The DEME Model: An Asynchronous Genetic Algorithm

Victor Coleman
Master's Project Report
Computer and Information Science Department
University of Massachusetts

COINS Technical Report 89-35

May 12, 1989

The DEME Model: An Asynchronous Genetic Algorithm
Master's Project Report

Victor Coleman

May 12, 1989

Abstract

A genetic algorithm is a problem-solving method in which points in the search space are encoded as a population of bitstrings called chromosomes. Like evolution in genetics, from which the algorithm takes its name, the algorithm produces successive generations of chromosomes by applying a series of reproductive operators. Chromosomes that are more 'fit' are allowed to reproduce with a higher probability, where 'fitness' is a relative measure of which chromosomes are closer to solution points in the search space. The primary reproductive operators, crossover and mutation, are then applied to the population of chromosomes to produce a whole new population set of the same size. The algorithm stops when the search no longer seems to be advancing; the assumption being that a global maximum has been reached.

Since genetic algorithms are similar to evolution in nature, they are utilized not just as a search technique in computer science, but as biological models of evolution within various subdisciplines of biology: genetic algorithms serve a dual purpose in science.

We propose a variant of the genetic algorithm which offers an improvement in search performance, as well as being more highly parallelizable, for the needs of computer science, with the added benefit for biology of being a more accurate model of natural evolution.

1 Introduction

This paper introduces and investigates an asynchronous parallel variation of the standard genetic algorithm [Holland, 1975]. It is called the DEME model. Grefenstette has proposed a set of software engineering approaches to parallelize genetic algorithms [Grefenstette, 1981]. The DEME model is most similar to Grefenstette's asynchronous concurrent genetic algorithm, insofar as it is also asynchronous. Otherwise, the goals and implementations of the two are quite different.

The standard algorithm is transformed to the DEME model by allowing asynchronous reproduction to occur within subgroups of the population, rather than enforcing it on the

whole group. A *deme* is the biological term for a reproductive subgroup, giving us the name of the model. The mechanism for forming demes is a side effect of the different speeds at which simultaneously executing processors complete subtasks. Unlike the standard algorithm, the DEME model is necessarily a parallel algorithm, as it requires asynchronicity in the system. Other variants of the standard algorithm have been proposed in which demes are formed, for instance [Goldberg and Richardson, 1987]. These criteria for deme formation have fallen under the general heading of spatial isolation. Again, this term borrows from biology. When demes are formed in a natural species it is often due to some barrier in the environment, such as a river, or mountain range. Demes are then said to have arisen through spatial isolation. Thus, at first hand, it might seem appropriate to claim that the deme model proposes a novel mechanism for spatial isolation. However, the DEME model uses temporal, and not spatial isolation. Temporal isolation does not require that organisms be spatially isolated. Instead, at any point in time only a subset of the species is ready for reproduction, i.e., has reached adult reproductive maturity, and so only they form the deme. Where immature organisms happen to be spatially located at that moment is beside the point, as they cannot reproduce. The DEME model implements temporal isolation by assigning each chromosome to a different processor and running them asynchronously. When a chromosome reaches the reproductive stage, its processor then synchronizes with other ready processors to form the deme and reproduce among themselves. The size of the deme is a parameter of the algorithm. Note that when the deme size is equal to the population size, the DEME model is identical to a parallel version of the standard algorithm.

For computer science the distinction between spatial and temporal isolation appears insignificant. Both mechanisms produce the same results as long as the demes formed are identical in every case; just as a selection sort and an insertion sort are functionally equivalent. As long as there is a standard ordering of elements (a, b, c, etc.) both sorts produce the same result (if elements are all distinct). As with the sorting algorithms, for computer science the merits of temporal versus spatial isolation are to be found in resource utilization (time and space). But the distinction between temporal and spatial isolation is crucial for biology. Generally, spatial isolation is considered central for the formation of new species, whereas temporal isolation is rarely mentioned [White, 1978]. The DEME model offers biology a tool for reassessing the importance of temporal isolation and its role in speciation.

The reader should keep in mind the contrasting and sometimes opposite interests of computer science and biology. For computer science, the DEME model advances a novel mechanism for deme formation, which is shown to execute more quickly and improve searching performance over the standard algorithm. For biology, the deme model is an improvement over synchronous evolutionary models, in that the effects of temporal isolation can be observed.

Experiments in this paper compare the DEME model to the standard algorithm using De Jong's five functions as a testset [De Jong, 1975].

2 Introduction to Genetic algorithms

To understand genetic algorithms and their role as evolutionary models it is necessary to understand some terms from basic biology.

2.1 Genetics

Chromosomes, strands of DNA, direct the synthesis of proteins that ultimately determine (along with environmental conditions) the construction of an organism. Chromosomes are composed of genes, which are the sub-units responsible for individual traits, e.g., eye color. Alleles are the different values which a gene may take on. The genotype is the set of chromosomes of an organism. The phenotype is the organism characterized by all of its physical traits after a developmental stage. Haploids have one full set of chromosomes, whereas diploids have two full sets, one from each of the parents. Reproduction is either sexual, or asexual, depending on the number of parents. During reproduction, with some probability, pairs of chromosomes will swap parts of their strands with each other. This is known as crossover. The sole source of allele variation is due to random changes in the value of a gene, known as mutation. This is not to say that evolution is always random, as some gene values may be favored by the environment more than others, even though they appeared randomly.

2.2 Evolution

Natural selection is the primary mechanism for the evolution of organisms in nature. A sufficient condition for evolution by natural selection is contained in three propositions:

1. Variation: There exist differences in the genetic makeup among the members of a species.
2. Heritability: This variation is in part heritable, so that individuals resemble their parent(s).
3. Differential fitness: Different variants leave different numbers of offspring either in immediate or in future generations [Lewontin, 1984, p. 244].

A second mechanism for evolution is random drift. A population undergoes chance fluctuations without differential fitness. There is disagreement among evolutionary biologists as to the relative importance of these mechanisms. Some have come to believe that the role of random drift, or non-Darwinian evolution, has been underestimated, that is, that more evolution is to be attributed to random drift than was previously thought. However, by and large, natural selection is viewed as the primary of the two. Neo-Darwinism, the received view in evolutionary biology, combines Darwinism and the advances made in genetics from Mendel and afterward in the 20th century. Prior to the discoveries of genetics it was unclear how variability arose, and how traits were passed on from parents to offspring.

Darwinism can be contrasted with Lamarckian theories which postulate that mutation is not random, but goal-directed. The attractiveness of Lamarckism is that it is difficult for some to imagine undirected random mutation as being sufficient to fuel variability for the evolution of highly complex organisms. Lamarckism remains unfavored in most scientific circles.

2.3 The standard genetic algorithm

The standard genetic algorithm models sexual reproduction of haploid organisms. There is no gender associated with an individual. Each gene on the chromosome can have one of two alleles, or values, 1 or 0. The number of chromosomes in the population is fixed from generation to generation. No distinction is made between genotype and phenotype, as the values of a chromosome's genes are identical with its traits. Said differently, a chromosome is the same as the organism; there is no developmental stage. Chromosomes are designed to encode parameter values for the objective function under consideration. The objective function describes the shape of the search space. The goal for a genetic algorithm, like other search techniques, is to find the maximum/minimum of the objective function. Developing an objective function that describes the search space for a particular problem is difficult. Therefore, testing the effectiveness of a genetic algorithm is often done by applying it to an objective function whose maximum/minimum is known beforehand. The progress of the algorithm can then be measured in terms of how close and at what rate it arrives to the known solution. This also circumvents the task of neatly mapping a problem into an objective function.

Once a chromosome is decoded, it can be evaluated by the objective function and compared to the evaluations of other chromosomes in the population. Mimicking natural selection, chromosomes that are more successful (greater evaluation score) are allowed to reproduce more often. Reproduction pairs two chromosomes, crosses them over starting from some randomly chosen gene all the way to the ends, and produces two new ones. Essentially, if the chromosomes are strings of bits, crossover swaps the tail end of the two parent-strings starting at the randomly chosen gene. Consider an example of crossover:

- Father: 111100
- Mother: 010110
- Random point: Third bit from the left
- Offspring 1: 110110
- Offspring 2: 011100

The progeny are composed entirely of their parents' genes. Mutation is modeled by flipping the value of each gene in every chromosome with a certain probability, usually quite small (0.01 or less). Here is the standard genetic algorithm [Goldberg, 1988]:

1. Initialize population of chromosomes with random values.

2. Decode and evaluate chromosomes.

3. Produce the next generation of chromosomes (same size), by biasing reproduction in favor of chromosomes with greater evaluation scores.

4. Induce mutation.

5. If termination condition is met, stop, otherwise goto step 2.

In some implementations the crossover step is omitted with a certain probability (usually less than 0.4) for each reproducing pair. In this case the 'offspring' are identical to the 'parents'. Commonly, the algorithm is allowed to run a fixed number of cycles, or generations. The method for biasing reproduction toward the fitter is termed the selection algorithm. The standard selection algorithm is the 'spinning-wheel' method [Baker, 1987]. Finally, genetic algorithms do not consider extinction. No matter how poor a chromosome, if it is better than others, it is considered successful enough to reproduce. In population genetics, this is known as relative fitness. Nature is not as forgiving, since being better does not necessarily imply survival and reproduction. For a more thorough introduction to genetic algorithms, and related research, see [Goldberg, 1988].

2.4 Isolating mechanisms and speciation

One of the most important questions in biology is why and how so many different kinds of organisms exist, i.e., how do species arise. Speciation is thought to be a consequence of isolation. Isolating mechanisms fall under two categories, spatial and non-spatial (allopatric and sympatric, respectively). Spatial isolation in nature occurs when members of a species are separated by physical barriers, such as canyons, rivers, land distance, etc. The result is a division of the whole population into subpopulations called *demes*. A deme is defined as a local interbreeding unit [Hartl, 1988]. Demes will tend to become dissimilar from other demes over time. As each deme specializes to the idiosyncrasies of its own niche it can eventually form a new species. Inter-deme migration counteracts that effect by spreading chromosomal differences. For instance, if individuals migrate to other demes for reproduction, then they carry with them any novel chromosomal differences that may exist between the demes, preventing the process of speciation from arising. How much isolation is needed for speciation, and how much migration prevents speciation, are questions that biologists continue to explore.

An advantage of smaller groups is that they can react more quickly to their environment, since there are fewer chromosomes competing for propagation in the next generation. However, there are fewer directions in which a smaller deme can change because there is less chromosomal diversity. But inter-deme migration counteracts this defficiency by introducing novel chromosomes between the demes. So, one might expect that some combination of isolation and inter-deme migration will be more effective (in terms of survivability and reproduction) than having only one reproductive group composed of the entire popultion.

Temporal isolation occurs when organisms have different breeding seasons, or cycles. For example, the European forest cricket has a life cycle of two years. Those who mate in

odd years seldom meet the even ones so that even if they live in the same forest, they are different demes [White, 1978, p. 250]. Occasionally, the crickets reproduce a year early, or late, which is the analog of inter-deme migration under spatial isolation.

Spatial and temporal isolation need not be mutually exclusive. For instance, suppose we divided the crickets into groups by placing barriers throughout the forest. Within each group there is temporal isolation with odd and even crickets. Throughout the forest there is spatial isolation induced by the barriers. This kind of reproductive complexity is more akin to real situations in nature.

3 The DEME model

To determine individual fitness it is first necessary to compute the population average during the reproductive phase of the standard genetic algorithm. For parallel implementations, this is the primary serial bottleneck as the average cannot be computed until all individuals have completed function evaluation[Bethke, 1976]. This is true regardless of the number of available processors. This is more than just a speed limitation, if the genetic algorithm is intended to accurately model evolution. In nature, the reproductive success of an individual is not necessarily contingent on the success of every last one of his peers, although it may sometimes be so. To compensate for these shortcomings, the author has constructed the DEME model. The key difference between it and the standard algorithm is that the size of the reproductive group is not fixed to be the whole population, but can also be set to some subgroup, or deme. In the special case where the deme size is the entire population we have exactly the standard genetic algorithm described above. Thus, the DEME model is more general, insofar as it encapsulates the standard algorithm. What was previously the main serial bottleneck has now been reduced. Computing the fitness of an individual does not have to wait for the function evaluation of the rest of the population, but only for a deme. The smaller the deme size the greater the number of different demes that can be reproducing in parallel.

Assume that we have a different processor assigned to each member of the population. The population, as with the standard algorithm, is initialized randomly. Each processor evaluates its assigned chromosome. Upon completion, a processor enters a queue set to the desired size of a deme. When the queue fills, those processors in the queue form a deme, and reproduce in exactly the same way as the whole population would reproduce in the standard algorithm. After reproduction, the cycle starts again, the new chromosomes are evaluated, and the reproduction queue is reentered. The queue serves as a mechanism to group demes for reproduction.

3.1 Genetic algorithms as evolutionary models

The development of genetic algorithms was and continues to be inspired by biology [Holland, 1975]. The opposite is also true. Biologists of various sub-disciplines are increasingly borrowing from the genetic algorithm literature within computer science to construct evolutionary models. Previous evolutionary models have focused on tracking gene frequen-

cies in a formal system with a set of simplifying assumptions associated with the Hardy-Weinberg model[Hartl, 1988]. One successively violates the assumptions to measure their contribution to the system's behavior, as a whole. Genetic algorithms differ from this variety of models in several key respects. Instead of concentrating on tracking gene frequencies, chromosomes are explicitly manipulated in a complex search space, with a set of operators that closely mimic the reproductive operators of true organisms at the chromosomal level. Hardy-Weinberg based models have no such operators, no explicit notion of a search space, and an inherent difficulty in working with more than a small number of genes (usually, fewer than five). On the other hand, genetic algorithms offer the possibility of studying the non-linear interactions of thousands of genes, as well as providing gene frequencies at each time step of the algorithm. In addition, there is a well-established body of computer science theory to explain those interactions [Holland, 1975]. These factors make genetic algorithms an excellent tool for studying evolution. They are simulations that go beyond incomplete analytical models.

The DEME model takes one step beyond the standard algorithm in modelling evolution by introducing asynchronous reproduction. This takes into account the fact that, in nature, members of a species do not all reproduce simultaneously. Also, the fitness of an individual is not necessarily contingent on the success of all other members of the species. Synchronous evolutionary models including the standard genetic algorithm are committed to view the survival of organisms in a population as a strict competition for limited resources. This is so because the formation of the next generation during reproduction forces each organism to be evaluated in comparison to all other members of the population. To the extent that one organism is considered more fit, other organisms will be less fit. Each chromosome competes to have itself copied into the next generation, but as there are limited number of available slots, the success of some implies the failure of others. But there are numerous scenarios in nature that are unlike this, namely, whenever there are demes and a relative abundance of resources. This limitation as a biological model is in part a result of a fixed population size. Unfortunately, a dynamic population size is largely unexplored in the field of genetic algorithms, and remains an open problem. However, the DEME model offers an alternative, if only partial, solution. The DEME model extends the standard algorithm by introducing temporal isolation. If some mix of isolation and migration does improve evolutionary progress in nature, then we expect that, likewise, the DEME model will perform a more effective search. In this case, it would constitute a better search method for computer science.

For biology, we can use the DEME model to simulate the evolutionary effects of isolation. Note that spatial and temporal isolation have the same consequences. If we conclude that spatial isolation can induce speciation, it follows that temporal isolation can do so, as well. However, biologists generally do not believe that temporal isolation plays a key role in speciation, and its effects on evolution, as a whole, have been largely ignored. Simultaneously, spatial isolation is given a prominent role in speciation [White, 1978]. The assumption is that when temporal isolation does occur it is accompanied by enough migration to prevent the formation of new species. There is good empirical evidence for this

belief. However, this says nothing about how temporal isolation affects evolution as a weak force, even if it is not the primary mechanism of speciation. In part, temporal isolation has been ignored because the focus in biology has been to answer how speciation arises, downplaying evolution within a species. Additionaly, temporal isolation is more difficult to track empirically than spatial isolation, and has thus resisted any research. For example, a dog breeder can observe spatial isolation at work by permanently separating his kennel into groups for an extended period of time. Possibly, new breeds will arise. The dog breeder need not track anything about the dogs; large fences suffice. On the other hand, to observe temporal isolation as it happens in the kennel he will have to track the whole family tree of every dog. Furthermore, since temporal isolation is typically accompanied by a large amount of migration, it is likely that as a weak force its effects will not be observable in anything other than a very long period of time; perhaps on the order of hundreds or thousands of generations. To test temporal isolation a faster replicating organism is needed, such as bacteria. Unfortunately, it is not possible to track the family tree of each and every one of the bacteria in a culture. Even fruit flies, with which tracking family trees are feasible, may reproduce too slowly to conduct an experiment within a reasonable amount of time. The DEME model is a simulation that takes temporal isolation into account. If temporal isolation is a factor in natural evolution, then the DEME model should produce different results than the standard algorithm without any isolation.

4 Simulation of the DEME model

Since temporal isolation is induced with parallel asynchronous processing the ideal implementation of the DEME model requires a parallel architecture. Unfortunately, no parallel machine with a sufficient number of processors was available to run De Jong's testbed with a population size of 50. It was necessary to devise a simulation for the DEME model to run on a single processor machine. The difficulty with this task is that the performance of the model is directly related to the actual execution time for function evaluation, and deme reproduction, so that these factors cannot be easily simulated. Two clocking alternatives were considered to resolve the problem. Firstly, each chromosome could have its own clock that records the actual execution time devoted to that chromosome by the single available cpu for function evaluation and reproduction. The second alternative also associates a clock per chromosome, but instead of timing the cpu, a number is chosen from a set with a known distribution, mean, and standard deviation. The first method is realistic, insofar as deme formation becomes a consequence of actual variations in cpu execution times. Unfortunately, experiments cannot be duplicated reliably even if the distribution for cpu time is known. Cpu times vary with the load of the system, priorities of the user, and other uncontrollable aspects of a multiuser system. The second method overcomes this complication at the risk of using a distribution that does not reflect how real hardware operates. Nonetheless, the latter was chosen to insure experiments could be controlled, repeated, and were more nearly conclusive.

4.1 Simulation algorithm

The data structure for the population is an array of chromosomes. D is the desired deme size, and P is the population size.

1. Initialize population randomly.
2. Set all chromosome clocks to 0.
3. Evaluate all chromosomes. Each evaluation is timed, and added to the chromosome's corresponding clock.
4. Sort the population array from lesser to greater clock times.
5. Collect the first D chromosomes, and set each of the clocks equal to the clock of the last chromosome that entered the deme (this accounts for the time early processors were spinning).
6. Determine fitness locally and reproduce (selection algorithm and crossover). Add the reproduction time to each clock of the deme members. (Each clock gets increased by the same amount).
7. Induce mutation.
8. Evaluate the new chromosomes. Each evaluation is timed and added to the chromosome's corresponding clock.
9. If termination condition is met, stop, else goto step 4.

The simulation, like the DEME model, is not restricted to the use of a particular selection or scaling algorithm. The parameters of population size, crossover rate and mutation rate are variable, as with the standard algorithm. The deme size must never be smaller than two, or larger than the population size, for the obvious reasons. For generality, the DEME model as well as the simulation were designed to allow for the later modifications sometimes added to the standard algorithm.

5 Experiments and results

The experiments compare the performance and speed of the deme model against the standard algorithm. This comparison was made for two purposes: to show that temporal isolation with migration does have an effect on evolution (biology), and that the DEME model improves the performance and executes more quickly than the standard algorithm.

Following conventions in the literature, there are two measures of performance: offline, and online [De Jong, 1975]. With the standard genetic algorithm, offline performance is the moving average of the best evaluations per generation. Since asynchronicity dissolves the notion of generation (there is no lockstep creation of the next generation), offline for

these experiments is modified to be the moving average of the best evaluations per P function evaluations. When D is equal to P the two offline measures are identical. Online performance is the moving average of every evaluation.

To keep the analysis simple, reproductive time is held constant at D time units, as time for reproduction is linear in the number of reproducing individuals. For instance, a deme of size 10 takes 10 time units to inter-reproduce, whereas a deme of 50 takes 50 units. Function evaluation time is normally distributed with a mean of 1.0 units and a standard deviation of 0.1. Reproductive time is much longer than function evaluation time to approximate what has been observed in practice with real cpu time clocks.

Temporal migration is analogous to spatial migration where organisms travel among more or less established groups, except that in the temporal case a shorter or longer function evaluation time forces that member to fall in with a different crowd the next time it reproduces. Temporal migration is induced solely by the variance in function evaluation time. The larger the variance, the less likely demes will be composed of the same members the next time they reproduce. On the other hand, no variance at all guarantees perfect isolation. However, how variance affects migration over an extended period of time is too complex for these experiments to resolve.

Stochastic universal sampling, called SUS, was used for the selection algorithm [Baker, 1987]. SUS was used for its property of being an optimal sampling algorithm. No attempt was made to parallelize SUS, or to choose known parallel selection algorithms [Baker, 1987].

De Jong's five functions were used as a testset [De Jong, 1975]. Since function 3 has negative values a chromosome's function evaluation score divided by the deme mean cannot be used to compare chomosomes' fitness. Function 3 needs to be scaled. In addition, scaling improves performance for all functions in the later stages of the search, when chromosomes resemble each. Scaling is useful by mapping points that are nearly identical into a larger space so that their evaluations will allow comparisons. Otherwise, their evaluations would be indistinguishable and the search would come to a halt. To overcome these difficulties *linear* scaling was included for these experiments. For a more thourough explanation of scaling see [Goldberg, 1988].

Three configurations of the DEME model were tested with each function:

1. C1: Asynchronous reproduction. Population size of 50. Deme size of 10.
2. C2: Standard algorithm. Population size and Deme size of 50.
3. C3: Standard algorithm. Population size and Deme size of 10.

C1 tests asynchronous reproduction. C2 and C3 are special settings of the DEME model that are equivalent to the standard algorithm, and are intended as the controls of the experiment. C2, like De Jong's experiments, has the a population size of 50. C3 was included to make sure that any advantages C1 had over C2 were not solely due to a deme

size of 10 (independently of population size). If it were, then C3 should perform better than C2. It must be emphasized that apart from these parameter settings there exist no other differences. The crossover rate is 1.0, and the mutation rate is 0.001.

There are three graphs for each function: offline per number of evaluations, online per number of evaluations, and time units per number of evaluations. Since all five graphs for time units per number of evaluations are identical for all five functions they were collapsed into one graph. They are identical because reproductive time is constant, and function evaluation time has the same mean, regardless of the function; the differences were too small to be noticeable.

There are three lines per graph, one for each configuration. Each line is an average of ten runs initialized with different seeds for the random number generator. Each configuration gets the same ten seeds to eliminate any differences that might arise due to better or worse starting populations. For convenience, the minimum of the functions is indicated by the y-value at the origin in the offline and online graphs.

Although results for function 5 are included, they are problematic. None of the three configurations were able to perform better than a random search. This is in conflict with the findings of [De Jong, 1975, p. 249]. We postulate that this inconsistency is due to the differences in the reproduction plan, that is, the selection and scaling algorithms. De Jong's testset was chosen so that readers of the genetic algorithms literature would be familiar with as much of the experiment as possible. C2 was intended to be identical to De Jong's algorithm and every attempt was made to conform. Fortunately, choosing a different reproduction plan did not affect the expected results for functions 1-4, so the informed reader should have no difficulty comparing that set of results. To prevent confusion the remaining discussion is with reference to functions 1-4. In any case, this aberration is not in conflict with the conclusions of this report.

Offline and online graphs should be interpreted in light of the time per number of evaluations graph (graph 11). It indicates that C1 is finished with 10000 evaluations, while C2 and C3 have not yet reached 2000 evaluations. [1] Thus, C1 executes about 5 times faster, since different demes are reproducing in parallel. C1 performs better per unit time if its measure of offline/online at 10000 evaluations is less than the other configuration's measures at 2000 evaluations (if we were maximizing instead of minimizing the function, it would need to be greater). This is always true when comparing C1 and C2. C3 has a better online than C1 for functions 1 and 2, and worse in all other cases (offline and online). This result from C3 is expected for online performance because of its small population size of 10. A larger population may evolve points that are as good, but an average of all its points when many are still poor will reflect a lower score. That C1 did have points closer to the minimum than C3 is evident from the offline graphs. This result would have been unexpected if both the online and offline had been better for C3. However, it is significant that C1 had a better online for functions 3 and 4. This was likely due to variability running

[1] It would have been easier to just provide graphs with offline and online per number of time units, but readers are unfamiliar with that format, and it does not contain information on the performance per number of evaluations.

dry with only a population of 10.

Except for function 4, C2 has a better offline than C3. This implies that C1 outperforming C2 in every case must have been a consequence of temporal migration, since the results for C1 and C3 would be identical without migration. Consequently, the data supports the primary hypothesis of this project: temporal migration improves evolution.

When variability decreases in the later stages of a run, and chromosomes begin to resemble each other, the chances of crossover producing offspring identical to the parents is increased. Whereas most experiments in the literature only evaluate a chromosome when it is known to have changed from the previous generation, in these experiments evaluation always occurs to maintain a controlled variance in function evaluation time. Otherwise, evaluation time for a non-changing chromosome would have to be 0. Since one cannot predict when this will happen, the experiment could not be precisely controlled. The major effect is that in the later stages of a run it was observed that charging a time of 0 tends to clump non-changing chromosomes. Demes continuously being composed of the same members counteracts the focus of the experiment: to observe the results of deme formation with a sustained amount of migration. Even when reproduction and function evaluation times were altered with different distributions, means, etc., migration always decreased over time until demes became isolated for dozens of cycles. Further experiments are required to ascertain that this phenomenon is not the consequence of other parameter settings, such as the population size, crossover rate, and mutation rate. As a search method competing with the standard algorithm, this is a limitation of the DEME model. If variability decreases below some threshold, migration also decreases, which further decreases variability: the DEME model becomes unstable and the search comes to a halt. It may be necessary to modify the DEME model. One possibility is to introduce random wait states into the algorithm. A processor would waste time according to some prespecified probability to increase its chances of falling into mixed demes in successive reproductions. This is partially accomplished in a roundabout way in the current version of the DEME model by making sure that a chromosome is always evaluated, even when it is known that the chromosome is non-changing and the previous function evalution could have been saved. Unfortunately, these mechanisms waste time. A promising alternative is to find better settings for the other parameters. For instance, the deme size could be altered dynamically. Whenever the rate at which variability decreases becomes critical, the deme size would increase until variability restabilized. This method would result in little overhead as an additional processor could act as an observer, only interrupting the rest of system when it became necessary. This option is not included in these experiments.

If a decrease in variability can be predicted and controlled by reducing migration, the DEME model could be instrumental for showing how and when isolation can propel a species toward extinction. This would be useful not just for explaining and predicting evolutionary changes, but to actually bring them about, say in the pest control industry. For instance, the DEME model could simulate the evolution of harmful insects sprayed with insecticides in a large corn field.[2] The population size, crossover and mutation rates

[2]This scenario is included to show readers how genetic algorithms can serve as prominent tools for biology.

can be estimated from samples. The objective function would be designed to reflect how pests have reacted in the past to insecticides, i.e., survival rates: this is the most difficult part of the simulation. The chromosomes would not have to be similar to real ones, but would encode points in the objective function. The simulation would show how migration rates affect survival. Field workers would then have guidelines for weighing the benefits of reducing migration against the costs of insecticides, labor, damage to the environment, etc. Spatial isolation would be implemented with effective physical barriers, and temporal isolation by spraying different insects at different times in such a way that survivors of one blast are less likely to reproduce with survivors of a different blast because their reproductive cycles are not synchronized.

6 Summary

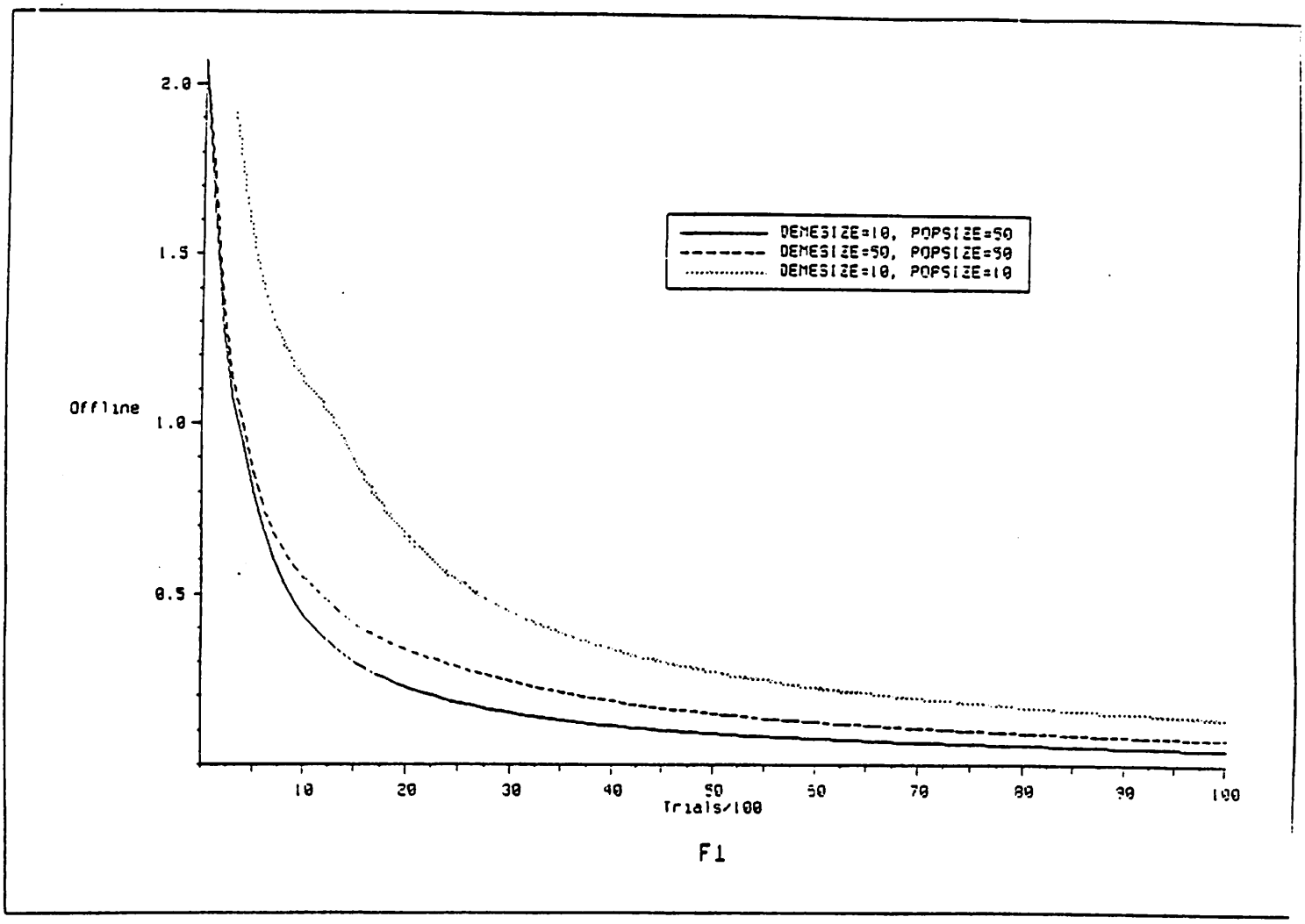
The central hypothesis of this work is that temporal migration improves evolution. The experiments support this. For computer science, the DEME model offers an improved search at a faster rate. For biology, it provides a tool for studying temporal isolation and its effects on evolution.

7 Future work

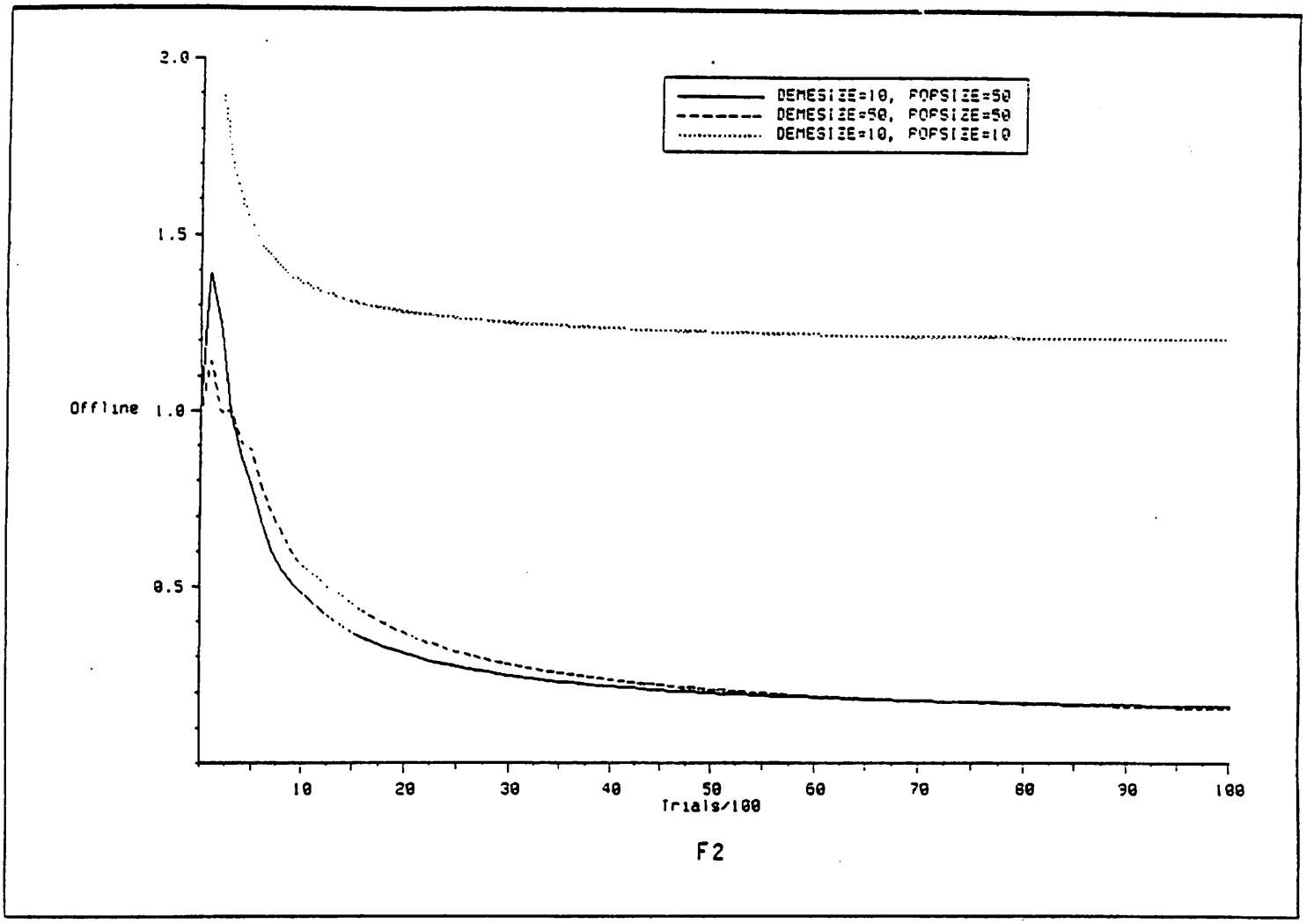
This paper introduces the DEME model and its significance for genetic algorithms and biology. There is much to be done:

- What effect does deme size have on performance and/or speed?
    1. Change the deme size dynamically.
    2. Change the population size dynamically.
    3. How small can the deme size be before the DEME model drops in performance?
    4. What happens when the population size is not evenly divisible by the deme size? (Even though there is no temporal migration, demes would change over time because not all organisms can be reproducing simulataneously. There are always leftover organisms who randomly fall into a deme in the next reproducing cycle).
- Instead of simulating the DEME model, use a real parallel machine.
- How does the DEME model perform with real problems, e.g., the gas pipeline problem?
- While the experiment was being developed it was observed that regardless of the parameter settings demes tended to become more stable over time. That is, migration always decreased over time. Why? Does this happen in nature, or is it a feature of the simulation only?

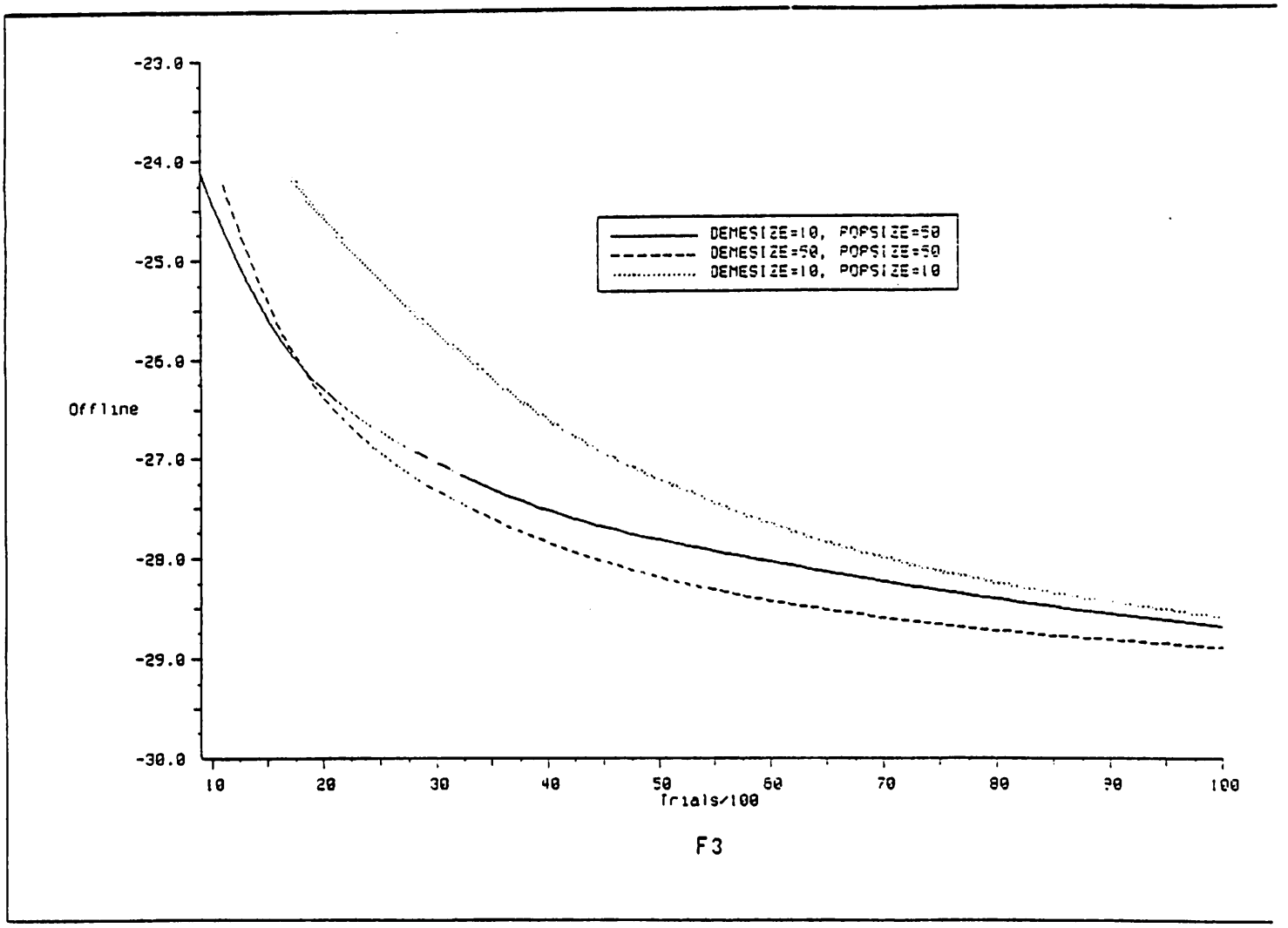
- What are the theoretical underpinnings of asynchronous reproduction? That is, why does the DEME model work better/worse than the standard algorithm? When does it work better/worse? Can the same proofs be made for the DEME model that have been made for the standard algorithm?
- Is the DEME model as sensitive to parameter settings (e.g., mutation rate, selection algorithm) as the standard algorithm?
- How does the performance change when the DEME model is expanded to include other genetic algorithm enhancements?
- What parallel architectures are best suited for the DEME model?
- Construct a fault tolerant DEME model.
- What new mechanisms could be added to the DEME model to maintain the migration level? How does one measure migration?
- Offline and online measures of performance have proven to be weak with the DEME model. Construct more sound measures.
- What effects would combining spatial with temporal isolation have?
- When function evaluation time varies considerably, how does the deme model respond? If certain regions of the search space take less time than others, how does the DEME model respond?
- If the DEME model in asynchronous mode performs worse than the standard algorithm in certain circumstances, what controls need to be added to the model to switch between modes at the right times?
- Conduct a detailed study of how distribution and variance of execution time affects performance.


DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10
Offline
2.0
1.5
1.0
0.5
10
20
30
40
50
60
70
80
90
100
Trials/100

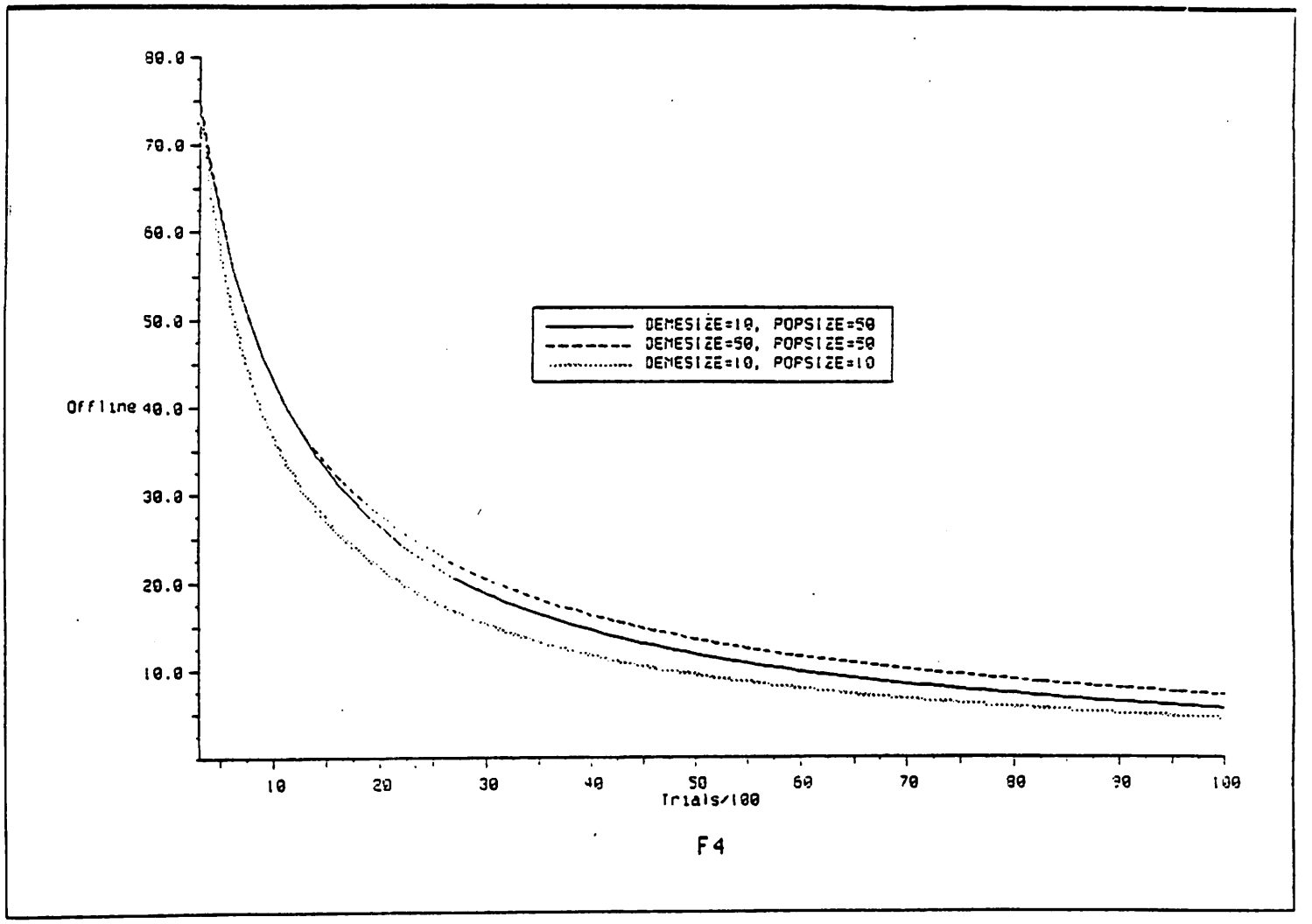
F1


DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10
Offline
2.0
1.5
1.0
0.5
10
20
30
40
50
60
70
80
90
100
Trials/100

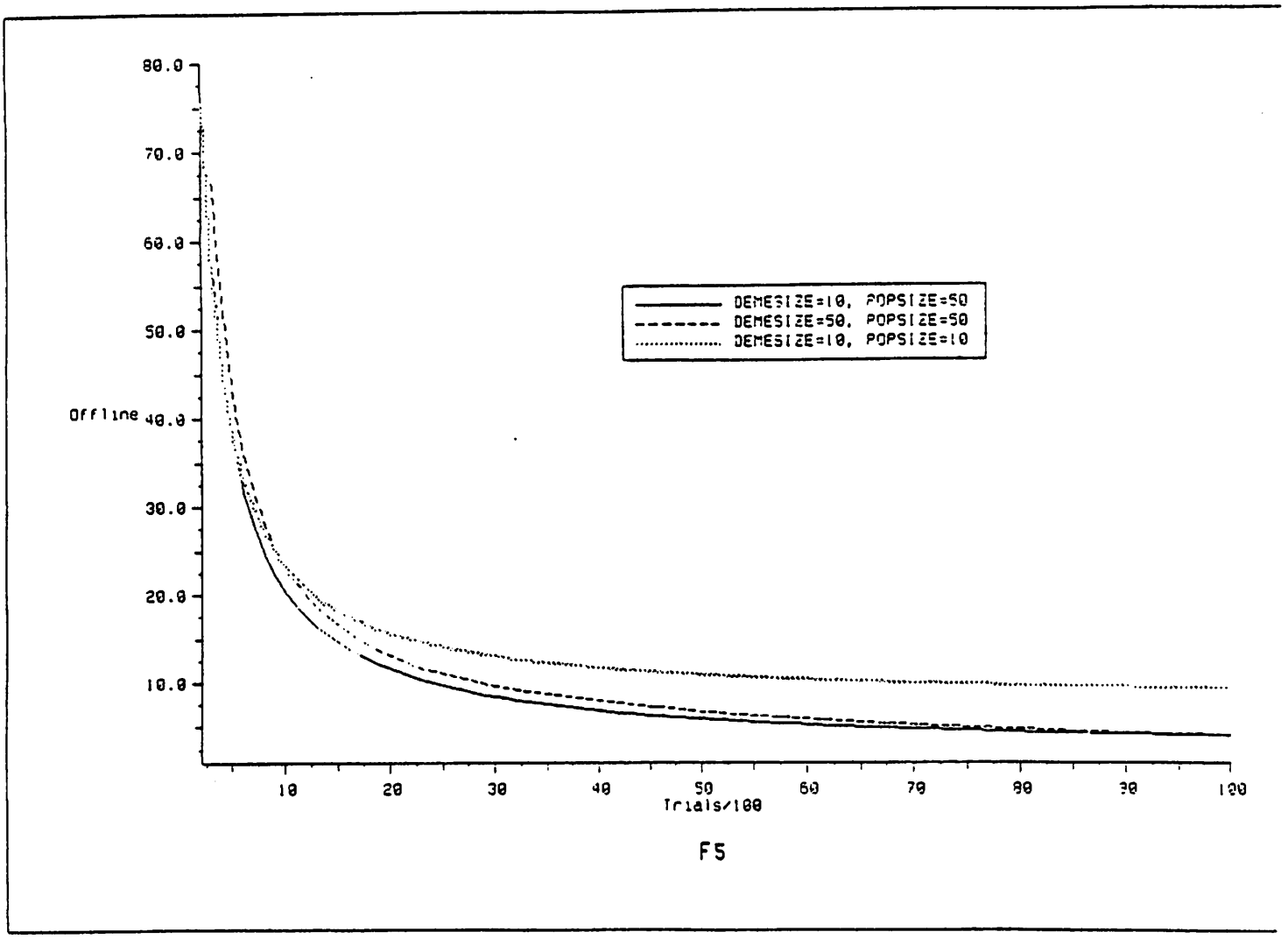
F2


-23.0
-24.0
-25.0
-26.0
-27.0
-28.0
-29.0
-30.0
Offline
10
20
30
40
50
60
70
80
90
100
Trials/100
DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10

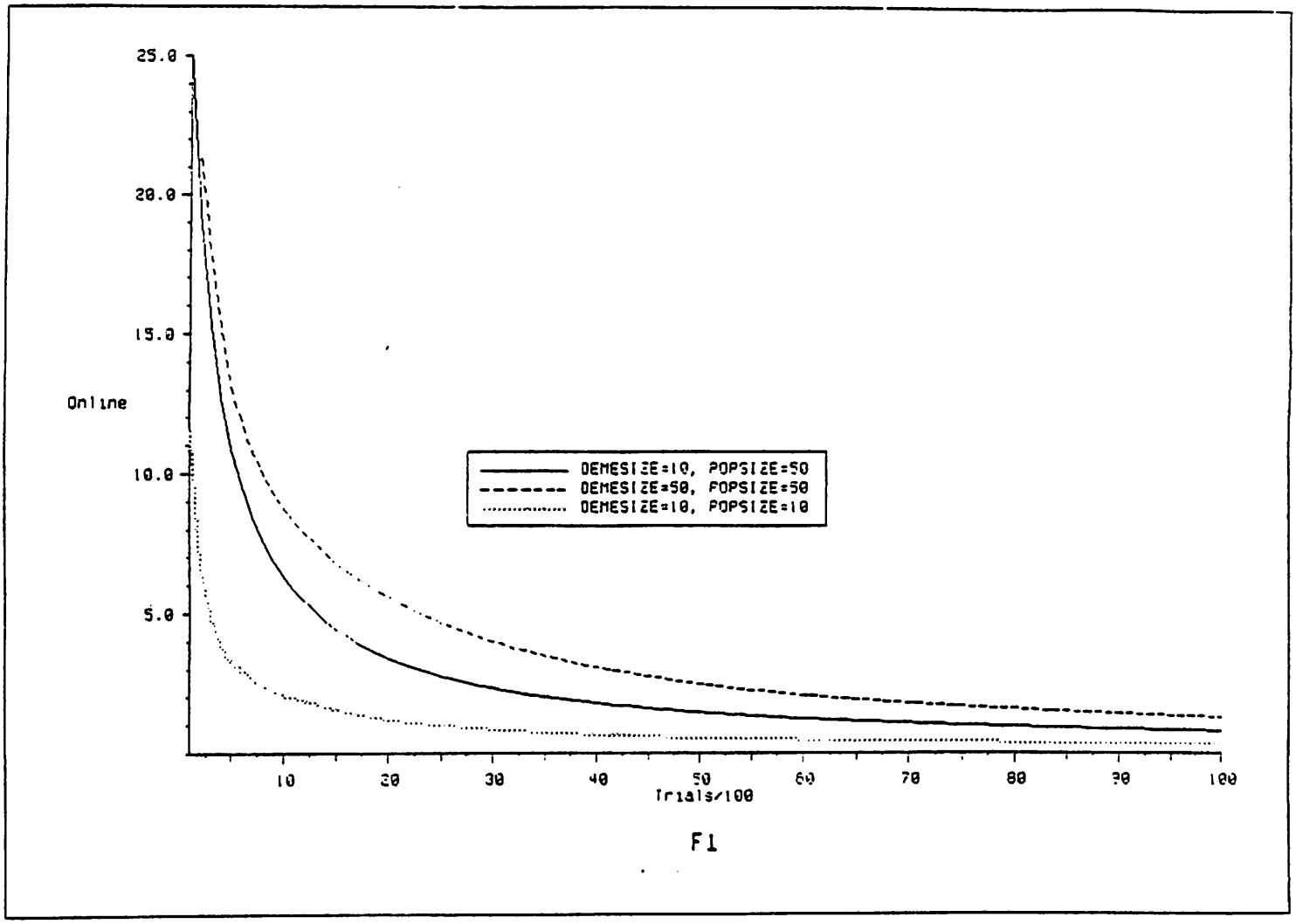
F3


80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
Offline
DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10
10
20
30
40
50
60
70
80
90
100
Trials/100

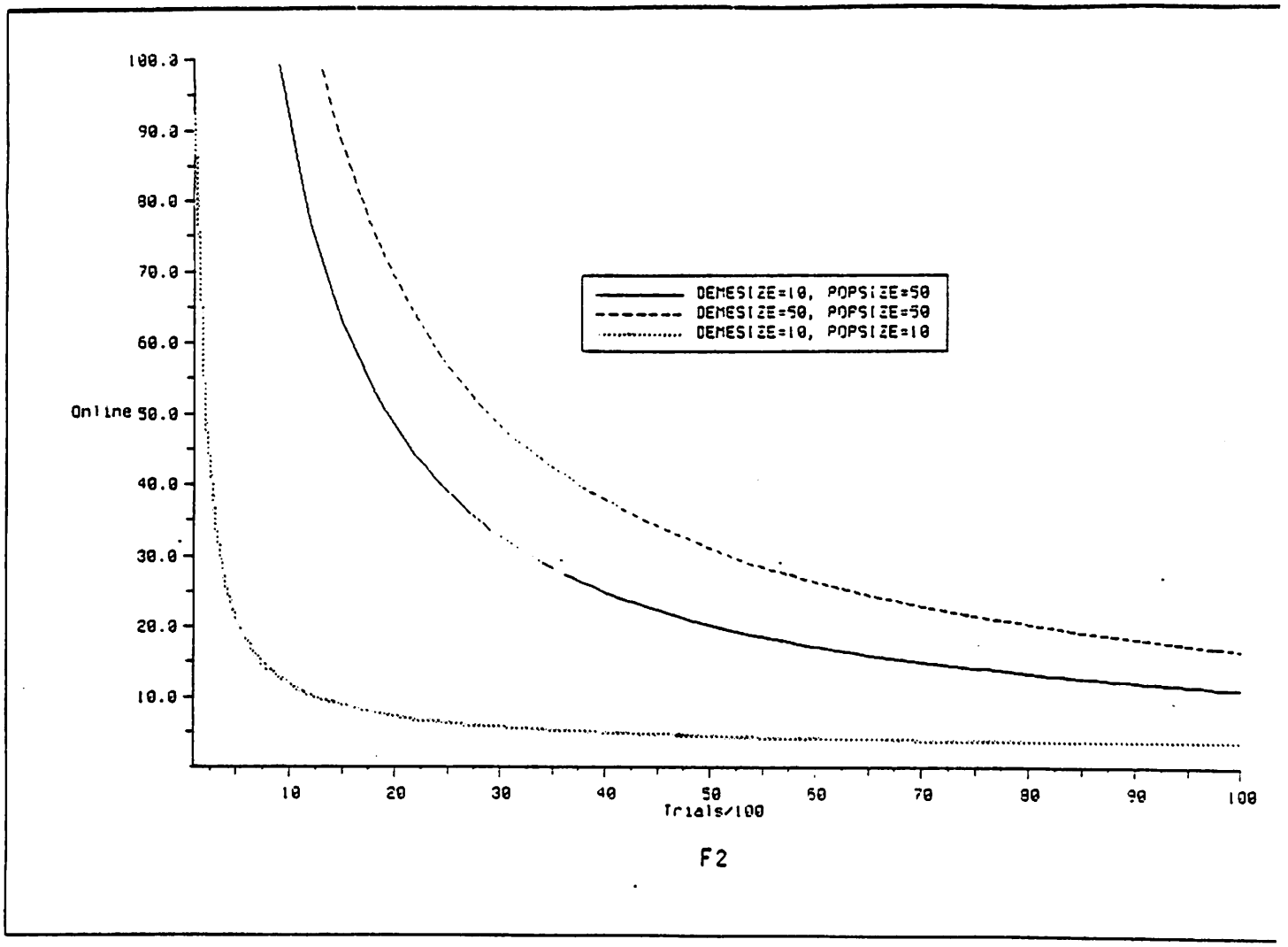
F4


80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
Offline
DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10
10
20
30
40
50
60
70
80
90
100
Trials/100

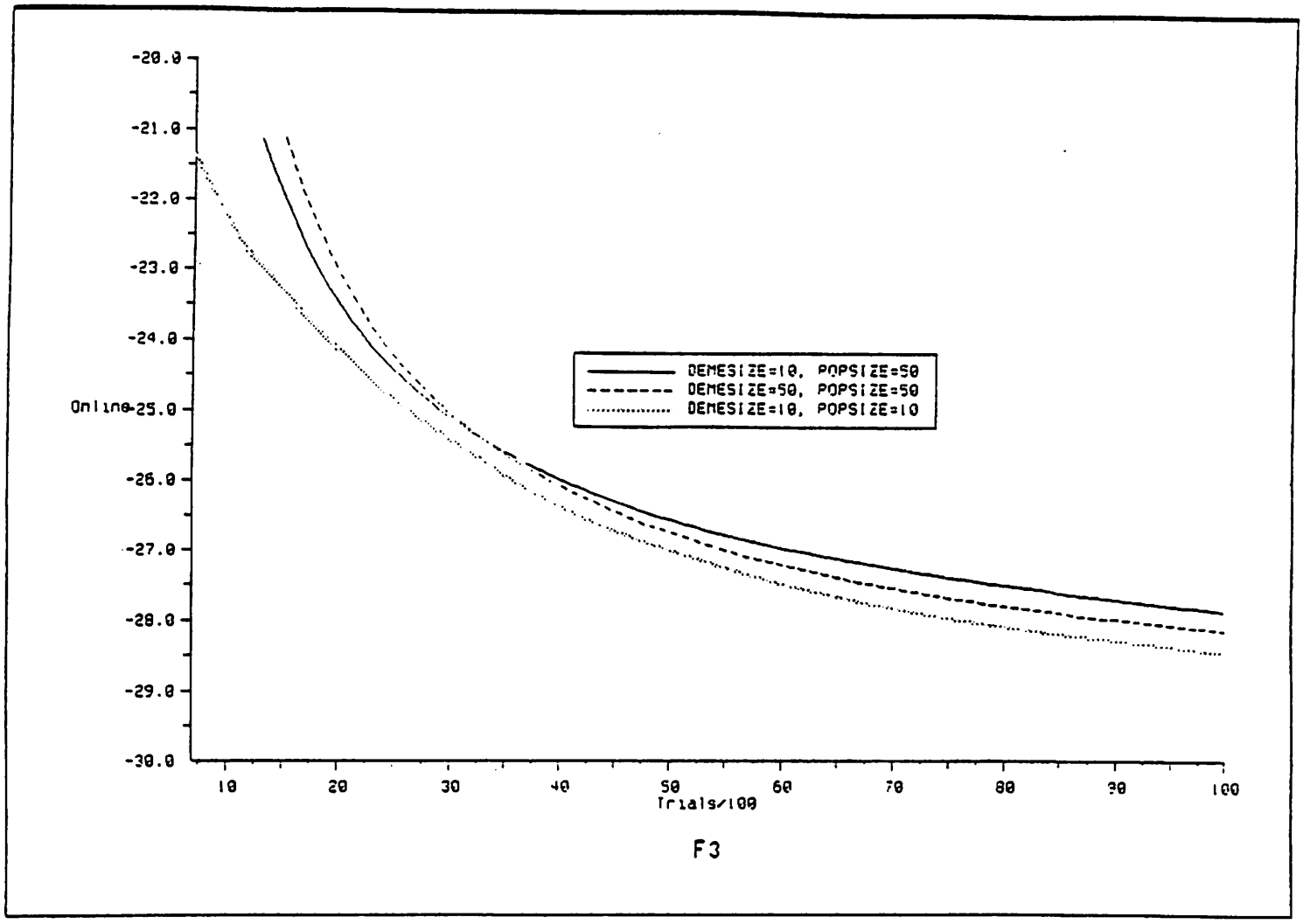
F5


25.0
20.0
15.0
10.0
5.0
Online
DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10
10
20
30
40
50
60
70
80
90
100
Trials/100

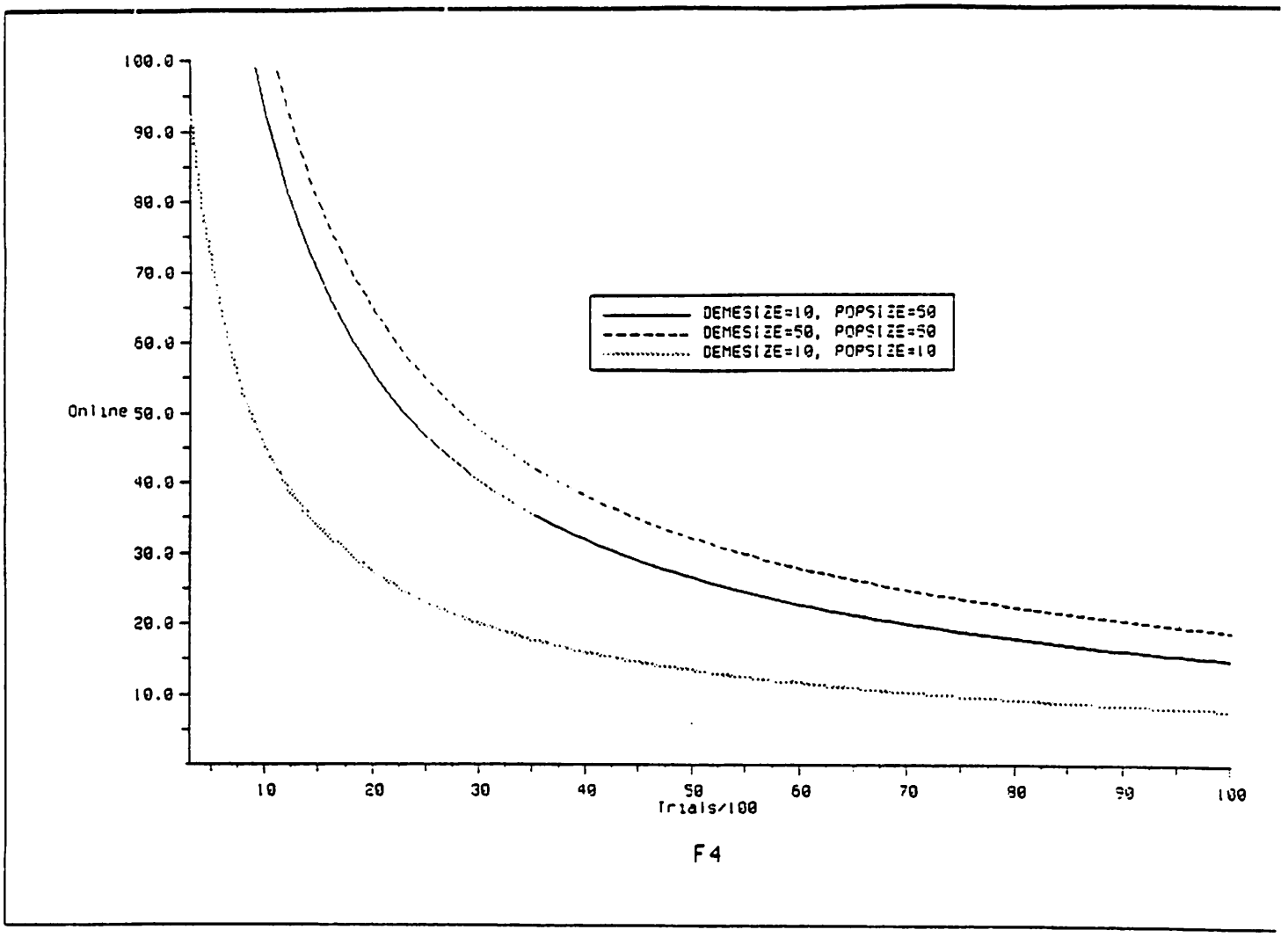
F1


100.0
90.0
80.0
70.0
60.0
Online 50.0
40.0
30.0
20.0
10.0
10
20
30
40
50
60
70
80
90
100
Trials/100
DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10

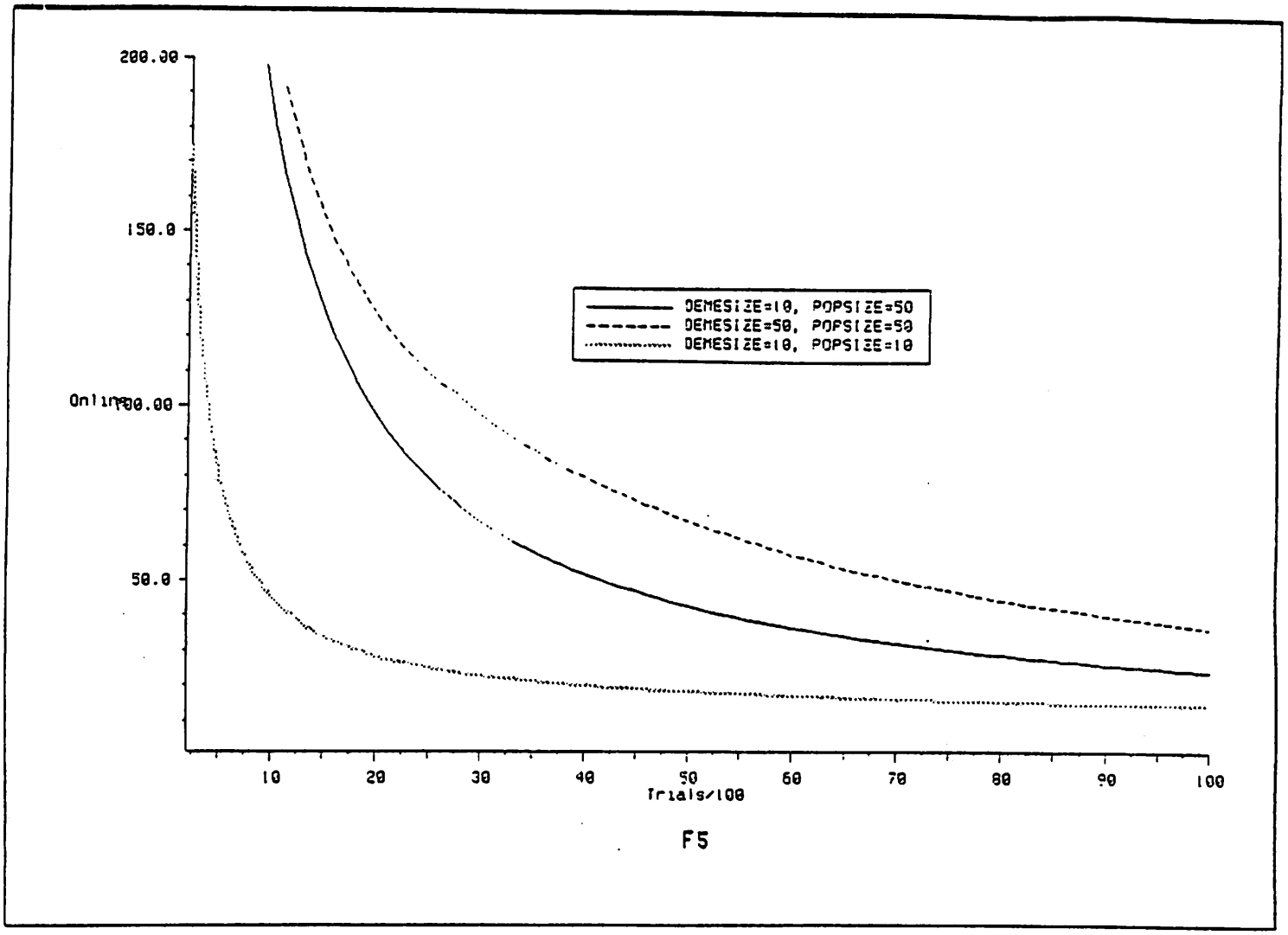
F2


-20.0
-21.0
-22.0
-23.0
-24.0
Online-25.0
-26.0
-27.0
-28.0
-29.0
-30.0
10
20
30
40
50
60
70
80
90
100
Trials/100
DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10

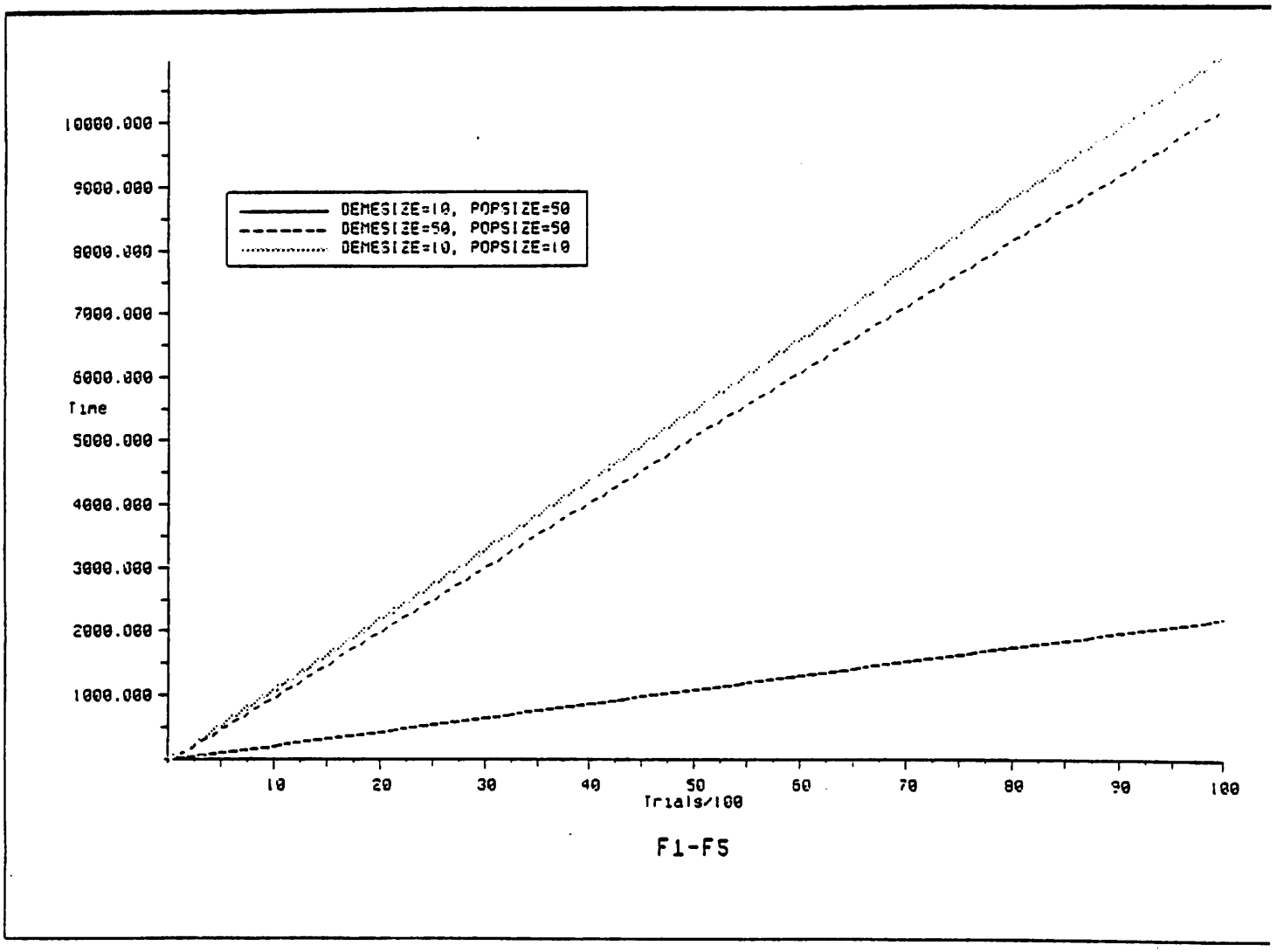
F3


100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
Online
10
20
30
40
50
60
70
80
90
100
Trials/100
DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10

F4


200.00
150.0
100.00
50.0
Online
DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10
10
20
30
40
50
60
70
80
90
100
Trials/100

F5


10000.000
9000.000
8000.000
7000.000
6000.000
Time
5000.000
4000.000
3000.000
2000.000
1000.000
DEMESIZE=10, POPSIZE=50
DEMESIZE=50, POPSIZE=50
DEMESIZE=10, POPSIZE=10
10
20
30
40
50
60
70
80
90
100
Trials/100

F1-F5

Acknowledgments

This master's project was supported by a grant from DEC, and my advisor's GE faculty fellowship (Paul Utgoff). The readers for the project were Paul Utgoff, and Dave Barrington. They provided many hours of discussion, as well as careful reviews of all versions of this report. Jack Stankovic lead me to a working simulation of the DEME model. Bruce Levin from the zoology department, here at UMASS, took me into the genetics world for a semester. David Goldberg from the Univ. of Al, Tuscaloosa, reviewed an early draft of the paper. Insightful comments were added by various students, including Jay Corbett, Martin Herbordt, Sharad Saxena, Jamie Callan, Tom B. Fawcett, and Keith Decker. Lastly, my wife, Barbara Coleman, reviewed all versions of the paper, discussed the material at length, and was helpful in designing the various programs that were needed to complete the project.

References

[Baker, 1987] Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms.* (pp. 14-21). Lawrence ErlBaum Associates, Publishers.

[Bethke, 1976] Bethke, A. D. (1976). *Comparison of genetic algorithms and gradient-based optimizers on parallel processors.* Technical Report No. 212. Ann Arbor: University of Michigan, Logic of Computers Group.

[Cohoon et al, 1987] Cohoon, J. P., Hegde S. U., Martin W. N., Richards, D. (1987). Punctuated equilibria: a parallel genetic algorithm. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms.* (pp. 148-154). Lawrence ErlBaum Associates, Publishers.

[De Jong, 1975] De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems. Doctoral dissertation, University of Michigan. University Microfilms No. 76-9381.

[Goldberg, 1988] Goldberg, D. E. (1988). *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley Publishing Company, Inc.

[Goldberg and Richardson, 1987] Goldberg, D. E., Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms.* (pp. 148-154). Lawrence ErlBaum Associates, Publishers.

[Goldberg and Smith, 1987] Goldberg, D. E., Smith, R. E. (1987). Nonstationary function optimization using genetic algorithms with dominance and diploidy. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms.* (pp. 148-154). Lawrence ErlBaum Associates, Publishers.

[Grefenstette, 1981] Grefenstette, J. J. (1981). *Parallel adaptive algorithms for function optimization.* Technical Report No. CS-81-19. Nashville, TE: Vanderbilt University, Computer Science Department.

[Hartl, 1988] Hartl, D. L. (1988). *A primer of population genetics.* Sunderland, MA: Sinauer Associates, Inc.

[Holland, 1975] Holland, J. H. (1975). *Adaptation in natural and aritificial systems.* Ann Arbor, MI: The University of Michigan Press.

[Lewontin, 1974] Lewontin, R. C. (1974) *The genetic basis of evolutionary change.* Columbia University Press.

[Lewontin, 1984] Lewontin, R. C. (1984). Adaptation. In *Conceptual issues in Evolutionary Biology.* (pp. 234-251), ed. Sober, E. The MIT Press.

[Perry, 1984] Perry, Z. A. (1984). Experimental study of speciation in ecological niche theory using genetic algorithms. Doctoral Dissertation, University of Michigan. University Microfilms No. 8502912.

[White, 1978] White, M. J. D. (1978). *Modes of Speciation.* San Francisco CA: W. H. Freeman and Company.