population in the same manner as a sequential GA. Thus, even though empirical results (and a resemblance to the concept of a polytypic species of population genetics [Dobzhansky 70]) argue for the efficiency of PGA's, it remains to be demonstrated theoretically that PGA's maintain the efficiency of standard sequential genetic algorithms. The remainder of this paper is devoted to investigating the efficiency of PGA's.

3 Theoretical Description of a PGA

As DeJong [DeJong 75] demonstrated for sequential GA's, if it can be shown that a PGA allocates exponentially increasing (decreasing) trials to individuals based on their performance, then PGA's can be considered efficient algorithms. An investigation of the efficiency of a PGA must consider the effects of mutation, crossover, and selection on the entire population. Therefore, this section will begin by looking at the effects of mutation and crossover, and will conclude with a consideration of selection.

The first consideration is the effect of mutation on a PGA's population. Given that the mutation rate, p_m, (the probability that an allele is changed) is the same for each nodal GA, then the probability that a schema S of order $n(S)$ in the overall population undergoes mutation is $p_m n(S)$. Hence, as in a standard sequential GA, the probability that a schema survives is $1 - p_m n(S)$. Thus, converting a GA to a PGA does not change the effect of the mutation operator.

The second consideration is the effect of crossover on a PGA's population. Intuitively, one would expect that the effect of the crossover operator changes when a GA is converted to a PGA. Consider a randomly chosen individual in the population of a GA. There is a probability, p_c, that this individual will undergo crossover. The characteristics of this individual's children depend in part on which individual is chosen to be the other parent. In a population of size P, there is a $\frac{1}{P-1}$ chance of any other individual in the population being chosen to be the other parent. In general, if individuals are chosen without replacement, then, for the n^{th} individual chosen for crossover, there is a probability of

$$(1 - \frac{1}{P})(1 - \frac{1}{P-1}) \ldots (1 - \frac{1}{P-n+1})(\frac{1}{P-n})$$

that any other individual in the population will be chosen to be the other parent. For large n, this probability will be small, but nonzero. On the other hand, in a PGA, the probability that an individual from a given subpopulation will be crossed with an individual from another subpopulation is zero. In a population of size P that is distributed evenly among N nodes, there are $P - \frac{P}{N}$ individuals with zero probability of crossing with a randomly chosen individual.

While this constraint on the pairing of individuals may affect the outcome of the search, it does not affect a PGA's efficiency. The efficiency of a GA is completely determined by the survival rate of schemata rather than by the actual outcomes of recombination. Therefore, it is important to determine what effect the choice of an individual for crossover will have on the survival of the schemata which the individual represents. We conservatively assume that a schema will be destroyed if crossover occurs within the defining length of the schema. Given that the probability of choosing an individual in a subpopulation for crossover is p_c, then the probability of choosing an individual in the entire population for crossover is also p_c. Thus, for both GA's

and PGA's the probability that a schema S of defining length $d(S)$ will survive a crossover is $1-p_c\frac{d(S)}{L-1}$ (where L is the length of an individual). Therefore, during the recombination phase, a GA transformed into a PGA loses none of its efficiency.

On the other hand, the survival rate of schemata in a PGA differs from the survival rate of schemata in a GA due to the distributed nature of selection and to the communication of information. An average or poor individual in one subpopulation might be a super individual in another subpopulation. Furthermore, individuals could be replaced and/or distributed to other subpopulations by communication.

In this paper, the communication phase is considered in conjunction with the selection phase, and it is assumed that communication always precedes selection. It is also assumed that communication occurs every generation. Given these assumptions, there are several things to consider when analyzing the selection phase of a PGA. The first consideration is the probability of replacing an individual during the communication phase. This probability could be a function of the particular subpopulation of which an individual is a member. For individual i in subpopulation k, the probability of being replaced during the communication phase is r_{ik}; the probability of surviving the communication phase is $1-r_{ik}$. Therefore, the probability that an individual i in subpopulation k will be chosen to appear in the next generation during any one selection in subpopulation k is $(1-r_{ik})\frac{e_{ik}}{\sum_{j=1}^{P_k} e_{jk}}$, where e_{mk} is the performance of individual m at node k, and $P_k = \frac{P}{N}$. This implies that, given P_k selections, the expected number of offspring (copies) of individual i in subpopulation k in the next generation is $(1-r_{ik})\frac{e_{ik}}{e_{k\prime}}$, where $e_{k\prime}$ is the average evaluation of the subpopulation.

Another consideration is the probability of an individual appearing in another subpopulation after a communication and selection. Assume that s_{ik} is the probability of choosing individual i in subpopulation k for communication; assume that n_{ikl} is the probability that individual i from subpopulation k will be sent to subpopulation l if chosen for communication; and assume that a_{ikl} is the probability that individual i from subpopulation k will be accepted if sent to subpopulation l. Then $s_{ik}n_{ikl}a_{ikl}$ is the probability that individual i from subpopulation k will appear in subpopulation l after communication. Thus, the expected number of offspring of individual i from subpopulation k in subpopulation l is $s_{ik}n_{ikl}a_{ikl}\frac{e_{ik}}{e_{l\prime}}$.

The previous considerations imply that the expected number of offspring of a schema S in the overall population at time t is

$$m(S,t+1) = \sum_{k=1}^{N}\sum_{i=1,i\in S}^{P_k}[(1-r_{ik})\frac{e_{ik}}{e_{k\prime}} + \sum_{l\in C_k} s_{ik}n_{ikl}a_{ikl}\frac{e_{ik}}{e_{l\prime}}] \ ,$$

where C_k is the set of nodes with which node k communicates. It is assumed that each node has the same number of nodes in its communication set, that is, $\mid C_k \mid = C$ for all k.

In the limiting case of one node

$$m(S,t+1) = \sum_{i=1,i\in S}^{P}\frac{e_i}{e\prime} = \frac{m(S,t)e_{S\prime}}{e\prime} \ ,$$

where $e_{S\prime}$ is the average performance of S, which is the expected number of offspring of a schema in a sequential GA.

For more than one node

$$0 \le m(S,t+1) \le \sum_{k=1}^{N}\sum_{i=1,i\in S}^{P_k}(\frac{e_{ik}}{e_{k\prime}} + \sum_{l\in C_k}\frac{e_{ik}}{e_{l\prime}}) \ .$$

If it can be assumed that $e_{k\prime} \approx e\prime$ for all k, then

$$0 \le m(S,t+1) \le \sum_{k=1}^{N}\sum_{i=1,i\in S}^{P_k}(\frac{e_{ik}}{e\prime} + C\frac{e_{ik}}{e\prime})$$

$$= \sum_{k=1}^{N}\sum_{i=1,i\in S}^{P_k}(1+C)\frac{e_{ik}}{e\prime}$$

$$= m(S,t)(1+C)\frac{e_{S\prime}}{e\prime} \ .$$

Without making further assumptions about r_{ik}, s_{ik}, n_{ikl}, and a_{ikl}, it appears to be impossible to do more than bound the expected number of offspring of a schema by an exponentially increasing (decreasing) function. In the next section, a set of assumptions about these four probabilities is investigated with the aim of establishing a tighter bound for $m(S,t)$, the expected number of offspring of a schema.

4 An Instantiation of a PGA

Perhaps the simplest example of communication is to communicate only one individual from each subpopulation to all neighboring subpopulations. This implies that $n_{ikl} = 1$. Given this assumption, an obvious special case of a PGA is to perform all replacements due to communication and all selections of individuals to be sent in a uniformly random manner. Thus, the probability that individual i is sent, s_{ik}, is $\frac{1}{P}$.

Of the many ways of handling the communication, one is to accept all incoming individuals by placing

them in a buffer. The individuals are then randomly chosen to be placed in the population. Thus, incoming individuals can be overwritten by other incoming individuals. The probability that individual i will be replaced during one replacement is $\frac{1}{P}$. Therefore, r_{ik} is $1-(1-\frac{1}{P})^C$. The probability, a_{ikl}, that individual i is accepted at node l given that i has been sent to node l is:

$$\frac{1}{C}(1-\frac{1}{P})^{C-1}+(1-\frac{1}{C})\frac{1}{C-1}(1-\frac{1}{P})^{C-2}+\ldots$$

$$+(1-\frac{1}{C})(1-\frac{1}{C-1})(1-\frac{1}{C-2})\ldots(1-\frac{1}{C-(C-2)})(1-\frac{1}{P})^{C-(C-0)}$$

$$=\sum_{m=1}^{C}\frac{1}{C}(1-\frac{1}{P})^{C-m}$$

$$=\frac{\frac{1}{C}-\frac{1}{C}(1-\frac{1}{P})^C}{1-(1-\frac{1}{P})}$$

$$=\frac{P(1-(1-\frac{1}{P})^C)}{C} \quad .$$

Given the above probabilities, it is now possible to calculate the expected number of offspring of a schema S in the overall population.

$$m(S,t+1)=\sum_{k=1}^{N}\sum_{i=1,i\in S}^{P}[(1-\frac{1}{P})^C\frac{e_{ik}}{e_k\prime}+\sum_{l\in C_k}\frac{(1-(1-\frac{1}{P})^C)}{C}\frac{e_{ik}}{e_l\prime}] \quad .$$

If $e_k\prime \approx e\prime$ for all k, then

$$m(S,t+1)=\sum_{k=1}^{N}\sum_{i=1,i\in S}^{P}[(1-\frac{1}{P})^C\frac{e_{ik}}{e\prime}+\frac{(1-(1-\frac{1}{P})^C)e_{ik}}{e\prime}]$$

$$=m(S,t)\frac{e_S\prime}{e\prime}$$

Another possibility for handling communication is to expand the population to hold all incoming individuals. Then, when all C incoming individuals are present, randomly select C individuals out of the population of $P+C$ individuals to be thrown out. In this case, the probability, r_{ik}, that individual i is replaced during communication is $1-$ (*the probability that it is never removed*) which is

$$1-[(1-\frac{1}{P+C})(1-\frac{1}{P+C-1})\ldots(1-\frac{1}{P+C-(C-1)})]$$

$$=1-\frac{P}{P+C}=\frac{C}{P+C}$$

Similarly, the probability that individual i is accepted at node l, a_{ikl}, is

$$\frac{P}{P+C}$$

Thus, the expected number of offspring of a schema S is

$$m(S,t+1)=\sum_{k=1}^{N}\sum_{i=1,i\in S}^{P}[(1-(1-\frac{P}{P+C}))\frac{e_{ik}}{e_k\prime}+\sum_{l\in C_k}\frac{1}{P}\frac{P}{P+C}\frac{e_{ik}}{e_l\prime}]$$

$$=\sum_{k=1}^{N}\sum_{i=1,i\in S}^{P}[\frac{P}{P+C}\frac{e_{ik}}{e_k\prime}+\sum_{l\in C_k}\frac{1}{P+C}\frac{e_{ik}}{e_l\prime}]$$

Once again, if it is assumed that $e_k\prime \approx e\prime$ for all k then

$$m(S,t+1)=\sum_{k=1}^{N}\sum_{i=1,i\in S}^{P}\frac{e_{ik}}{e\prime}\frac{(P+C)}{(P+C)}$$

$$=m(S,t)\frac{e_S\prime}{e\prime} \quad .$$

The assumption $e_k\prime \approx e\prime$ for all k, which is made in the preceding theoretical discussions, needs to be investigated further. This assumption is experimentally validated in the following section.

5 Experimental Investigation of Assumption

In investigating the assumption, $e_k\prime \approx e\prime$ for all k, experiments were run on DeJong's functions f1, f2, f3, and f5. The results of these experiments are presented in figures 1 - 4 respectively. These experiments were run on a hypercube multiprocessor containing 8 nodes. The subpopulations at each node contained 50 individuals. Each graph represents the average of 40 runs of 100 generations each, and shows the maximum difference between all population averages, the maximum difference between neighboring population averages, the average difference between neighboring population averages, the minimum difference between neighboring population averages, and the minimum difference between all population averages. Although the theory is concerned only with neighboring population averages, all five measures show the tendency to converge to zero. Thus it appears that $e_k\prime \approx e\prime$ for all k is a reasonable assumption.

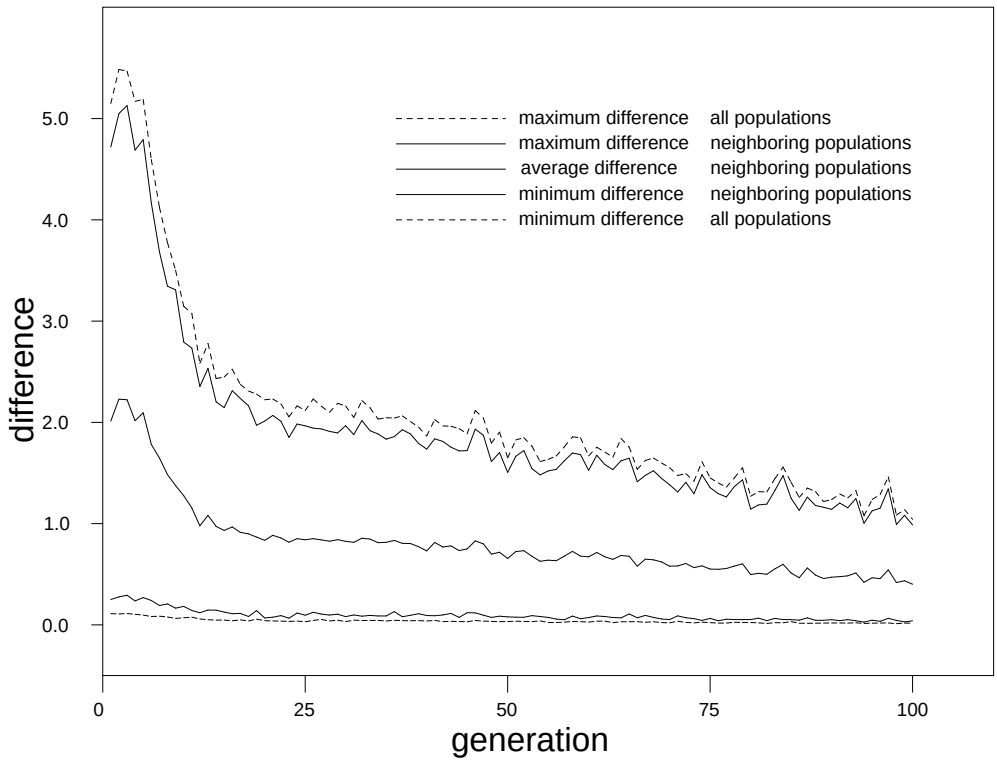

Figure 1: Function F1 Using Uniform Distribution for Communication

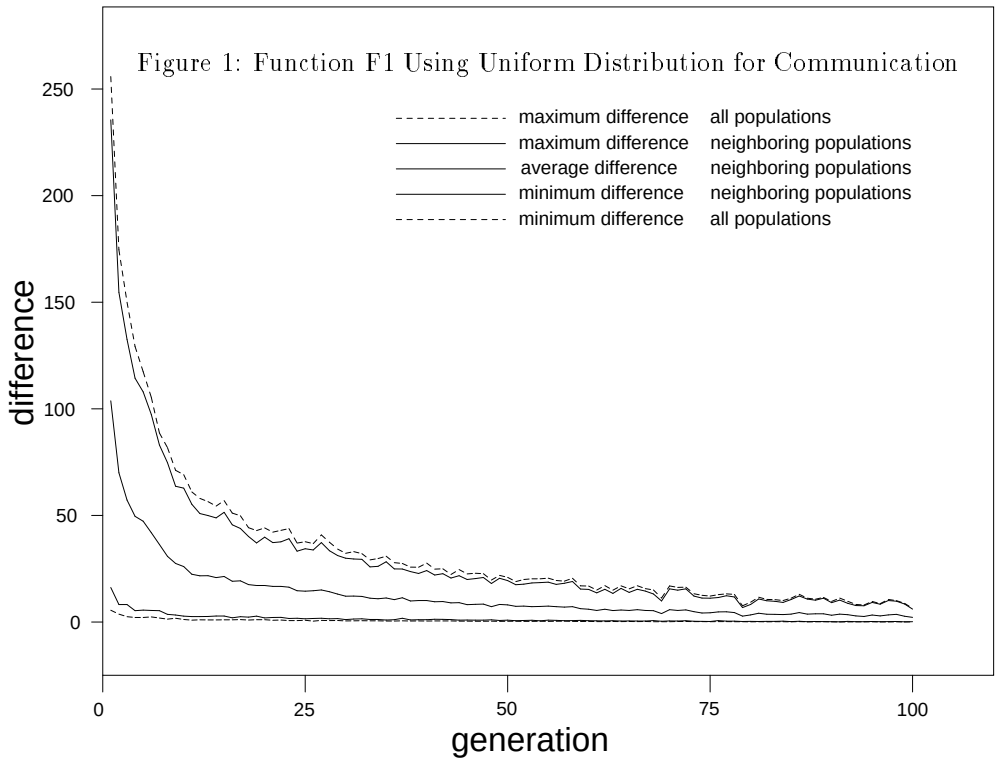

Figure 2: Function F2 Using Uniform Distribution for Communication

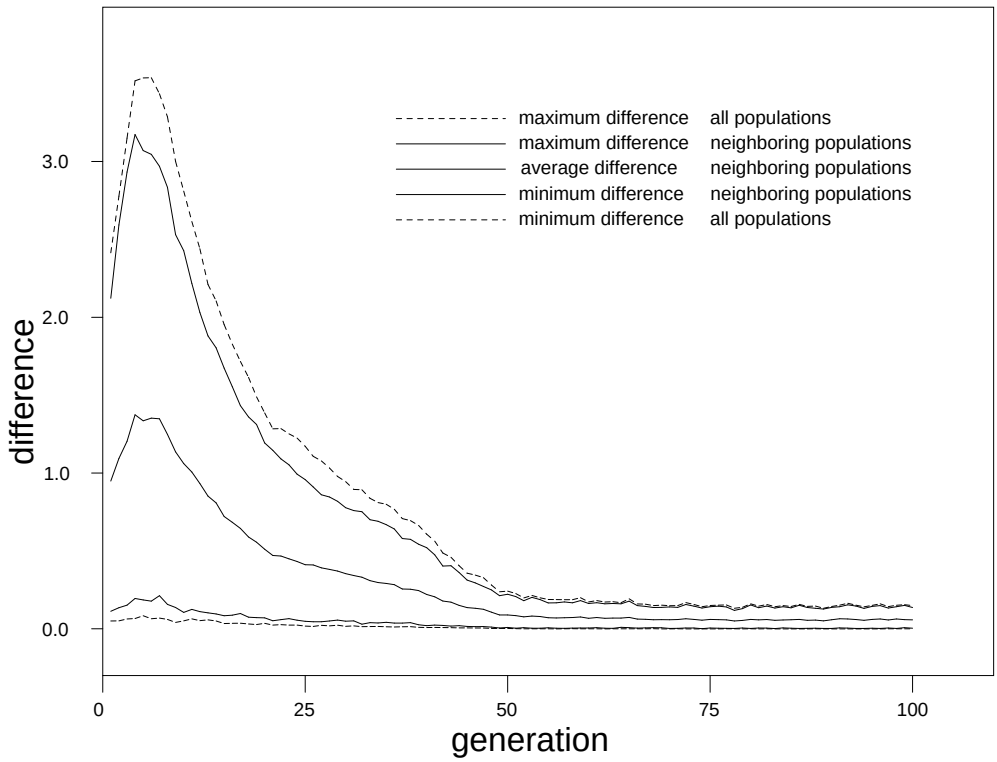

Figure 3: Function F3 Using Uniform Distribution for Communication

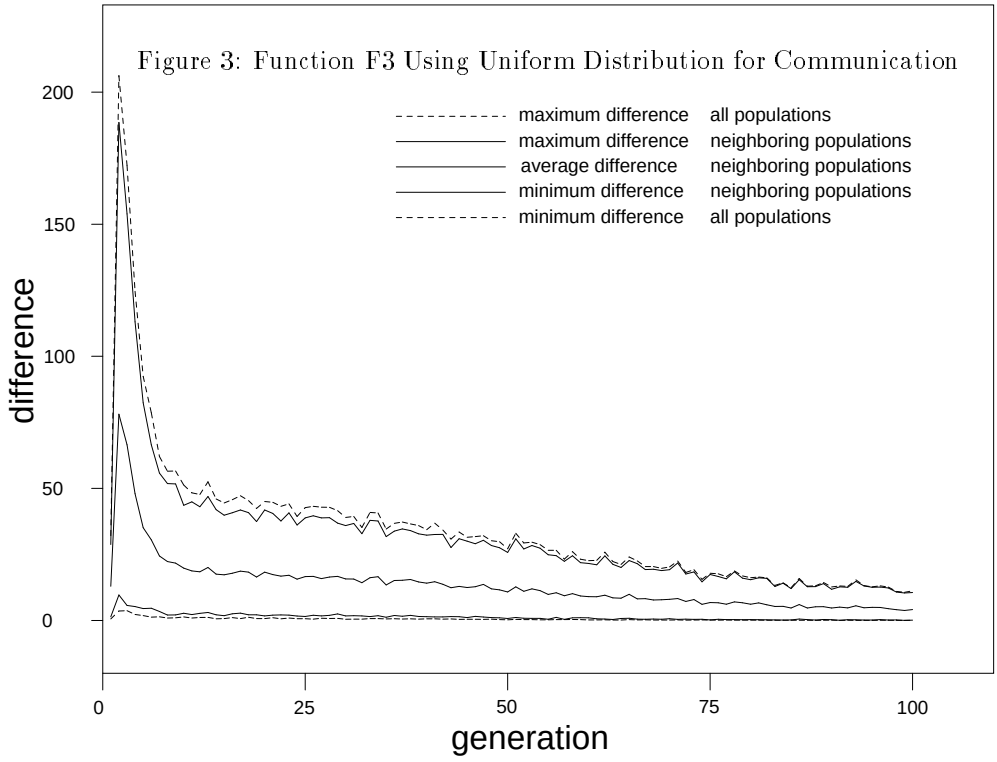

Figure 4: Function F5 Using Uniform Distribution for Communication

6 Conclusions and Future Research

This research shows that a PGA that uses a uniform distribution for communication tends to maintain the efficiency that is observed in a standard sequential GA. While this result does not say anything about PGA's in general, it does provide a theoretical basis for one parallel algorithm. Furthermore, from the preliminary results of section III, it is shown that the allocation of trials to schemata by PGA's in general can be bounded above by an exponentially increasing (decreasing) function.

Work is continuing in this area. Presently, two other means of communication are being investigated along with experimentation on DeJong's f4 and the Traveling Salesman Problem. Also of interest to the authors is a comparison of PGA's to the recent theoretical work of Baker [Baker 88].

References

[Baker 88] James Edward Baker, *An Analysis of the Effects of Selection in Genetic Algorithms*, Ph.D. Thesis, Department of Computer Science, Vanderbilt University, December 1988.

[Cohoon 87] J. P. Cohoon, S. U. Hegde, W. N. Martin, and D. Richards, Punctuated Equilibria: A Parallel Genetic Algorithm, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1987, pp.148 - 154.

[DeJong 75] Kenneth A. DeJong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Ph.D. Thesis, Department of Computer and Communication Sciences, University of Michigan, 1975.

[Dobzhansky 70] Theodosius Dobzhansky, *Genetics of the Evolutionary Process*, Columbia University Press, New York, 1970.

[Dolan 87] Charles P. Dolan and Michael G. Dyer, Toward the Evolution of Symbols, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1987, pp.123 - 131.

[Grefenstette 81] John J. Grefenstette, *Parallel Adaptive Algorithms for Function Optimization (Preliminary Report)*, Technical Report CS-81-19, Computer Science Department, Vanderbilt University, November 1981.

[Grefenstette 86] J. Grefenstette and C. Pettey, Approaches to Machine Learning with Genetic Algorithms, *Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics*, October, 1986, pp.55 - 60.

[Hilliard 88] Michael R. Hilliard, Gunar E. Liepins, Mark Palmer, and Gita Rangarajan, The Computer as a Partner in Algorithmic Design: Automated Discovery of Parameters for a Multi-Objective Scheduling Heuristic, *Proceedings of Impact of Recent Computer Advances on Operations Research*, 1988.

[Holland 75] John H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Michigan, 1975.

[Jog 87] Prasanna Jog and Dirk Van Gucht, Parallelisation of Probabilistic Sequential Search Algorithms *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1987, pp.170 - 176.

[Oosthuizen 87] Deon G. Oosthuizen, SUPERGRAN: A Connectionist Approach to Learning, Integrating Genetic Algorithms and Graph Induction, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1987, pp.132 - 139.

[Pettey 87a] Chrisila C. Pettey, Michael R. Leuze, and John J. Grefenstette, Genetic Algorithms on a Hypercube Multiprocessor, *Hypercube Multiprocessors 1987*, SIAM.

[Pettey 87b] Chrisila B. Pettey, Michael R. Leuze, and John J. Grefenstette, A Parallel Genetic Algorithm, *Genetic Algorithms and Their Applications: Proceedings of the Second International*

Conference on Genetic Algorithms, 1987, pp.155 - 161.

[Pettey 88] Chrisila C. Pettey and Michael R. Leuze, Parallel Placement of Parallel Processes, *Proceedings of the Third conference on Hypercube Concurrent Computers and Applications*, 1988, pp. 232 - 238.

[Robertson 87] George G. Robertson, Parallel Implementation of Genetic Algorithms in a Classifier System, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1987, pp. 140 - 147.

[Sannier 87] Adrian V. Sannier II and Erik D. Goodman, Genetic Learning Procedures in Distributed Environments, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1987, pp. 162 - 169.

[Tanese 87] Reiko Tanese, Parallel Genetic Algorithms for a Hypercube, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1987, pp. 177 - 183.