Chapter 1

FORMAL ADAPTIVE SYSTEMS

1.1 Introduction

The adjective "adaptive" is frequently encountered in the highly scientific and technological age in which we live. We read of sophisticated radar guidance systems which are capable of adapting quickly to changes in terrain and thus permit high-speed low-altitude flying. The space program has focused our attention on the need for machines which are flexible enough to adapt their responses to unexpected environmental factors. Artificial intelligence research has generated complex game-playing computer programs which have learned by experience to play better than their authors. Biologists continue to study the fascinating adaptive capabilities of organisms as simple as bacteria and as complex as man.

The question as to what constitutes an adaptive system has been widely debated, most recently in Tsypkin's survey of control theory (1971). This debate will not be continued here; rather, a broad view of what constitutes an adaptive system will be adopted, a view succinctly stated by Tsypkin (1971, p. 45):

> ... the most characteristic feature of adaptation is an accumulation and slow usage of the current information to eliminate the uncertainty due to insufficient a priori information and for the purpose of optimizing a certain selected performance index.

Tsypkin has focused his attention on artificial adaptive systems used in control theory. Holland (1975), on the other hand, has been studying the characteristics of both natural and artificial adaptive systems from this broad viewpoint. Out of this work has come a formal framework for describing, analyzing, and comparing adaptive systems. This framework is the basis for the formal definition of adaptive systems used in this thesis. However, before presenting the formalism, let us consider some examples of problems which are candidates for an adaptive solution.

1.2 Some Problems for Adaptation

I am particularly interested in the application of adaptive system theory to the problem of adaptive software design. This bias will show itself in the choice of examples and applications discussed in this thesis. The reader is reminded that the adaptive system theory presented here is limited in its application only by the imagination, and he is encouraged to consider examples from his own experience.

1.2.1 Data Structure Design

Suppose we are faced with designing a data structure for a generalized information-retrieval system. If it is really intended to be general purpose, the characteristics of input data sets are unknown at design time. A standard approach is to assume random input and choose the data structure which minimizes some performance criterion for a standard set of data structure operations (e.g. search, delete, insert). Unfortunately, many applications consist of distinctly non-random input resulting in sub-optimal performance. An adaptive approach would explore the possibility of deferring the choice of a specific data structure until the characteristics of a particular data set are available in order to enhance on-line performance.

1.2.2 Algorithm Design

Suppose we are faced with designing a sophisticated time-sharing system which supports a large number of batch and terminal users simultaneously. The heart of such a system is a supervisor program which is responsible for sharing limited system resources among competing processes. The performance of a time-sharing system (usually specified in terms of terminal response time, batch throughput, and system overhead) is directly affected both by the algorithms chosen for resource sharing and by the demand characteristics for system resources. Unfortunately, the demand characteristics can vary widely from day to day and are often difficult to predict. A standard approach is to base resource sharing algorithms on average demand characteristics and hence obtain good performance "on the average". An adaptive approach would explore the possibility of modifying resource sharing algorithms in response to current demand characteristics (see, for example, Bauer (1974)).

1.2.3 Game-playing Programs

Some of the most fascinating aspects of software design have arisen in the area of game-playing programs. Credible systems have been developed for playing games as complex as checkers and chess. The difficulty in designing such programs lies in our inability to specify a winning strategy in a precise algorithmic way. The standard approach has been to specify as precisely as possible the strategies used by expert players. Unspecified parameters (of which there are many) are externally "tuned" during development by observing their effects on performance. This approach has led to the development of several good chess-playing programs. An adaptive approach would explore not only the possibility of self-tuning programs, but also the possibility of strategy-generating systems.

1.2.4 Two-armed Bandits

Two-armed bandit problems arise in the context of statistical decision theory, but have considerable bearing on the problem of adaptation. In its simplest form, the problem is stated as follows: you are presented with two slot-machines, one of which pays better than the other. If you are unaware of which is the better-paying machine, what strategy would you use to minimize your expected losses over N trials? The optimal (but alas, non-realizable) strategy is to play the better-paying machine all the time. Lacking this a priori information, the problem becomes one of minimizing the expected number of trials to the lower-paying machine. Each trial yields more information about the relative performances of the two machines. The goal is to exploit this information as quickly and efficiently as possible. It should be clear by now that two-armed bandit problems capture adaptation in its simplest form: the dynamic gathering and exploitation of information to reduce uncertainty and improve performance.

1.3 A Formal Framework

With these examples in mind, we now ask what are the essential characteristics of adaptation. It has already been suggested that a problem in adaptation arises out of a lack of a priori information which prevents one from choosing between competing alternative solutions to the problem. Implicit in the idea of competing solutions is a measure of performance used to compare alternative solutions. The performance of a solution is a function both of its own characteristics and the particular environment in which it is tested. Adaptation consists of a strategy for generating better-performing solutions to the problem by reducing the initial uncertainty about the environment via feedback information made available during the evaluation of particular solutions.

Holland has been studying the properties of both natural and artificial systems. Out of these studies has come a formalism for representing problems in adaptation which will be used in this thesis. Briefly, a problem in adaptation is formally represented as:

E: the set of environments to be faced.

A: a set of structures describing alternative solutions to the problem.

U: a performance measure for evaluating solutions in a particular environment, i.e.

$U: A \times E \rightarrow R$ (R representing the real line)

I: a feedback function providing dynamic information to the adaptive system about the performance of a particular solution in a particular environment, i.e.

$I: A \times E \rightarrow R^n$

S: the collection of adaptive strategies under study. Each $s \in S$ is a strategy for generating better-performing solutions based on feedback information from previous trial solutions, i.e.

$$s: \left[\left\{ (A(t).I(t)) \right\}_{t=1}^{T} \right] \rightarrow A$$

X: the criterion used for comparing the performances of adaptive strategies, i.e.

$X: S \rightarrow R$

As an example of the formalism, consider how one might formally represent the previously discussed problem of choosing data structures for an information retrieval system:

E: the set of all possible input data sets.

A: the set of alternative data structures.

U: the performance of the information-retrieval system on a particular data set.

I: data structure performance statistics (e.g. search, insert, delete timings).

S: alternative strategies for changing data structures based on input data set characteristics.

X: usually U averaged over random samples from E.

1.4 The Problem of Function Optimization

In this section we will consider the close relationship between the problems of adaptation and function optimization. Function optimization is a well-studied problem in applied mathematics and is briefly stated as follows: given a function $f: A \rightarrow R$, find those points in A on which f takes its maximum (minimum) values. To see its relationship to the problem of adaptation, consider again the formalism discussed above. The performance measure $U: A \times E \rightarrow R$ is more precisely the composition of two functions, a behavioral function $B: A \times E \rightarrow R^n$ specifying the behavioral characteristics of a particular solution in a particular environment, and a metric function $M: R^n \rightarrow R$ specifying the performance rating associated with behavioral characteristics. We can further emphasize the role of the environment by considering a family of behavioral functions $\{b_e\}_{e \in E}$, where each b_e is simply the restriction of B to A x $\{e\}$. In this way adaptation can be viewed as attempting to optimize the performance measure $u_e : A \rightarrow R$ associated with a particular environment $e \in E$ and defined by $u_e(a) = M(b_e(a))$. The difficulty of the problem of adaptation (i.e. the initial uncertainty) can then be expressed in terms of the richness of the set $\{u_e\}_{e \in E}$ of performance measures. Because of this close relationship between the two problems it is worth considering the applicability of function optimization theory to the problem of adaptive system design.

Function optimization theory is generally divided into two areas: constrained and unconstrained problems. The tractability of a constrained problem is often highly dependent on the complexity of the constraints; finding the maximum is often eclipsed by the problem of staying within the constraints. From an adaptive systems point of view, the problem of constraints can be subsumed in the definition of the representation space A and the performance measures $\{u_e\}$. For example, a complexly constrained space H can be embedded in a simply constrained space A with u_e defined to take on its minimal value on A-H. For these reasons we will restrict our attention to unconstrained problems which, as far as any implementation is concerned, are really linearly constrained problems where the constraints are of the form

$$l(i) \leq x(i) \leq h(i), \; i = 1, \ldots, n.$$

A second observation restricts our attention even further. It is the case that the performance measure u_e is almost never available in analytic form. Recall from above that u_e is really the composition of b_e and M. While M is often explicitly expressed in analytic form, b_e, the behavior function, is generally only a "black box" representing the complexity of the problem under adaptation. This observation immediately rules out classical analytic techniques and those iterative techniques which depend on exact expressions for first and possibly second order partial derivatives.

A third observation, and perhaps the most critical as far as the applicability of function optimization theory is concerned, is the fact that for problems of any complexity the behavioral function b_e (and hence in general u_e) is a high-dimensional, non-linear, multimodal function. As a consequence standard optimization techniques which assume linearity or unimodality, or techniques whose computation time grows rapidly with dimensionality are generally inapplicable to the adaptation problem.

With these constraints we are left with only a few alternative optimization techniques. The most commonly proposed search technique for multimodal functions is to run one's favorite local (unimodal) optimizer repeatedly using random starting points, the assumption being that each local maximum will be encountered after a sufficient number of trials. Alternate approaches perform some type of patterned search over the whole space looking for likely areas in which the local optimizer should be employed. Finally, for problems of high dimensionality, several authors (see, for example, Rastrigin (1963) or Schumer & Steiglitz (1968)) have recommended reverting to various forms of random search.

Whether or not these techniques produce the kind of adaptive performance we would like is at this point an open question which will be explored further in this thesis. Comparisons of function optimizers center around the number of function evaluations required to find the optimum within a certain tolerance. The emphasis here is on convergence. In contrast, adaptation is also concerned with the quality of interim performance, the criterion often involving the integral of the performance curve.

1.5 A Reduction in Scope

Having stated and explored the general framework for problems in adaptation, we will now focus our attention on a specific class of adaptive systems which will be the object of this study.

In the first place, we will consider only discrete time-scale adaptive systems. A time step generally consists of generating, testing, and receiving feedback about a particular solution to the problem. The interpretation of a time step is, of course, application-dependent.

Secondly, we will be concerned with the design of adaptive systems in which the only available feedback is the value of the performance measure u_e. Such systems are usually termed "first-order" feedback systems in the sense that the very minimal feedback information available about the behavior of a particular solution is its performance rating.

Finally, we will restrict our attention to two adaptive system performance criteria (X and X^* defined below). The motivation for these criteria arises from the concept of robustness. We say that an adaptive system is robust if it is able to generate and maintain acceptable solutions to a problem across a wide variety of environments. In order to formalize this concept, consider first the definition of local robustness, i.e. the ability of a strategy to generate and maintain acceptable solutions to a problem in a particular environment. Two such measures will be used in this thesis: local on-line performance and local off-line performance. On-line performance $x_e : S \rightarrow R$ will be defined as follows:

$$x_e(s) = \frac{1}{\sum_{t=1}^{T_e} c_t} * \sum_{t=1}^{T_e} c_t \cdot u_e(a_t)$$

That is, the performance of strategy s in environment e is a weighted average of the performances $u_e(a_t)$ of the generated solutions a_t over a time period T_e. On-line performance measures are motivated by situations in which adaptive systems are being used to dynamically improve the overall performance of an on-line system such as a time-sharing system. In such situations every new solution generated by the adaptive system for testing is included in the overall performance rating of the system.

In contrast, local off-line performance $x^* : S \rightarrow R$ will be defined as:

$$x_e^*(s) = \frac{1}{\sum_{t=1}^{T_e} c_t} * \sum_{t=1}^{T_e} c_t \cdot u_e^*(a_t)$$

where $u_e^*(a_t) \overset{d}{=} \min \left\{ u_e(a_1), \ldots, u_e(a_t) \right\}$. Off-line performance is motivated by situations in which the testing and evaluation of solutions is done off-line and is not included in the overall performance evaluation. In these situations the on-line system runs with the best solution generated to that point while off-line adaptation is continuing. Off-line performance is much closer to the standard measure of performance for function optimizers. The magnitude of trial errors is not included; only progress toward the minimum is measured. As a consequence, off-line performance places heavier emphasis on convergence while on-line performance emphasizes initial performance.

In both cases, the weights c_t provide a means of shifting the emphasis. If they are increasing ($c_t < c_{t+1}$), more emphasis is placed on convergence. If they are decreasing ($c_t > c_{t+1}$), more emphasis is placed on initial performance. For our purposes $c_t=1$ for all t is sufficient.

Global robustness is now defined in terms of these local measures. On-line performance X: S → R is given by:

$$X(s) = \frac{1}{\sum_{E} w_e} * \sum_{E} w_e \cdot x_e(s)$$

Off-line performance is similarly given by:

$$X^*(s) = \frac{1}{\sum_{E} w_e} * \sum_{E} w_e \cdot x_e^*(s)$$

In both cases, the weights w_e can be used to assign relative difficulties to the alternative environments.

1.6 Summary

In this chapter we have attempted to define formally what we mean by a problem in adaptation and have discussed some practical examples of such problems. We have noted the close relationship between the problems of adaptation and function optimization, and we have seen that the bulk of optimization techniques is not generally applicable to the design of adaptive systems. Finally, we have defined the specific class of adaptive systems which will be the subject of further study in the following chapters.

Chapter 2

GENETIC ADAPTIVE MODELS

2.1 Introduction

In the discussion of function optimization theory in chapter 1, we noted that, although the problems of adaptation and optimization are closely related, most of the standard optimization techniques are inadequate for adaptive problems of any complexity. This inadequacy can be viewed as an inability to process information relating to global aspects of the function to be optimized. Extremely efficient techniques have been developed for finding the nearest local maximum of a function; however, attempts to extend these techniques to find global maxima have met with little success. Some global search techniques have been proposed for low-dimensional problems (see, for example, Hill (1969) or Bremermann (1970)), their computation time growing rapidly with dimensionality. As a consequence, most global searching is accomplished with some form of random search. From an adaptive system point of view, random search is extremely inefficient because it makes no use of the available feedback information to reduce the initial uncertainty surrounding the problem for adaptation. These observations suggest a critical question for adaptive system design: are there efficient ways to exploit global information about a problem in order to generate better-performing solutions?

This thesis is part of a larger research project which is attempting to answer such questions under the direction of John Holland at the University of Michigan. The basic point of view of this research is that nature is an extremely rich source of examples of sophisticated information processing and adaptation. The goal of this project is to understand and abstract from natural systems the mechanisms of adaptation in order to design artificial systems of comparable sophistication. This research has centered around the design of artificial systems derived from standard models of heredity and evolution in the field of population genetics which we will briefly review.

2.2 Genetic Population Models

Population genetics is concerned with the characteristics of heredity and evolution at the population level. It assumes a Mendelian view of the mechanisms of heredity, i.e. genetic material is represented as strands of chromosomes consisting of genes which control observable properties in the individuals making up the population. A population is viewed as a dynamic pool of genetic information, the characteristics of which change from generation to generation in response to environmental factors. Numerous examples exist which demonstrate the ability of a population of organisms to adapt over a period of generations to complex changes in its environment. The goal is to explain these observable adaptations in terms of the mechanisms of heredity and evolution.

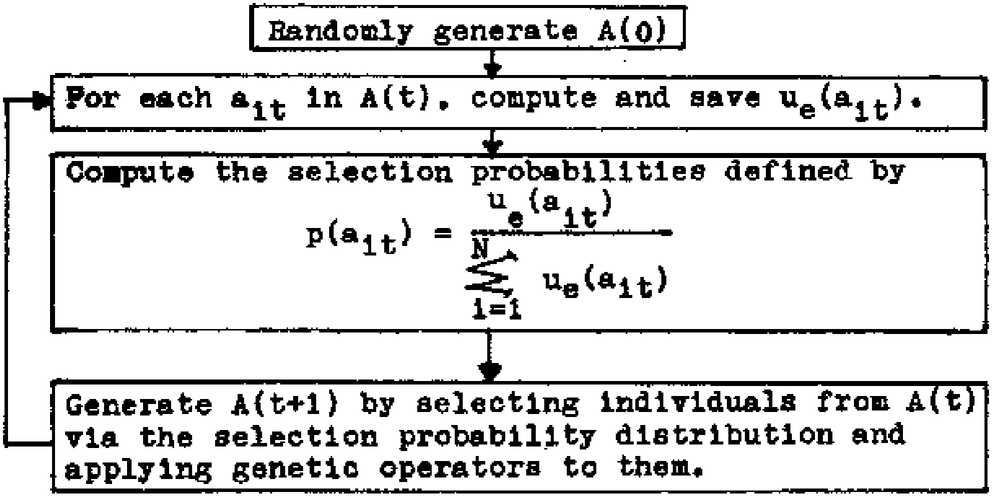
In a genetic population model, individuals are represented purely in terms of their genetic makeup. Representations of genetic material vary from simple one-chromosome individuals (haploid models) to complex multi-chromosome individuals (polyploid models). Having specified a representation for genetic material, the observable characteristics of an individual are defined as functions of the chromosomal genes. Environmental pressures, specified in terms of these observable characteristics, assign a measure of "fitness" to an individual. Finally, the dynamics of population development are defined in terms of fitness, life-death cycles, mating rules, mobility, sex, species, and so on.

We, of course, are not concerned with modeling the development of biological populations per se; rather, we are concerned with understanding the mechanisms of adaptation which provide for such development. Unfettered by biological facts, we are free to construct artificial systems which capture the essence of these mechanisms. The exciting aspect of this approach, as we will see, is that even very simple artificial systems exhibit considerable adaptive capabilities.

2.3 Reproductive Plans

In this section we will describe the basic class of artificial adaptive systems which has arisen from the genetic population models. This class of adaptive systems, called reproductive plans, was first proposed by Holland; subsequent variations have been studied by others (see, for example, Caviccio (1970), Hollstien (1971), Frantz (1972)).

Recall from the formalism introduced in chapter 1 that alternative solutions to the problem for adaptation are represented by the set A. In a reproductive plan, the memory of the system at time t consists of a population A(t) of N individuals a_{it} from A together with their associated performance ratings $u_e(a_{it})$. These representations a_{it} of solutions to the problem for adaptation are considered the genetic material to be processed by a reproductive plan. New individuals (and hence new alternative solutions) are produced by simulating genetic population dynamics. That is, individuals from A(t) are selected as parents and idealized genetic operators are applied to produce offspring. More specifically, a reproductive plan operates as follows:



Randomly generate A(0)

For each a_{it} in A(t), compute and save $u_e(a_{it})$.

Compute the selection probabilities defined by

$$p(a_{it}) = \frac{u_e(a_{it})}{\sum_{i=1}^{N} u_e(a_{it})}$$

Generate A(t+1) by selecting individuals from A(t) via the selection probability distribution and applying genetic operators to them.

To get a feeling for how reproductive plans work, note that the expected number of offspring produced by an individual is proportional to its performance. This can be seen by considering the process of selecting individuals for reproduction as N samples from A(t) with replacement using the selection probability distribution. Hence, the expected number of offspring from individual a_{it} is given by

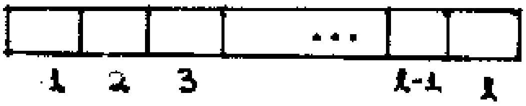
$$\begin{aligned} O(a_{it}) &= N \cdot P(a_{it}) \\ &= N \cdot \frac{u_e(a_{it})}{\sum_{i=1}^{N} u_e(a_{it})} \\ &= \frac{u_e(a_{it})}{\frac{1}{N} \sum u_e(a_{it})} \\ &= \frac{u_e(a_{it})}{\overline{u_e}(A(t))} \end{aligned}$$

So we see that individuals with average performance ratings produce on the average 1 offspring while better individuals produce more than 1 and poorer individuals produce less than 1. Hence, with no other mechanisms for adaptation, reproduction proportional to fitness produces a sequence of generations A(t) in which the best individual in A(o) takes over a larger and larger proportion of the population.

However, in nature and in these artificial systems, offspring are almost never exact duplicates of a parent. It is the role of genetic operators to exploit this selection process by producing new individuals which have high-performance expectations. The choice of operators is motivated by the mechanisms of nature: crossover, mutation, inversion, and so on. The exact form taken by such operators depends on the "genetic" representation chosen for individuals in A. In order to see more clearly the role of genetic operators, let us consider a very simple (from a biological viewpoint) reproductive plan which exhibits surprising adaptive capabilities.

2.4 The Basic Reproductive Plan: R1

The simplest reproductive plans use fixed-length haploid representations for elements of A. That is, an individual is represented by a single chromosome consisting of a fixed number (ℓ) of genes:



Each gene position is defined to take on one of a specified number of (allele) values. Hence, the set A of all possible individuals can be considered an ℓ-dimensional space in which an individual is represented by the value of its genes. To obtain a representation of this form for a specific problem for adaptation, alternative solutions to the problem are characterized uniquely by an ordered set of ℓ parameters which in turn play the role of genes.

Having thus defined the representation space A, a reproductive plan is now free to explore A by submitting individuals for testing and evaluation as solutions to the problem. The basic reproductive plan accomplishes this via two genetic operators: crossover and mutation. To specify precisely how these operators work, we let an element a_i from A be represented as the string $v_{i1}v_{i2} \cdots v_{i\ell}$ in which the v_{ij} represent the gene values (alleles).

Crossover generates a new individual a_k from two existing individuals a_i and a_j by concatenating an initial gene segment from a_i with a final gene segment from a_j. The segments are defined by selecting a crossover point via a random sample from a uniform distribution over the ℓ-1 positions between the genes. So, for example, if crossover occurs between the second and third gene positions, individual a_k is generated from a_i and a_j as illustrated:

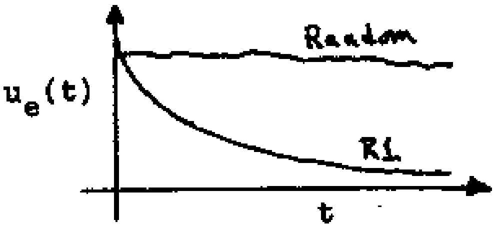
$$\begin{array}{l} a_i = \underline{v_{i1}v_{i2}}v_{i3} \cdots v_{i\ell} \\ a_j = v_{j1}v_{j2}\underline{v_{j3} \cdots v_{j\ell}} \end{array} \rightarrow a_k = v_{i1}v_{i2}v_{j3} \cdots v_{j\ell}$$

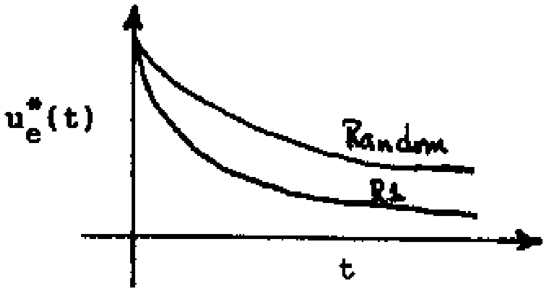
The crossover operation is embedded in plan R1 in the following way. Given an individual a_{it} selected from A(t) to produce an offspring, a mate a_{jt} is chosen from A(t) using the selection probabilities. An offspring is then produced by crossover.

So we see that the strategy employed by crossover in searching A for better-performing solutions consists of constructing new sample points from existing ones selected on the basis of performance. Notice that if a particular allele (gene value) v_{ij} is not present in A(t), no off-spring produced by crossover will contain v_{ij}. In other words, crossover is unable to generate points in the sub-space $V_1 \times V_2 \times \ldots \times \left\{v_{ij}\right\} \times \ldots \times V_\ell$ of A. An allele can be missing from A(t) for several reasons. It may have been deleted by selection because of associated poor performance. It may also be missing simply because of the limited size of A(t). Obviously, if $|V_i| = 1000$, a minimum population of size 1000 is required for A(t) to contain an instance of each v_{ij}. In plan R1 new alleles are introduced into A(t) by the second genetic operator: mutation.

Mutation generates a new individual by independently modifying the value of one or more genes of an existing individual. A gene is selected for modification via a random sample from a uniform distribution over the ℓ gene positions. The new gene value is selected via a random sample from a uniform distribution over the associated set of alleles V_j. So, for example, if individual a_i is selected to undergo a mutation at position 2, an individual $a_j = v_{i1}v_{\underline{j2}}v_{i3}\ldots v_{i\ell}$ is generated. The mutation operator is embedded in plan R1 as follows: a small percentage of individuals generated by crossover for A(t+1) additionally undergo a mutation. In nature the probability of a gene undergoing mutation is generally less than .001 indicating that mutation (a form of random search) is not the primary genetic operator. Rather, it should be viewed as a background operator guaranteeing no allele will permanently disappear from A(t).

In order to evaluate the adaptive capabilities of plan R1, an environment E was defined consisting of a broad class of performance measures u_e defined on A (see appendix A). Included were instances of continuous, discontinuous, convex, non-convex, unimodal, multimodal, low-dimensional and high-dimensional functions as well as functions with Gaussian noise. The plan R1 was implemented in PL1 and its behavior observed over E in comparison to pure random search (see appendix C). While R1 did not always converge to a global maximum in the time allotted, it exhibited a considerable improvement over the performance generated by random search. Typical curves from these simulations are shown below:





Recall that the performance criteria X and X* for adaptive systems were defined in terms of the average values of u_e and u_e^*, respectively, over time. With these encouraging results, we consider in more detail the properties of plan R1.

2.5 K-armed Bandits

Before we explore in more detail the way in which plan R1 searches the space A for better-performing elements, we will take a brief, but relevant, diversion to consider solutions to the generalization of the 2-armed bandit problem introduced in section 1.2, namely, the optimal allocation of trials to K machines. Holland (1975) has shown a mathematical solution exists if one is given a bit more a priori information about the K machines. Suppose we know that each machine pays stochastically according to a normal distribution $N(u_i,s_i^2)$, but we are not told which distribution is associated with which machine. In this case, an optimal strategy for allocating T trials to the K machines is roughly characterized as follows:

allocate exponentially more of the T trials to the observed best than to the remaining K-1 machines, where the exact form of the exponential depends on the K distributions $N(u_i,s_i^2)$. Notice that this strategy is non-realizeable in that no strategy can decide which machine will be the observed best after T trials without allocating the T trials, and then it is too late to distribute the trials optimally. However, such a solution gives us a characterization of the way in which trials should be allocated, and it yields a lower bound on the expected losses over T trials. The question, of course, is whether there are any realizeable strategies which are good approximations to the optimal one. To answer this question, we consider in more detail the optimal solution to the 2-armed bandit problem.

In this case we have two machines B1 and B2 which pay according to the distributions $N(u_1,s_1^2)$ and $N(u_2,s_2^2)$ respectively. For convenience, let B1 be the machine with the higher payoff and $\widetilde{B1}$ be the machine with the highest observed payoff after all T trials have been allocated with t_1 going to $\widetilde{B1}$ and t_2 to $\widetilde{B2}$. Holland (1975) has shown that the expected loss incurred over these T trials is given by:

$$L(t_1,t_2) = |u_1-u_2| * \left[t_1 * q(t_1,t_2) + t_2 * (1-q(t_1,t_2))\right]$$

where $q(t_1,t_2)$ is the probability that B2 will be the observed best and is well-approximated by

$$q(t_1,t_2) = \frac{1}{\sqrt{2\pi}} * \frac{\exp(-x^2/2}{x} \quad , x = \frac{u_1-u_2}{\sqrt{s_1^2/t_1 + s_2^2/t_2}}$$

To get a feeling for how $L(t_1,t_2)$ varies over the interval $0 < t_2 < T$, consider L rewritten as a function of T:

$$L(T,t_2) = \left|u_1-u_2\right| * \left[(T-t_2)*q(T,t_2) + t_2*(1-q(T,t_2))\right]$$

$$= \left|u_1-u_2\right| * \left[(T-2t_2)*q(T,t_2) + t_2\right]$$

As illustrated in figure 2.1, the term

$$(T-2t_2)*q(T,t_2) \cong (T-2t_2)* \frac{1}{\sqrt{2\pi}} * \frac{\exp(-x^2/2)}{x}$$

dominates L for small values of t_2, but drops off exponentially as a function of t_2 since

$$x = \frac{u_1-u_2}{\sqrt{s_1^2/t_1 + s_2^2/t_2}} \cong \frac{u_1-u_2}{\sqrt{s_2^2/t_2}} = (\frac{u_1-u_2}{s_2})*\sqrt{t_2} = k_2\sqrt{t_2}$$

$$\therefore \quad (T-2t_2)* \frac{1}{\sqrt{2\pi}} * \frac{\exp(-x^2/2)}{x} \cong \frac{T}{\sqrt{2\pi}} * \frac{\exp(-k_2^2t_2/2)}{k_2\sqrt{t_2}}$$

$$= \frac{T}{k_2\sqrt{2\pi}} * \frac{\exp(-a_2t_2)}{\sqrt{t_2}}$$

FIG 2.1: BANDIT LOSS FUNCTION FOR T=50

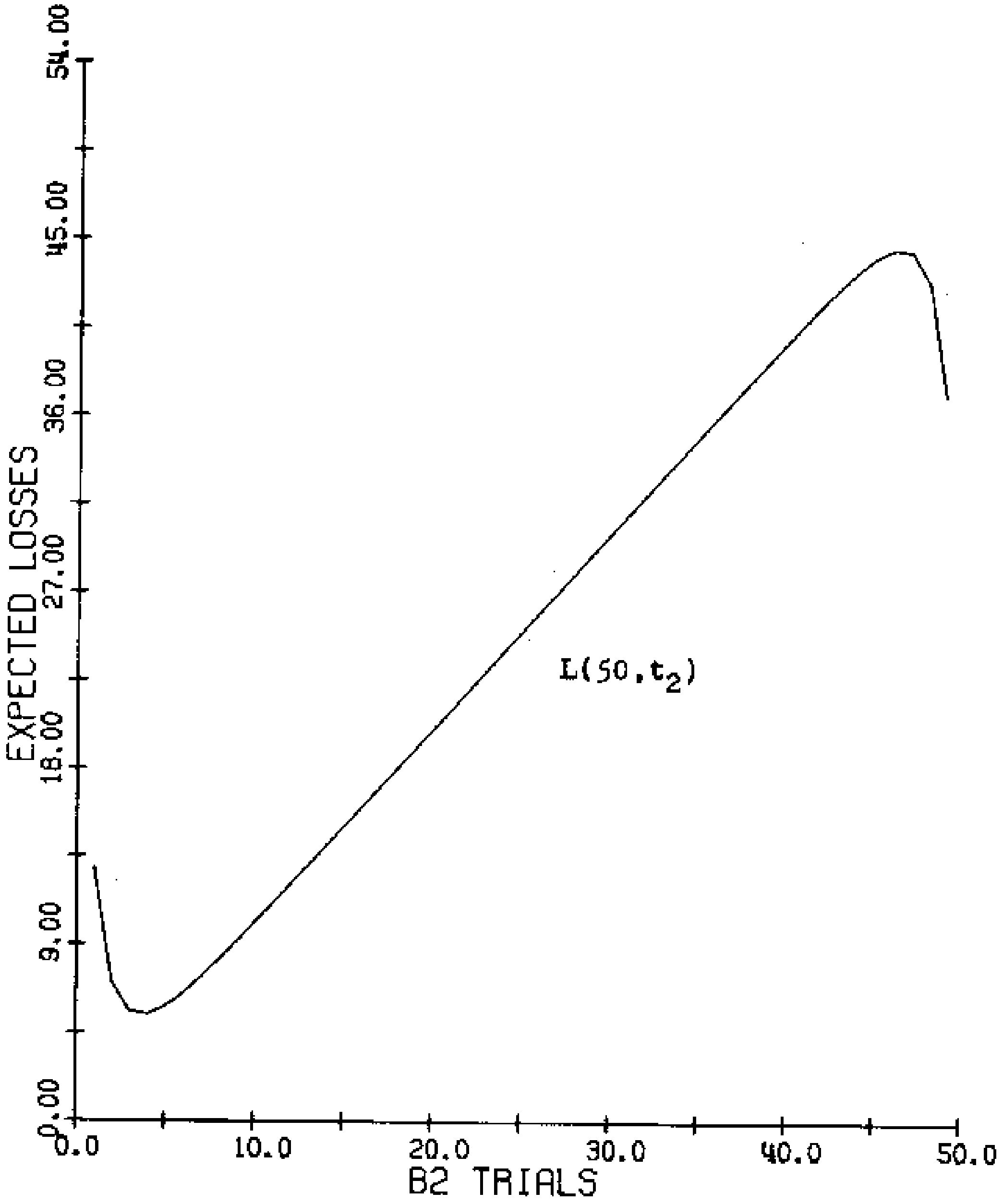


Figure 2.1: Expected losses over 50 trials on bandits B1(9,1) and B2(8,1).

As t_2 increases, the term t_2 dominates and L is essentially linear with respect to t_2. Finally, as $t_2 \rightarrow T$, the term

$$(T-2t_2)*q(T,t_2) \cong \frac{-T}{\sqrt{2\pi}} * \frac{\exp(-k_1^2 t_1/2)}{k_1\sqrt{t_1}} = \frac{-T}{k_1\sqrt{2\pi}} * \frac{\exp(-a_1 t_1)}{\sqrt{t_1}}$$

re-emerges as the dominant term with a negative sign.

In order to minimize our expected losses over T trials, we must find the value t_2^* such that

$$L(T,t_2^*) \leq L(T,t_2) \ , \ 0 < t_2 < T$$

Finding an analytic expression for t_2^* by considering those points at which $\frac{dL}{dt_2} = 0$ is fairly complex. Holland (1975), for example, has derived the approximation

$$t_2^* \sim b^{-2} * \ln\left[\frac{b^4 T^2}{8\pi \cdot \ln(T^2)}\right] , \ b = \frac{u_1-u_2}{s_2}$$

For our purposes the optimum is found via a one-dimensional iterative search technique applied directly to L for various values of T. Figure 2.2 illustrates how the optimal loss function $L(T,t_2^*)$ varies with T. It is this kind of performance that a realizeable strategy must hope to approximate. Finally, figure 2.3 illustrates the previously mentioned relationship between $t_1^* = T-t_2^*$ and t_2^*, namely, that an optimal strategy allocates ex-

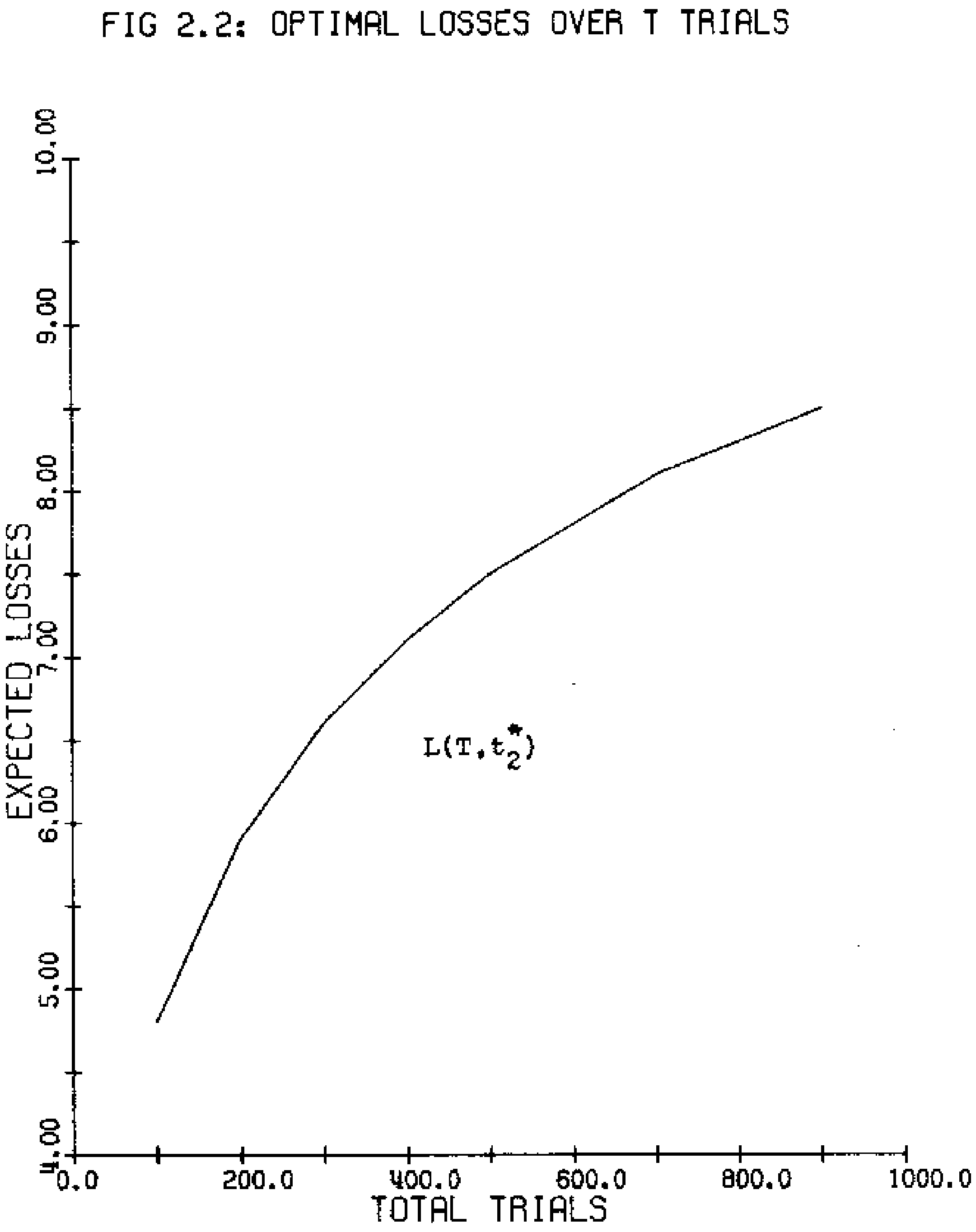


Figure 2.2: Optimal losses incurred over T trials on two bandits B1(9,1) and B2(8,1).

FIG 2.3: OPTIMAL DISTRIBUTION OF T TRIALS

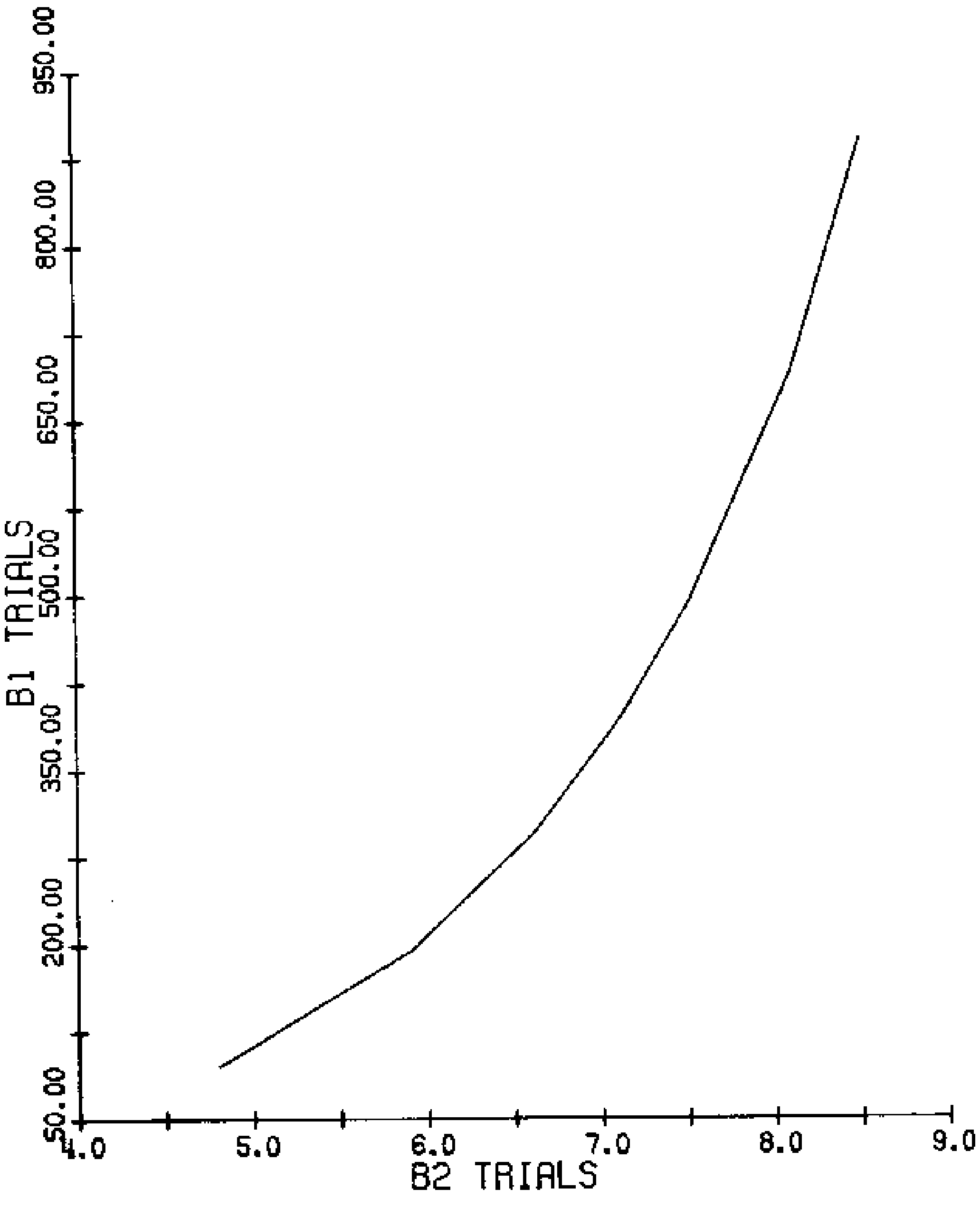


Figure 2.3: Optimal distribution of T trials between two bandits B1(9,1) and B2(8,1).

ponentially more trials to the observed best.

We are now in a position to evaluate the performance of some realizeable strategies. The first one which comes to mind is the standard decision theory approach (hereafter referred to as DTS) which goes as follows: allocate a small number t of trials to each machine; then allocate the remaining T-2t trials to the observed best. Paralleling the preceding analysis, we have an expected loss function:

$$L_1(T,t) = |u_1-u_2| * \left[(T-t)*q(t) + t*(1-q(t))\right]$$

where q(t) is the probability that B2 is the observed best after allocating t trials to each machine. In this case we have

$$q(t) \cong \frac{1}{\sqrt{2\pi}} * \frac{\exp(-x^2/2)}{x}, \quad x = \frac{u_1-u_2}{\sqrt{s_1^2/t + s_2^2/t}}$$

$$= \frac{u_1-u_2}{\sqrt{s_1^2 + s_2^2}} * \sqrt{t}$$

Again, we seek the value t^*, $0 < t^* < \frac{T}{2}$, which minimizes $L_1(T,t)$, as illustrated by figure 2.4. It should be clear that we can define a DTS which, when given T, u_1, u_2, s_1, and s_2, computes the optimal initial sample size t^* and allocates its trials accordingly. Intuitively one feels that this DTS will approximate the optimal strategy as T

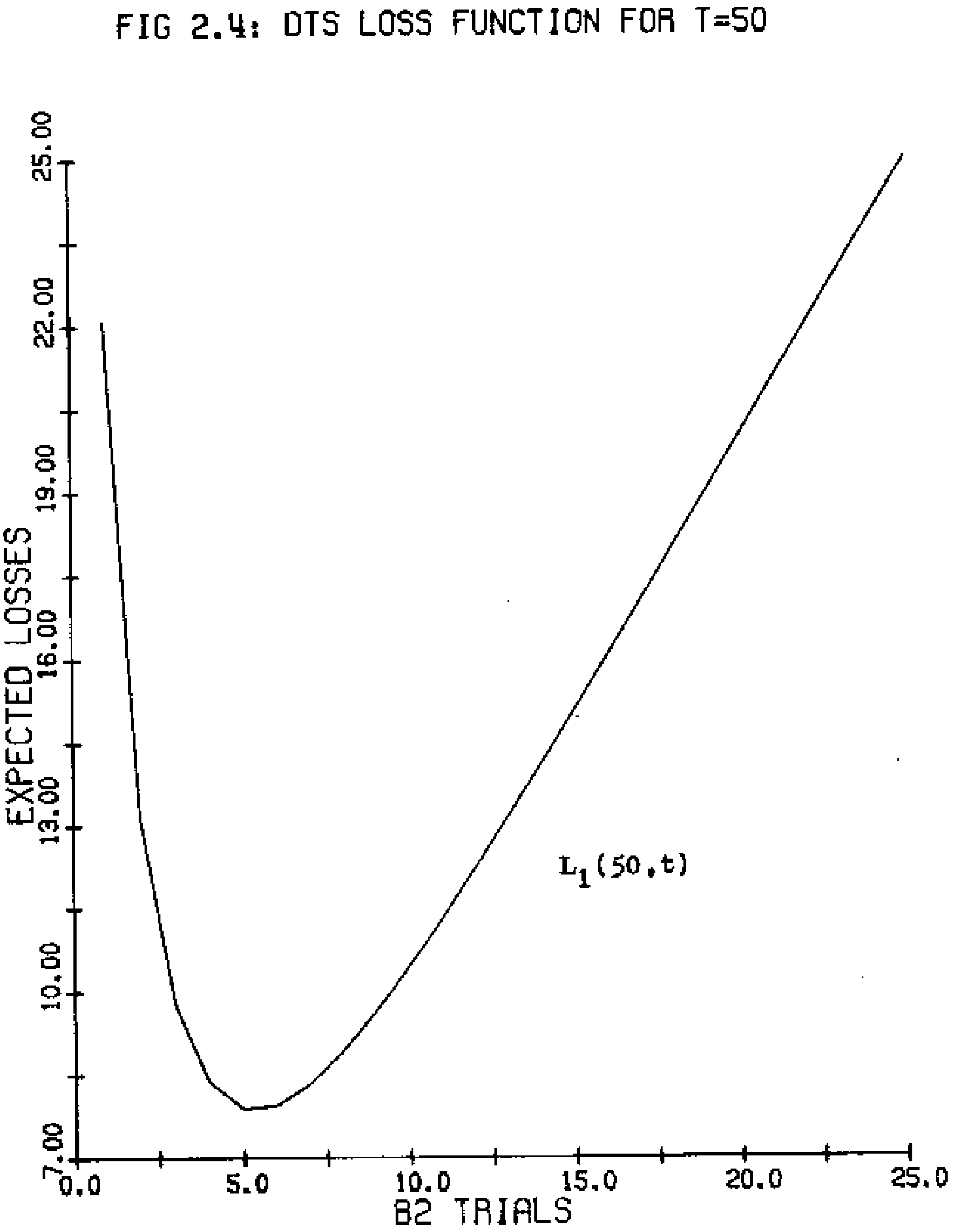


Figure 2.4: DTS expected losses over 50 trials on bandits B1(9,1) and B2(8,1).

increases. Figure 2.5, however, illustrates that it is a fairly crude approximation since the optimal number of trials allocated to B2 grows very slowly with T.

A second more interesting approach incorporates some of the ideas presented in the discussion of reproductive plans in section 2.3. The basic idea is to make a series of reversible decisions during the sequence of trials rather than one non-reversible decision. This is accomplished by defining a selection probability distribution over the machines. Initially, the distribution is uniform; however, it changes over time as follows:

$$P_i(t+1) = P_i(t) * \frac{\overline{f_i}(t)}{\overline{f}(t)} * K_{t+1}$$

That is, the probability of selecting machine i changes over time in proportion to its observed performance relative to the average, where K_{t+1} is the normalization factor required for $\sum_i P_i(t+1)=1$. If at each time step we select a machine for trial by sampling from this time-varying selection distribution, it should be clear that a machine which continues to show above-average performance will rapidly dominate the allocation of trials.

Initially t samples are allocated to each machine for estimates $\overline{f_i}(t)$ of u_i before the first decision is made. This, of course, incurs an initial loss $|u_1-u_2| *t$, but adds certainty to the subsequent decisions. As $t \to 1$, the initial overhead is reduced at the expense of making de-

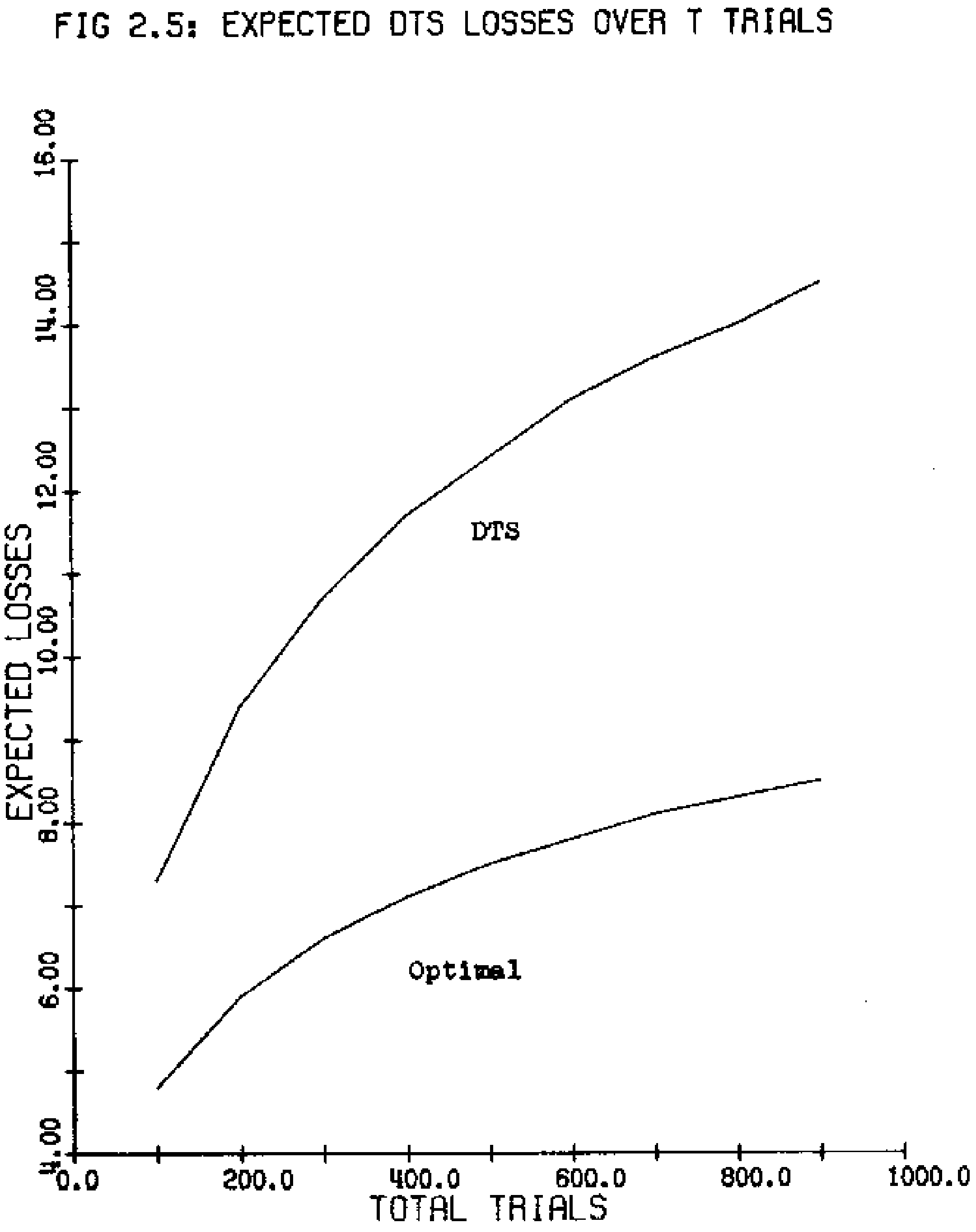


Figure 2.5: A comparison of the expected losses for DTS and the optimal on two bandits B1(9,1) and B2(8,1).

cisions with more uncertainty.

So we have expected losses over T trials given by:

$$L_2(T,t) = \left|u_1-u_2\right| * t + \tilde{L}_2(T,t)$$

where $\tilde{L}_2(T,t)$ specifies the expected losses during the time-varying decision processes from 2T+1 to T. We can express $\tilde{L}_2$ as

$$\tilde{L}_2(T,t) = \sum_{j=2t+1}^{T} \ell(j)$$

where $\ell(j)$ is the expected loss on the j^{th} trial and is given by

$$\ell(j) = \left|u_1-u_2\right| * E\left[P_2(j)\right]$$

where $E\left[P_2(j)\right]$ is the expected value of the selection probability $P_2(j)$ at time j.

While it is relatively straightforward to calculate the expected initial value, $E\left[P_2(2t)\right]$, subsequent expected values are extremely difficult to analyze since the transition function

$$P_2(t+1) = P_2(t) * \frac{\overline{f}_2(t)}{\overline{f}(t)} * K_{t+1}$$

is non-Markovian and depends on the random variable $t_2(t)$, the number of trials allocated to B2 through time t.

Consequently, we are faced with optimizing $L_2(T,t)$ with respect to t by simulation as illustrated in figure 2.6. Two hundred samples were taken of $L_2(100,t)$ for

FIG 2.6: TVS LOSS FUNCTION FOR T=100

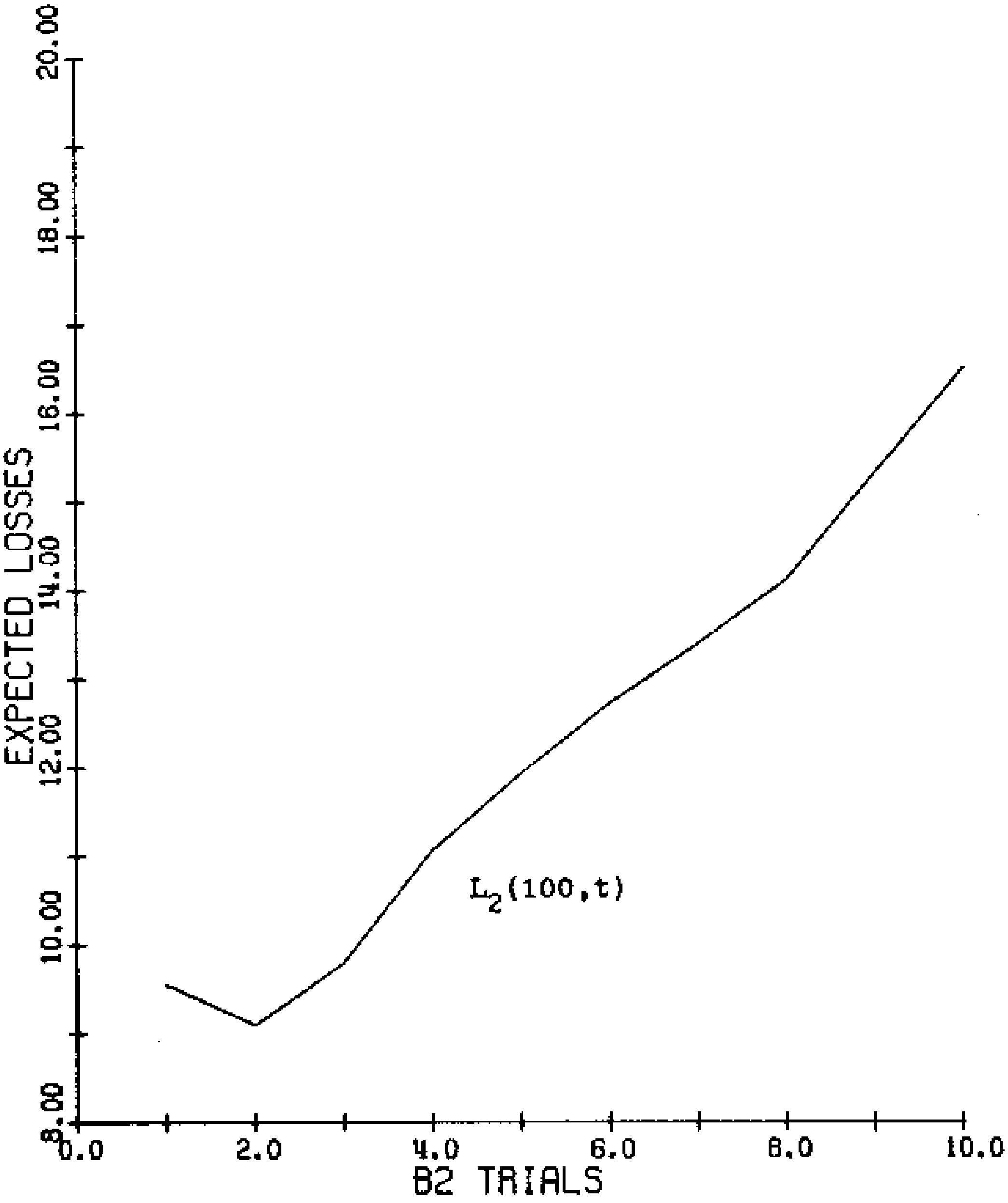


Figure 2.6: Simulated losses over 100 trials using TVS on two bandits B1(9,1) and B2(8,1).

t=1,2,3,...,10. These figures, and others not shown here, suggest that a good approximation for t^* is given by:

$$t^* \geq \frac{\sqrt{s_1^2 + s_2^2}}{u_1-u_2}$$

which, for the illustrated case, yields $t^* \geq \sqrt{2}$ or $t^* \cong 2$. This formulation is motivated as follows: choose enough initial samples t so that, with a priori probability q, $\overline{f_1}(t) - \overline{f_2}(t)$ will have the same sign as u_1-u_2. We know the a priori probabilities associated with $f_1(t) - f_2(t)$ falling in the interval $(u_1-u_2) \pm K* \frac{\sqrt{s_1^2/t + s_2^2/t}}{\sqrt{t}}$ For the signs to be the same, we must have

$$|u_1-u_2| \geq K* \frac{\sqrt{s_1^2/t + s_2^2/t}}{\sqrt{t}}$$

or

$$t \geq K* \frac{\sqrt{s_1^2 + s_2^2}}{|u_1-u_2|}$$

The value K=1 or q=.68 seemed to fit the data best.

Using the above approximation for t^*, figure 2.7 compares the expected TVS losses with those of the two previous strategies and illustrates that it rapidly approaches the optimal one. Finally, figure 2.8 compares the way in which the three strategies divide the trials between the two machines.

With this analysis in mind, we now consider in more

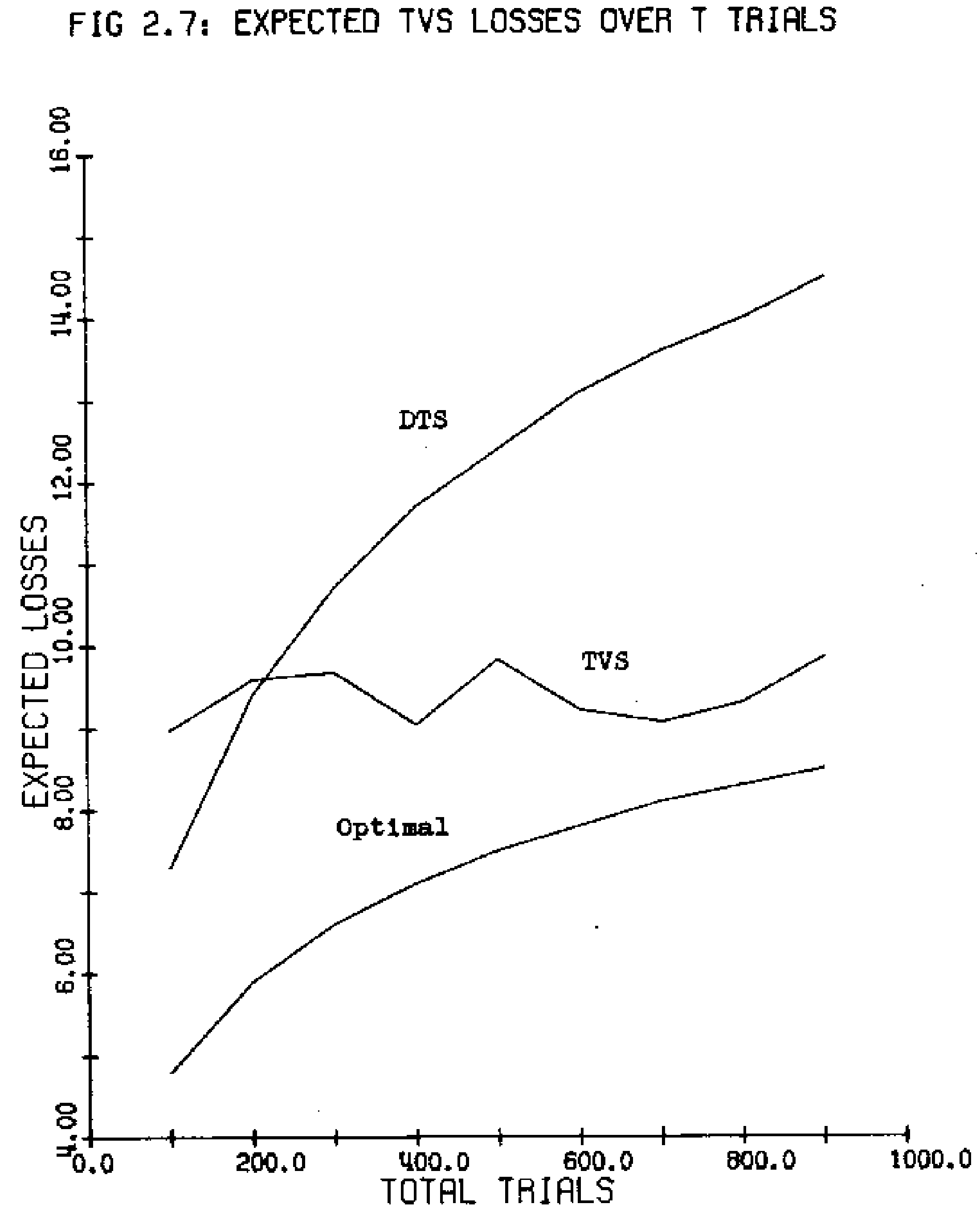


Figure 2.7: A comparison of expected losses over T trials on two bandits B1(9,1) and B2(8,1).

FIG 2.8: COMPARATIVE ALLOCATION OF T TRIALS

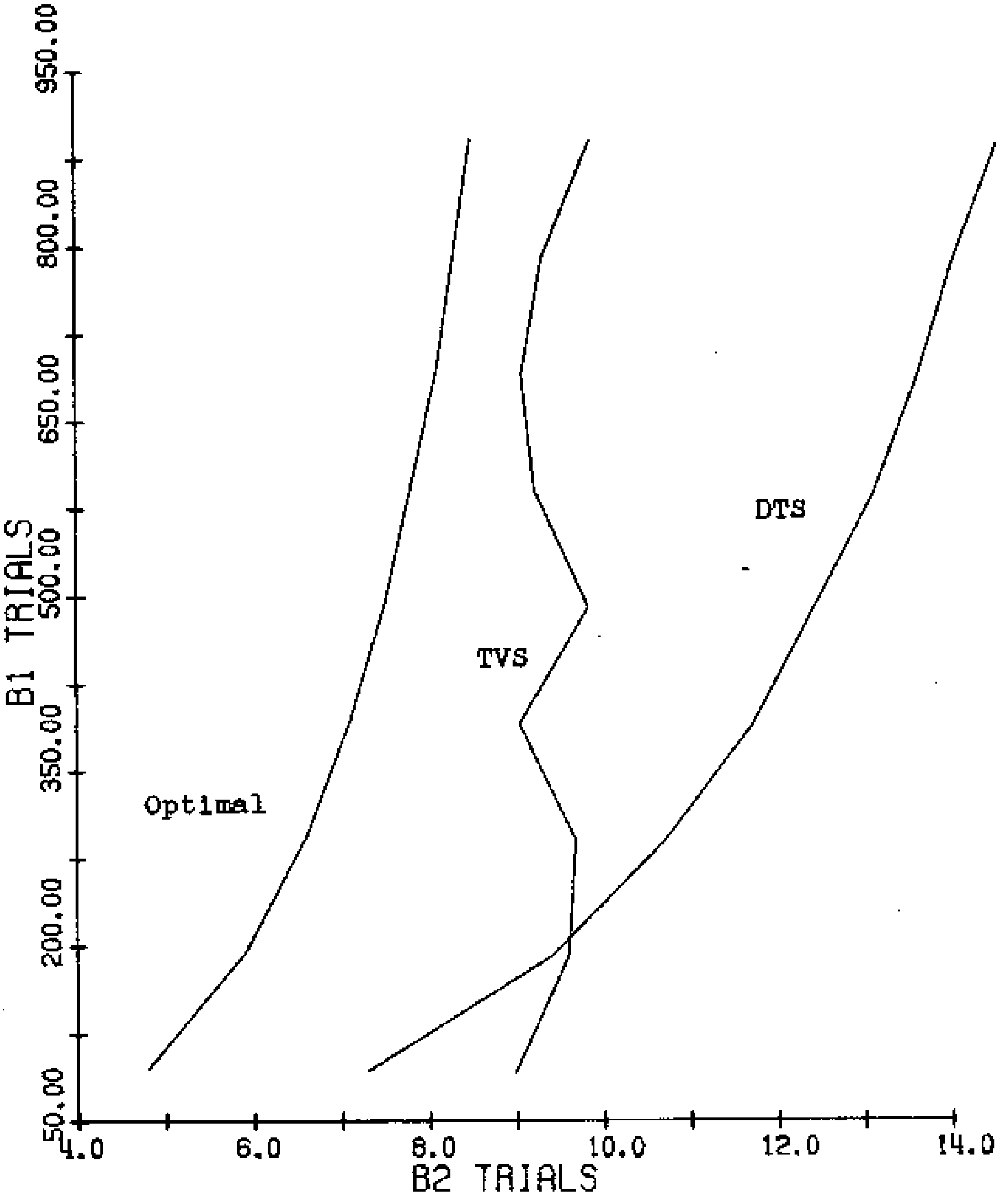


Figure 2.8: A comparison of the allocation of T trials to two bandits B1(9,1) and B2(8,1).

detail how adaptive plan R1 allocates its trials within the space A.

2.6 Hyperplane Analysis of R1

In this section we will attempt to understand more clearly how the genetic plan R1 searches the representation space A for better-performing elements by focusing our attention on hyperplane partitions of A as suggested by Holland (1975).

As suggested in section 2.4, we consider A as an ℓ-dimensional space in which a point $a_i \in A$ is specified by giving its ℓ gene values $\langle v_{i1}, \ldots, v_{i\ell} \rangle$. A kth-order hyperplane is then defined to be the $(\ell-k)$-dimensional subspace of A specified by giving only k of the ℓ gene values. These hyperplanes can be represented visually as follows:

$$0--\ldots- \overset{d}{=} \left\{ a_i \in A \ : \ v_{i1} = 0 \right\}$$

$$-11-\ldots- \overset{d}{=} \left\{ a_i \in A \ : \ v_{i2} = 1 \ \& \ v_{i3} = 1 \right\}$$

If we consider all possible hyperplanes which can be defined by specifying the gene values of a fixed set of K positions, this set $\left\{ H_i \right\}$ of hyperplanes forms a uniform partition of the space A. For example, if $V_1 = \left\{ 0, 1 \right\}$, the allowable values for the first position, then

$$H_1 = 0--\ldots-$$

$$H_2 = 1--\ldots-$$

form a first-order partition of A with exactly half of the points falling in each hyperplane. If we consider the performance measure $u_e : A \to R$ restricted to a particular hyperplane,

$$u_e\Big|_{H_i} : H_i \to R$$

it has a well-defined mean and variance which are, of course, unknown to an adaptive strategy. Hence, associated with each hyperplane partition $\left\{H_i\right\}_{i=1}^{K}$ of the space A, is a K-armed bandit problem, namely, the optimal allocation of trials among the partition elements H_i. Since any sequence of trials in A simultaneously distributes trials among the elements of each of the $\sum_{j=0}^{\ell} \binom{\ell}{j} = 2^{\ell}$ distinct hyperplane partitions of A, we can view the problem of searching A as simultaneously solving 2^{ℓ} K_j-armed bandit problems. The question we are exploring in this section is how well plan R1 allocates its trials to these K_j-armed bandits.

In order to accomplish this we fix our attention on a particular hyperplane partition $\left\{H_i\right\}$ in relationship to the population A(t) of N individuals maintained by plan R1. Since $\left\{H_i\right\}$ is a partition of the space A, each a_{it} in A(t) lies in some H_i. Let $M_i(t)$ represent the number of individuals from A(t) which lie in H_i at time t. Because of the way in which the selection probabilities were defined for reproductive plans (section 2.3), we know that the expected number of offspring $O(H_i)$ produced by individuals in H_i at time t is given by:

$$O(H_i) = \sum_{j=1}^{M_i(t)} \frac{u_e(a_{jt})}{\bar{u}_e(t)}$$

$$= \frac{M_i(t)}{\bar{u}_e(t)} * \frac{\sum_{j=1}^{M_i(t)} u_e(a_{jt})}{M_i(t)}$$

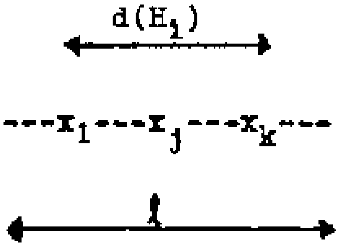
$$= M_i(t) * \frac{\tilde{u}_e(H_i(t))}{\bar{u}_e(t)}$$

If in fact the offspring $O(H_i)$ themselves lie in H_i, then we have

$$M_i(t+1) = M_i(t) * \frac{\tilde{u}_e(H_i(t))}{\bar{u}_e(t)}$$

That is, the number of trials allocated to H_i varies from one time step to the next in proportion to its performance relative to the average, which of course is the TVS solution to the K-armed bandit problem discussed in the preceding section.

Whether or not $O(H_i) \subseteq H_i$ depends on the genetic operators used to construct them. In plan R1 there are two such operators: crossover and mutation. An offspring will lie in H_i only if the k positions which define H_i remain unchanged between parent and offspring. Intuitively, if crossover occurs within these defining positions, one or more of them will likely be changed. Hence it is fairly easy to show that the probability of a parent in H_i producing an offspring outside H_i is no greater than $\frac{d(H_i)-1}{\ell-1}$, where $d(H_i)$ is the "definition length" of H_i, namely, the length of the smallest segment containing all the defining positions of H_i as illustrated below.



As a consequence we note that crossover has little effect on the allocation of trials to the bandits associated with short-definition hyperplanes (relative to ℓ), while the allocation of trials to long-definition hyperplanes is considerably disrupted.

The probability of a parent in H_i producing an offspring outside H_i via mutation is just $P_m * \frac{k}{\ell}$, where P_m is the probability of a gene undergoing a mutation and k is the order of H_i. In nature and generally in plan R1, $P_m \leq .001$. Hence, mutation has very little effect on the allocation of trials according to performance.

In summary, then, by looking at hyperplane partitions of A, we have gained considerable insight into the behavior of reproductive plans. In the first place, this analysis yields a criterion for artificial genetic operators, namely, the ability to generate new individuals in A without disturbing too much the near-optimal TVS allocation of trials. Secondly, we can now describe the way plan R1 searches the space A. It generates near-optimal allocation of trials simultaneously to short-definition hyperplane partitions. As elements of high-performance hyperplanes begin to dominate A(t), we have a reduction in the dimension of A and a corresponding reduction in the definition lengths of hyperplanes, providing for another cycle of near-optimal sampling.

2.7 An Example of R1

As an illustration of the discussion in the previous sections, we consider a simple problem for adaptation. Suppose each alternative solution to a problem is represented by a single real number in the interval $[0,10]$ with a precision of 2 decimal places. Suppose further than the performance associated with each solution point is given by $f(x)=x^2$ with the higher valued solutions being the better ones. We choose the representation space A for R1 as follows: There are $(10-0)*10^2$ distinct solutions; hence, $\log_2(10^3)=10$ bits are required for a binary representation. The correspondence between $[0,10]$ and A is given by:

$$x = \frac{a_i}{100}$$

So, for example,

$$0.01 \longleftrightarrow 0000000001$$

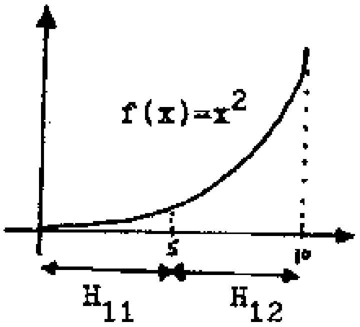
$$5.12 \longleftrightarrow 1000000000$$

In order to see how R1 searches A, we focus our attention initially on a first-order partition P_1 defined by:

$$H_{11}\text{: } 0\text{---}\ldots\text{---}$$

$$H_{12}\text{: } 1\text{---}\ldots\text{---}$$

P_1 simply divides the space in half:



Since R1 generates an initial population A(0) randomly from a uniform distribution over A, we expect half of A(0) to lie in H_{11} and half in H_{12}. Notice, however, that $\overline{f}(H_{11}) < \overline{f}(H_{12})$. Since P_1 is a short-definition partition relative to $\ell=10$, plan R1 will allocate trials to H_{11} and H_{12} according to the near optimal time-varying strategy (TVS) described earlier. In other words, R1 quickly generates a population A(t) consisting almost entirely of individuals from H_{12}.

We now consider a refinement P_2 of P_1 given by:

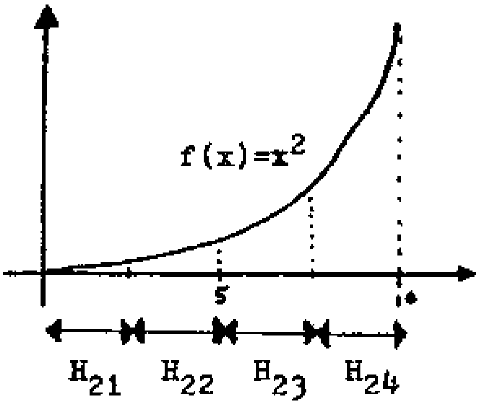
H_{21}: 00---...--

H_{22}: 01---...--

H_{23}: 10---...--

H_{24}: 11---...--

P_2 simply divides the space in quarters:



As we noted above, R1 rapidly generates a population $A(t_1)$ in which most individuals begin with a 1. This is in effect a reduction in the search space A to an ℓ-1 dimensional space. Hence, after a few generations, P_2 effectively becomes a first-order partition of $A_{\ell-1}$ to which R1 now allocates a near-optimal sequence of trials. Since $\overline{f}(H_{23}) < \overline{f}(H_{24})$, R1 rapidly generates a population $A(t_2)$ which lies almost entirely in H_{24}, effecting yet another reduction in the search space.

The important thing to note here is that the same remarks hold for any other short-definition partitions, for example:

---...--0

---...--1

While such partitions are harder to visualize, each is being sampled at a near-optimal rate simultaneously by R1. It is this parallelism which gives even simple reproductive plans like R1 their surprising adaptive capabilities.

2.8 Summary

In this chapter we have defined a class of genetic adaptive models called reproductive plans. These artificial systems are motivated by the kinds of models used in population genetics to explain the adaptive behavior of natural systems. The central feature of these reproductive plans is that new solutions to the problem for adaptation are generated by selecting individuals from the current population on the basis of their observed performance to produce offspring via genetic operators. By focusing our attention on hyperplanes on A rather than individual elements of A, we were able to characterize the way in which reproductive plans search A, and the characterization provided a criterion for genetic operators. Finally, we saw that even the simple reproductive plan R1, because of its ability to simultaneously allocate trials at a near-optimal rate to a large number of hyperplanes on A, exhibit considerable improvement over random search.