University of Plymouth

PEARL **https://pearl.plymouth.ac.uk**

04 University of Plymouth Research Theses 01 Research Theses Main Collection

2000

Exploring and Exploiting Models of the Fitness Landscape: a Case Against Evolutionary Optimization

Moore, Jonathan Paul

http://hdl.handle.net/10026.1/1796

http://dx.doi.org/10.24382/4603

University of Plymouth

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

Exploring and Exploiting Models of the Fitness Landscape: a Case Against Evolutionary Optimization

by

Jonathan Paul Moore

A thesis submitted to the University of Plymouth
in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy

School of Computing
Faculty of Technology

August 2000

900432711 X

UNIVERSITY OF PLYMOUTH

Item No. 900 432711X

Date 14 SEP 2000 T

Class No. T 519.542 MOO

Contl. No. X704423710

LIBRARY SERVICES

REFERENCE ONLY

LIBRARY STORE

Exploring and Exploiting Models of the Fitness Landscape: a Case Against Evolutionary Optimization

Jonathan Paul Moore

Abstract

In recent years, the theories of natural selection and biological evolution have proved popular metaphors for understanding and solving optimization problems in engineering design. This thesis identifies some fundamental problems associated with this use of such metaphors. Key objections are the failure of evolutionary optimization techniques to represent explicitly the goal of the optimization process, and poor use of knowledge developed during the process. It is also suggested that convergent behaviour of an optimization algorithm is an undesirable quality if the algorithm is to be applied to multimodal problems.

An alternative approach to optimization is suggested, based on the explicit use of knowledge and/or assumptions about the nature of the optimization problem to construct Bayesian probabilistic models of the surface being optimized and the goal of the optimization. Distinct exploratory and exploitative strategies are identified for carrying out optimization based on such models—exploration based on attempting to reduce maximally an entropy-based measure of the total uncertainty concerning the satisfaction of the optimization goal over the space, exploitation based on evalutation of the point judged most likely to achieve the goal—together with a composite strategy which combines exploration and exploitation in a principled manner. The behaviour of these strategies is empirically investigated on a number of test problems.

Results suggest that the approach taken may well provide effective optimization in a way which addresses the criticisms made of the evolutionary metaphor, subject to issues of the computational cost of the approach being satisfactorily addressed.

To Lois

Take thou the writing: thine it is. For who
Burnished the sword, blew on the drowsy coal,
Held still the target higher, chary of praise
And prodigal of counsel—who but thou?
So now, in the end, if this the least be good,
If any deed be done, if any fire
Burn in the imperfect page, the praise be thine.

—R.L.Stevenson

Contents

List of Figures

List of Tables

Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This study was financed with the aid of a studentship from the Engineering and Physical Sciences Research Council.

Relevant conferences and seminars were attended, at which work was presented.

Supervisors

Professor Mike Denham.

Dr Susan Denham.

Dr Ian Parmee.

Conferences Attended

Adaptive Computing in Engineering Design and Control. University of Plymouth, UK. March 1996.

Inaugural Workshop of the Centre for Computational Neuroscience and Robotics. University of Sussex, UK. 1997.

Signed ....Jonathan Moore.................... Date 5 September 2000

Chapter 1

Introduction

1.1 Introduction

This thesis addresses the topic of computer-based optimization of engineering designs. Along with many other domains, in recent years this area has seen increasing interest in the adoption of problem-solving techniques inspired by complex biological systems, in particular the adoption of an evolutionary metaphor for design optimization.

It is argued in this thesis that the evolutionary metaphor is not a suitable framework with which to understand and conduct optimization processes. It is proposed that an alternative framework should be adopted, which considers as central the nature of the optimization problem and the knowledge available to apply to its solution.

The thesis is structured as follows:

Chapter 1 outlines a number of criticisms of the use of an evolutionary metaphor for design optimization.

Chapter 2 proposes an alternative framework based on the use of available knowledge to construct models of beliefs about the nature of the optimization problem. Exploitative and exploratory strategies for conducting optimization based on such models are proposed.

Chapter 3 provides a mathematical formulation of the conceptual model presented in chapter 2, using Bayesian belief revision to construct probabilistic models of the goal and performances surfaces for the optimization. Mathematical forms are suggested for the exploitative and exploratory strategies.

Chapter 4 describes an implementation of the exploratory and exploitative optimization strategies for a simple class of problem.

Chapter 5 reports experimentation carried out to investigate the behaviour and properties of the strategies implemented in chapter 4.

Chapter 6 discusses the findings of the experimentation in terms of the criticisms of the evolutionary metaphor presented in chapter 1, and indicates possible directions for future research.

1.2 Engineering Design Optimization

1.2.1 The Knowledge Principle and Optimization

Krottmaier [66] describes design optimization as the middle stage of a 3-stage process of engineering design:

1. System development.

2. Parameter optimization.

3. Determination of tolerances.

The first of these stages corresponds approximately to the conceptual and embodiment design stages in canonical models of the design process, where high-level engineering knowledge and experience are applied to identify the form of one or more systems which will meet the design requirements. This is a very broad and interesting area of research, but not one addressed in this thesis.

In the second stage the variable parameters of the design are optimized against some set of criteria, typically including performance and cost measures. One or more optimized designs are analysed in the final stage to determine the required manufacturing tolerances for the optimized parameter values, to ensure that the product can be manufacutured to a sufficiently consistent quality. It is the second stage—parameter optimization—which is the concern of this thesis.

Consider a design process within an engineering business which consists, broadly, of the following stages:

1. A request is made by a customer to submit a design to fulfil a particular specification. The specification develops through discussion with the customer to clarify requirements and flesh out details more thoroughly.

2. Initial design choices are made by the designer: for example, deciding which of two product ranges will best satisfy the requirements of the specification. Refinement of the space of possible designs proceeds through a number of design stages until the design (or, more likely, specific subsystems of it) is sufficiently well defined for:

3. Computer-based optimization of the design parameters.

At each of these stages, knowledge is being applied in a way which will shape the final form of design proposed:

1. The customer requires knowledge of the need which the designed artifact is to satisfy, along with knowledge of companies which are likely to be able to produce artifacts with the necessary qualities. In selecting one or more companies to submit designs, the customer is effectively using this knowledge to restrict the space of artifacts considered to those which may be designed and constructed by the companies approached. The subsequent refinement of the specification through discussion is brought about by the integration into the process of the more specific domain knowledge of the designers—for example, adjusting expectations of what is actually possible, or highlighting problems or advances in the domain with which the customer was unfamiliar.

2. The designer has a space of possible designs to choose from consisting of all artifacts which might conceivably be built by the company. The process of refining this space into a form sufficiently well-defined for optimization is another application of knowledge—each design choice made represents, at least, an assertion that, given all the previous choices made, the space of designs retained for consideration is believed more likely to contain a satisfactory solution than the space rejected.

3. Finally, the selection of an optimization technique (and any necessary parameters) requires knowledge on the part of the designer of the relative merits of different techniques when applied within the design domain of interest.

The knowledge employed by the designer in stages 2 and 3 above may be of many different forms, for example:

- Mathematical knowledge of the design domain which enables choices to be made based on precise (or acceptably so) analyses of the options.
- Knowledge gained by experience of past designs, either as heuristics describing the behaviour of the domain or as specific examples used in a case-based fashion.
- Knowledge compiled by others and made available in, for example, reference books or computer software.

It is important to appreciate that the knowledge used need not be accurate and factual—it may be heuristic, uncertain, or actually incorrect. The certainty, completeness, and correctness of the knowledge used in the design process is clearly capable of affecting both the quality of the final design obtained and the efficiency of the design process. If, for example, the designer is not familiar with important recent technological advances in the domain, we may expect the quality of the design developed to suffer: if he mistakenly believes that the best method of optimization to use is a random search, we may expect the efficiency of the parameter optimization stage of the design *process* to be adversely affected, regardless of the quality of the solution which is finally reached. It may happen, especially in the design of novel systems, that a stage is reached where existing knowledge is not sufficient to continue the design, and additional knowledge must be sought. Such additional knowledge might be obtained by eliciting it from others, by research into existing knowledge of the domain, or by experimentation with the class of system (or models thereof) being designed.

This characterization of the design process illustrates the *knowledge principle* from artificial intelligence: that problem-solving power derives primarily from the problem-specific knowledge which can be applied to the problem (see, for example, [69]). It is the correctness, specificity, and appropriateness of the design knowledge which can be brought to bear at each stage of the design process which determines the degree of success and efficiency with which a solution to the design problem is reached.

This thesis is based on the premise that inasmuch as the knowledge principle applies to the other stages of design, so it applies also the the parameter optimization stage: as a problem-solving activity, it is the degree to which specific knowledge of the nature of the problem can be brought to bear which will determine optimization success.

The view of optimization taken in this thesis is as a sequential decision process: at each stage in the optimization, a decision must be taken as to which possible design (or *point* in the design space) should be evaluated next. The decision made may be based on *a priori* knowledge of the design domain, coupled with the knowledge developed during the optimization: the latter contained in the set of design points so far evaluated, and their evaluations. In this respect, optimization may be viewed as a process of experimentation and refinement of knowledge, based on the availability of an executable model for assessing the expected performance of different designs. The desired end of an optimization process is an increase in the knowledge of the designer about the specific aspects of the nature of the design space: the key to successful attainment of this end is the appropriate deployment of existing knowledge in order to guide a process of experimentation and knowledge refinement.

1.2.2 Optimization Aims and Efficiency

Keane and Brown [64] point out that the prevalent view of optimization as an attempt to locate the global optimum is unrealistic in the face of most practical engineering optimization problems, where the size of the space of possible design and the expense of evaluating each mean that only a tiny fraction of the space can be investigated during optimization. In engineering contexts, a number of different types of aim may exist for optimization processes undertaken, including:

- Satisficing or constrained design problems, where some (or all) of the design requirements take the form of performance thresholds which must be met, but performance need not be optimized beyond that threshold.
- Global optimization, where the very best possible design from the space of all possible designs is sought.
- Improvement on existing systems, where a significant advance in performance is sought, but not necessarily the global optimum.
- Low sensitivity to design parameter or environment perturbation. Krottmaier [66] sees adjustment for manufacturing tolerances as a separate design stage, but others (e.g. [84]) see low sensitivity as a valid constraint for inclusion in optimization.
- A range of different high-quality alternative designs. In practice, an optimization problem is unlikely to be sufficiently well specified to allow the determination of the single best design—there will be trade-offs to be made once the range of feasible high-quality designs is established. For this purpose, it is better that an optimizer provide a range of distinct options, rather than a single "optimum" design.

Any individual optimization problem may have overall aims made up of a composition of any or all of the above types (and possibly others). In this thesis, the broader view of "optimization" as encompassing all the above possibilities is taken, although practical investigation will be restricted to satisficing problems.

What is sought when an optimization technique is applied to a practical problem is *efficient* optimization. All research into the applicability of particular optimization techniques in particular domains is essentially addressing the question of which optimization method is the most efficient for particular problems. Haataja [48] states that in practical situations, a good solution now is often better than the promise of a perfect one in the future, and that the time needed to provide the solution is an important factor, to be traded off against the final quality of solution.

From a practical point of view, the efficiency of an optimization algorithm must be traded off against the effort needed to adapt the algorithm to the problem in hand: for a single problem, it is not worth spending days refining the algorithm if the effort only results in a few hours of saved computer time. Hence the search for robust algorithms, which can be applied to a wide range of problems with reasonable success—i.e. an acceptable level of efficiency in finding a solution.

1.3 Genetic Algorithms

> "The concept that evolution, starting from not much more than a chemical "mess", generated the (unfortunately vanishing) bio-diversity we see around us today is a powerful, if not awe-inspiring, paradigm for solving any complex problem." David A Coley, [16]

Genetic Algorithms (GAs) are techniques based on mirroring in artificial systems (usually computer software) the mechanisms of biological evolution and natural selection. From foundational experimental work such as that by Rechenberg [95] and theoretical analysis by Holland [58], interest in the application of GAs to a wide range of problem-solving activities has grown enormously in recent years.

Interest in GAs forms part of a general trend towards the interchange of ideas between biology and engineering and computer science, exemplified by the book by Paton [85]. Such a "cross-over" area is bound to attract researchers from a wide range of disciplines, and with an even wider range of agendas. At one end of a spectrum are researchers in "Artificial Life", broadly interested in using computer simulations of natural systems to further understanding of those systems. The term "artificial evolution" is often used to describe such simulations (see, for example, [17]). Researchers at the other end of this spectrum are attracted more by the possibilities of applying biologically-inspired problem-solving techniques to technical and engineering problems. Somewhere in between lie those attempting to create novel forms of artificial systems with desirable properties which have to date only been exhibited by natural systems. In addressing optimization of engineering designs, this thesis takes a position firmly at the applied engineering end of the spectrum.

One common use of GAs is for optimization. In his introduction to the area for scientists and engineers, Coley (see the quote above) essentially classifies the genetic algorithm as an optimization technique, and Haupt and Haupt [54] approach the topic from much the same point of view. The title of Paton's book "Computing with Biological Metaphors" [85] characterizes the use of GAs in engineering well: the processes of natural selection and evolution are seen as a metaphor for a style of approach to problem-solving which may be applied in engineering domains.

An essential question to ask before adopting such a metaphor is whether or not it is appropriate to the domain and problem in question: Does the metaphor truly reflect the nature of the system we are using it to reason about? It is my contention that a number of the characteristics of evolutionary systems make them less likely to be well-suited for use to solve parametric design optimization problems in engineering domains than is commonly supposed, despite the wide range of research in which just such use is made.

In pursuit of this argument, several aspects are discussed below in which GAs differ from more traditional optimization and search procedures:

1. GAs work with a coding of the parameter set, not the parameters themselves.
2. GAs search from a population of points, not a single point.
3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
4. GAs are stochastic: they use probabilistic transition rules, not deterministic rules.

These differences are those identified by Goldberg [37] as being differences between GAs and traditional optimization methods. (Although Goldberg does not explicitly call these "advantages" of GAs, as he is a keen proselytizer for the technique—see, for example, [39], [40]—that implication is there.) It will be suggested below, however, that each of the above characteristics has distinct potential disadvantages, which are especially visible when the subject is approached from the knowledge-use-centred viewpoint adopted above.

To the above list, I shall add for discussion three additional characteristics of GAs:

1. GAs have complex adaptive dynamics, mirroring the natural system on which they are based.
2. GAs are evolutionary in nature. (By "evolutionary" here, I intend the broadest meaning of the word, not any specific biological interpretation: see 1.3.7 below for details.)

3. GAs achieve their results by competition between individuals within the population, leading to an overall pressure towards improved individuals.

Each of the above characteristics will be addressed individually below. However, I shall begin with a note on the teleology of genetic algorithms.

1.3.1 Teleology

> "Biologists often speak, in a kind of verbal shorthand, as though useful traits were evolved purposefully, using statements such as: Fish evolved complex motor systems to coordinate their quick swimming movements. What they really mean, though, is that the fish that by chance happened to have a few more neurons in the motor parts of their brains did a little better. They survived in greater numbers and had more offspring than those that happened to have fewer neurons or less effectively organized ones." Bruce Bridgeman, [10], p 10.

As the above quote illustrates, the ascription of purpose to natural evolution is by no means unknown in scientific literature: phrases such as "designed by evolution" are not uncommon. However, such phrases are figures of speech only: there are few biologists who would truly support the proposition that underlying natural evolution was a directing purpose such as those which direct "...courses of action *aimed at* changing existing situations into new ones."—Simon's view of the nature of design [107] (but my italics).

On the contrary, the theory of natural selection proposes a process by which particular characteristics—already distributed over a population, and possibly hitherto unimportant—have an effect in deciding the survival and breeding success or otherwise of individuals. As such characteristics are passed from successful parents to their offspring, they become more common in the population, leading to a "selection pressure" for particular combinations of characteristics. Evolution extends this theory with the capacity for the chance development of novel characteristics, which may be propagated by natural selection if they prove beneficial.

Harvey [53] emphasizes the difference between such incremental refinement and a typical optimization problem, which he characterizes as a *single* problem, in a well-defined search space. By contrast, "The Sussex approach" to evolutionary robotics [52] is an application of an evolutionary algorithm in a situation much more analogous to the biological context of natural selection: one of adaptive incremental improvement in the light of many consecutive short-term goals, and working with a genetically converged population.

Despite his influential early work on function optimization by GAs, De Jong [25] points out that Holland's [58] original proposal for GAs was as methods of maximizing *cumulative* payoff by adapting a system to a changing environment. He contrasts this with the typical optimization problem, which seeks a *single* solution to a static problem, and where the quality of other solutions tried along the way is irrelevant, so long as the ultimate answer is high-quality. Grefenstette [47] cites as a reason for trying to incorporate heuristic knowledge into GAs that the quality of individual solutions is not the focus of a GA, but rather the overall quality of the population.

Natural genetic processes are *not* goal-driven—this is one of Simon's [107] fundamental distinctions between the "natural" and the "artificial". The fact that complex systems appear to develop in response to natural processes does not mean that the development of those systems can be regarded as having been the *goal* of the processes. Hence it is not necessarily safe to assume that we can adopt similar mechanisms in order to develop a *specific type* of complex system, where the required behaviour of the system is specified as a goal beforehand. The equating of design and optimization with natural selection and evolution, which appears so common ([6], [33], [78], for example) requires more justification than the simple animism which appears usually to underlie it.

In advancing this argument, I echo Culberson's [19] sentiment: "...there is no global requirement on life other than it survive. Evolution was not necessarily looking for the human genome." except that there is not even a requirement that life survive: natural selection and evolution do not know about "requirements", and 99.9% of species to date have not survived [79].

There is thus reason to question whether the evolutionary metaphor forms a sound basis for attempting to solve optimization problems, since the natural process mirrored has different fundamental characteristics from the problem. Other characteristics of the metaphor must be appealed to if its use for optimization is to be justified. However, discussion below will suggest that the other characteristics of evolutionary systems are equally inappropriate for optimization.

1.3.2 Parameter Set Coding

As Goldberg [37] states, most genetic algorithms work with a coding of the parameter set to be optimized, not with the parameters themselves. This GA-encoded form of the parameter set is often referred to as the "genotype" of the solution, the predominating representation used within GA research literature for encoding genotypes being some form of binary coding (although an increasing trend towards real-valued representations for real-valued problems seems apparent in recent years).

In fact, engineering optimization problems will already be posed in some form of encoding, one which encapsulates models of physical systems in a number of parameters. A set of values for each parameter describes a design sufficiently well for the properties of the physical system it represents to be determined to tolerable accuracy. In GA literature, this domain-specific encoded form is often referred to as the "phenotype" for a solution.

A typical real-valued engineering optimization process has a parameter set determined by an engineer with experience in the engineering domain relevant to the problem. The phenotype representation used is thus one which the engineer understands, and in which he is able to think and reason: one in which the relationships between different designs can be expressed, and where similar designs are expected to exhibit similar performance. Mathematical techniques such as dimensional analysis (see [106]) have been developed to enable the engineer to work with representations which capture even more strongly the relationships between designs, by the introduction of concepts such as the Reynolds' number used to characterize situations in fluid dynamics.

If the knowledge to be applied to the solution of a problem is, as suggested here, the critical determiner of the success or failure of the solution process, then the representation used to encode candidate designs during optimization must be one amenable to the expression of the knowledge to be used. The representation used by engineers in the domain will be such a one: possibly developed over many years, and embodying considerable experience. Are we to believe that to take such a representation and re-encode it into a binary form in which domain knowledge cannot be so effectively expressed—in which designs judged similar by the engineer may have wildly differing representations, and very different designs may have very similar representations—will enable an optimization algorithm to make more effective use of the available domain knowledge? Peck and Dhawan [86] claim that in using a binary coding for a naturally real-valued problem, GA practitioners "... rely upon the fortuitous existence of exploitable similarities" between the binary and real representations, and Salomon's [102] report that GA performance on some test functions degrades significantly under a simple rotation of the axes would seem to identify just such a fortuitous similarity. Others ([91], [3]) have shown that a transformation to a binary coding can significantly change the nature of the landscape being optimized.

Considered from the point of view of trying to make maximum use of applicable domain knowledge, the binary representation seems unlikely to be appropriate for real-valued optimization problems. Small wonder, then, that Davis [22] reports never having used the "standard" binary encoding in a practical application of genetic algorithms. Davis takes a similar view to Michalewicz [76]; that domain-specific knowledge is more likely to be exploitable if the "native" problem representation is used. Other researchers have investigated real-coded variants of genetic algorithms for addressing real-valued problems, from both practical and theoretical perspectives, and found the real-coding superior ([4], [23], [31], [34], [54], [50]).

Approaching optimization problems from the perspective of their capacity to make maximum use of available knowledge about the problem seems inevitably to suggest that the encoding used by the algorithm should be the one with which engineers in the domain prefer to work.

1.3.3 Populations

GAs typically work from a population of candidate solutions. In this respect, the knowledge-using properties of GAs would appear to be an advance on typical classical optimization algorithms, which typically work from only a very small number of points. By contrast, GAs retain a larger number of points from which to work towards a solution. However, it is sobering to reflect that when a GA has been running for 1000 generations, typically 99.9% of the potential solutions which have so far been generated and evaluated have also been discarded. The only way in which the knowledge represented by these evaluations is available for use in the optimization is by its indirect effect on the structure of the current population.

It is not readily apparent why a computer-based optimization algorithm should discard any evaluations at all. The biological situation is clear: there is a limited supply of physical material from which creatures can be composed: it must be recycled. There may also be arguments that species with long lifespans are unable to adapt, as a species, to a rapidly-changing environment, and that therefore the "discarding" of previous "solutions"—i.e. the dying out of older individuals to be replaced by their offspring—is necessary for the long-term survival of the species as a whole. Such arguments serve only to throw into relief the differences between natural selection and evolution and processes of design: Evolution and natural selection are *not* design or optimization processes, and don't need to retain records of past "candidate solutions" against the possibility that the knowledge they represent will be needed in the future. What engineering firm would discard all records of previous projects because they are dissimilar to the project currently in-hand?

Thus, it is not clear why (apart from a fixation on replicating a natural system) potentially useful knowledge developed during an optimization process should be discarded. Of course, if such knowledge is not to be discarded, then steps must be taken to make use of it. How we might approach doing this is a separate question: the principle that we should attempt to retain all possible knowledge for use still stands. The loss of potentially useful knowledge in this way is a recognized problem with evolutionary

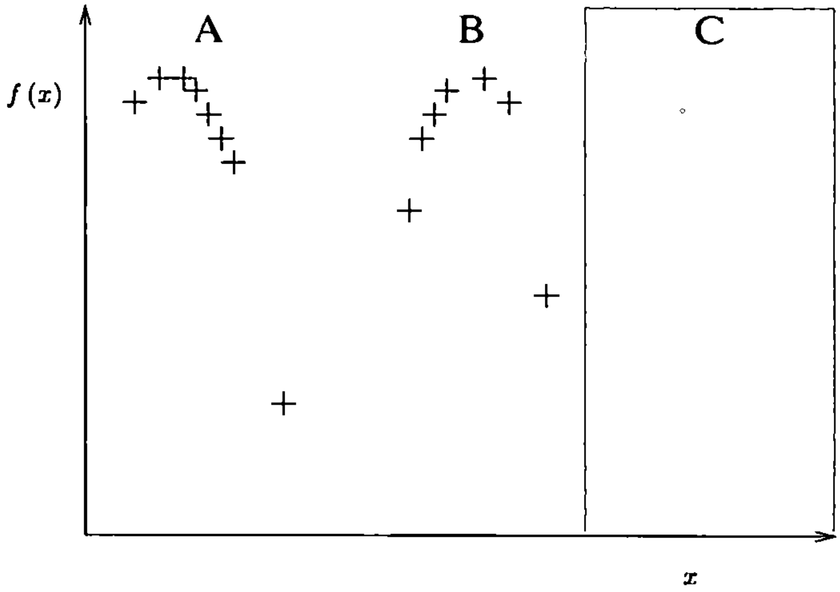


Figure 1.1: *An illustration of the difficulties inherent in discarding information during optimization. An hypothetical population during the later stages of a population-based optimization process is shown. Current data suggests two possible "peaks" of fitness, but is unenlightening about the shaded area of the space.*

approaches: it is the reason for the adoption of "elitism" in GAs, the motivation for the use of "niching" methods (of which, more below), and the drive behind attempts to retain genetic diversity in the GA population.

An illustration of the difficulties inherent in discarding information during optimization is shown in figure 1.1, which shows an hypothetical population during the later stages of an optimization. Two "fitness peaks"—A and B—are visible, but there is no information in the current population about the shaded area of the design space, C. Has area C been investigated, and found to contain only very poor solutions? Or have no points in the area actually been evaluated? Which of these alternatives is actually the case is important for deciding whether or not we should now investigate the region. In a design optimization task, the necessary information is typically in storage, but the genetic algorithm is not in itself capable of making use of it.

I have stated above that population-based methods appear to be an advance on classical optimizers from the point of view of the knowledge retained for use. On deeper reflection, however, it is clear that the knowledge so retained is not all made available for use in each decision about what point to evaluate next. New points to evaluate are chosen based only on a small number (typically two "parents") of the current population. Whether or not a GA is implemented so as not actually to evaluate repeated individuals (as suggested by Haupt & Haupt [54]), the fact that it is capable of generating a candidate solution for evaluation which has not only been evaluated before, but which may even still be present in the population, should sound a warning for those who equate evolution with "intelligent design".

As a final comment on populations, I note that generational evaluation—the accumulation of a pool of solutions before evaluating them—would not appear to be a sensible use of knowledge. Each evaluation affects our knowledge of the space, and the additional knowledge so obtatined may then be useful for deciding which point to evaluate next. Reeves [96] develops a GA for small populations, based on the observation that for problems with very expensive evaluations, a conventional GA is unable to "get into its stride" within the limited number of evaluations which may practically be performed—i.e. A large-population GA does not put the knowledge it develops to work quickly enough. The logical conclusion is that every candidate design should be evaluated immediately upon its selection, and the result of the evaluation made available for use in selecting the next design to be evaluated. This is an approach which is sometimes taken with evolution strategies, but rarely with GAs.

1.3.4 Payoff Information Only

The fact that the GA apparently needs only payoff information is an advantage in that it makes the technique applicable across a wider range of problems than techniques which require a specific form of ancillary information, such as the derivative of the fitness surface, which may not be available.

However, it will be argued in chapter 2 that any non-random optimization algorithm must be making use of more knowledge than merely payoff values, but that the knowledge used by the GA takes the form of assumptions about the design space which are implicit in the algorithm, and correspond in some sense to the use of estimated gradients in classical optimization.

Further discussion is deferred to chapter 2, except to note here that to achieve powerful optimization it is desirable to incorporate domain- or problem-specific knowledge into optimization. The important question for any optimization algorithm which is intended to be widely applicable in different domains is not solely "What ancillary information does it require?", but also "What forms of ancillary knowledge is it able to make use of, should they happen to be available?"

Incorporation of domain-specific knowledge into GAs tends to take one of two forms:

- Hybridization with another optimization method, where the other method embodies the domain knowledge.
- Development of heuristic-based GA operators, for domain-specific versions of crossover, mutation, selection or initialization.

1.3.5 Stochastic Nature

When can it be rational to make a random choice? Conventional Bayesian decision theory mandates a deterministic choice whenever there is even a slight difference between the expected utilities or costs of the options available (see [55]).

A random choice between options can only be justified when there are no means readily available by which to discriminate between the utilities of two or more of the options available: if one has even a suspicion that one option might be better than the others, then that is the option that should be taken.

It seems unlikely that a problem-solving technique focussed on making maximum use of the available knowledge about a problem would make extensive use of random processes, except in a "tie-break" situation between two or more apparently equivalent options. Conversely, it seems unlikely that a technique centred around random processes would be able to make maximum use of all the relevant knowledge about the problem in hand which might be available.

1.3.6 GA Dynamics

GAs and other adaptive optimization techniques are designed to mirror complex adaptive systems found in nature. A review of the literature in the area leads to two conclusions about such systems:

- Understanding them is hard.
- Controlling them is hard.

Control of GA behaviour is generally viewed as a problem in selecting appropriate values for a number of parameters which affect the operation of the algorithm. Typically, these parameters include population size, mutation rate and crossover rate. Early work on control of such parameters focussed on trying to establish a set of values which provided robust performance over a wide range of functions. De Jong [24] carried out an empirical investigation into the optimum parameter settings for a range of fitness functions. Grefenstette [46] applied a meta-level GA to the parameters themselves, measuring their fitness as optimization performance over a range of different functions, while Schaffer et al [104] employed a more exhaustive search of the parameter space. Goldberg [38] presented a theoretical analysis for determining population size.

More recently, with a growing realization that the appropriate parameter settings change during the progress of an optimization (e.g. [118],[32]), attention has shifted towards the dynamic on-line control of GA parameters. Davis [21], Tuson & Ross [112], and Hinterding et al [56] suggest inclusion of operator parameters within the genome, so that they adapt to the problem in-hand alongside the solutions themselves. Research in the GARAGe group (see [116], [117]) uses a separate meta-level GA, operating to adjust the parameters of multiple competing "sub-GAs". Other approaches include the development of rules for adjusting the parameters during an optimization, based on observations of the optimization's recent behaviour (e.g. [94], [1]). Deb and Agrawal [27] note the complexity of the interactions between GA parameters, as do Eiben et al [30]. Harik et al [49] put the case for trying to eliminate the setting of parameters entirely from GA optimization.

Even given a well-defined and understood static fitness function, analysing the dynamics and behaviour of a GA optimizing the function is hard. The difficulties (and the dynamics) are entirely an artifact of the GA—they are not inherent in the problem.

Applying the GA to a more realistic, less well understood problem is then to attempt to "...make a poorly understood system solve a poorly understood problem." [19].

One particular behaviour of the GA which is difficult to justify from a knowledge-centred point of view is the dependence of GA behaviour on the order in which points are evaluated. A GA's behaviour is determined only by the constitution of the current population, and not by the complete set of evaluations so far performed. Take two facts about the design space (two evaluations made during optimization):

1. $F(x_i) = f_i$
2. $F(x_j) = f_j$

Why should the consequences of these facts, and the interactions between them, depend on the stage of the optimization at which they were discovered? By viewing the evaluations so far performed through the "window" of the current population, dynamic behaviour is introduced which is difficult to justify in terms of the problem being addressed.

It is clear from the literature on GAs that modelling and understanding their dynamic behaviour is a difficult task. Is it worth it? If the desired end of the research is an understanding of such evolutionary systems, then the answer is undoubtedly "yes". However, if we are seeking a problem-solving tool, then for such difficulties to be acceptable, they must be offset by considerable advantages to be gained by adopting the approach. This chapter is currently engaged in outlining several reasons for doubting the existence of such advantages.

1.3.7 "Evolutionary" Algorithms

Unfortunately, there does not appear to be a satisfactory word other than "evolutionary" for the concept discussed in this section. It must be stressed, therefore, that the word is not used here in its biological sense, but rather with its more general meaning, which conveys a process of gradual change from an existing form to some similar other form.

Along with almost every classical and adaptive optimization technique, the genetic algorithm is "evolutionary" in this sense. Candidate solutions for evaluation are generated by modification to, and possibly combination of, existing solutions.

This means that the set of solutions which may be proposed as candidates for the next evaluation is constrained to those which are reachable from the current set of solutions (for stochastic algorithms, we have to talk rather in terms of the set of solutions *likely* to be reached, but the same principle then applies). While this might be an advantageous approach to promoting exploitation of existing knowledge about the fitness surface, algorithms which work by such evolutionary mechanisms are inherently limited in their ability to conduct exploration in a principled fashion.

Taking figure 1.1 again as an illustrative example, depending on the encoding and operators in use, region C may not be readily reachable from the current crop of solutions. Yet (depending on the history of the optimization) a strong case might be made that the optimization should now proceed by investigating region C—to return to A and B at a later stage if the investigation proves unfruitful. Qi and Palmieri [89] characterize the behaviour of crossover as exploring the solutions space without increasing the variance of each individual co-ordinate, suggesting that crossover represents a *bounded* stochastic search scheme. This suggests that situations analogous to figure 1.1 might arise in practice, where an unexplored region is difficult to reach from the current population. Evolutionary optimization is then dependent on the chance effects of mutation to initiate investigation of such regions: if these regions are not easily reached by mutation from the current population, premature convergence seems likely to result.

1.3.8 Competition

Competitive selection in the GA leads to predominance of solutions which *are better than the other solutions so far found.* Thus the only pressure for change is due to differences between the fitnesses of points in the current population. Satisficing design problems and optimization in the presence of constraints will therefore pose significant problems: if we have an area of the space which violates constraints less than the other areas currently known about, it will form a focus for convergence, regardless of whether it contains, or seems likely to contain a genuinely feasible solution.

GA optimization can thus converge on a non-satisfactory region of the space because there is no pressure to leave that area in search of somewhere better, until such time as somewhere more promising is actually found. This is a natural consequence of the GA's use of competition between the individuals in the current population to drive change, rather than direct comparison against the goal of the optimization. This is a property of the biological system mimicked: Haataja [48] points to various biological "designs", such as that of the eye, as being sub-optimal, but useful. Since such elements convey an advantage to a species, they are selected *without* "searching" for a "better solution" to the "problem".

There needs to be the capacity for an optimization algorithm to take a decision analogous to "Nothing tried so far seems to work. Let's try something completely different."

i.e. to move into hitherto unexplored regions of the space when no satisfactory solution seems to be forthcoming from those regions so far investigated. In order to do this effectively, the algorithm needs:

1. To know which areas of the space are hitherto unexplored. (In which population-based methods will experience difficulty—see section 1.3.3 above).
2. To have a direct representation of the optimization goal against which to test individual solutions, rather than just making a relative comparison between individuals' performances.

This issue is closely related to that of "evolutionary" methods discussed above: the combination of competition between individuals based on their relative fitnesses, rather than the overall goal, and the generation of solutions for evaluation by variations on existing solutions seems likely to represent a good mechanism for promoting incremental improvement on the current best, but to be inherently limited as far as conducting principled exploration of the space is concerned.

1.4 Some Issues for Adaptive Optimization

1.4.1 Natural Metaphors and the Obfuscation of Problems

To use a metaphor as an aid to understanding or solving a problem means drawing parallels between aspects of the problem and concepts in the metaphor. In doing this, one needs to be careful that the metaphor does indeed provides insight into the problem, and does not rather obfuscate it further. Some reasons have been set out above for doubting that evolutionary processes are a good metaphor for understanding parametric optimization processes.

In using the evolutionary metaphor to reason about optimization problems, one is obliged to attempt to coerce one's knowledge about the design space and fitness surface into a form which matches the terminology and concepts of the metaphor. If the metaphor, as suggested above, is not really matched to the problem, then we can expect it only to further confuse the situation, rather than helping to solve the problem. Falkenauer [31] proposes that the GA must be fitted to the problem in-hand, rather than the reverse. In this he echoes, from the perspective of "real-world problems", the theoretical work of Wolpert and Macready [119] (of which more in chapter 2), which concludes that the nature of the problem is central, and ideally an algorithm for optimization should be constructed from what is known of the problem.

The discussion of binary encoding above is a case in point. If we just consider the problem of optimization from the point of view of its appropriate use of available knowledge, then we conclude strongly that, whatever the optimization technique used, it should employ the encoding preferred by engineers in the domain. To adopt a binary encoding is then to coerce the problem in hand to fit the metaphor (by analogy with the discrete structure represented by the four bases in DNA).

Similar mismatches may be seen in other aspects of the application of genetic algorithms for engineering optimization: take the use of "niching" techniques, for example. These are often equated with "speciation"—the maintenance of multiple different species within an ecosystem, each exploiting a particular niche of the system in order to survive. Niching techniques are adopted to "keep alive" multiple regions of the space which look promising, to prevent premature convergence to a single region, which could overlook a better solution which might be reached from another niche.

Approaches to niching are many and varied, reflecting the centrality of the problem of maintaining sufficient diversity in a population to escape premature convergence. De Jong [24] proposed a technique he called *crowding*, while Goldberg and Richardson [41] developed the *fitness sharing* approach. Sareni and Krähenbühl [103] present an "elitist" modification of sharing. Variations and hybridizations of the GA and other adaptive techniques in attempts to maintain within the population multiple distinct promising sub-populations abound ([9], [28], [43], [72], [80], [82], [100], [111]).

How does an engineer proceed when faced with a situation in which there are multiple possible design alternatives to be investigated? A typical response might be:

1. Prioritize the alternatives: decide which are most likely to be successful, and which less so.

2. Put all but the most promising solution to one side, and investigate that one possibility.

3. Depending on the results of investigation so far, either determine that a satisfactory solution has been found, and investigation can stop, or choose the next most promising alternative, and repeat.

Nowhere does this procedure admit the possibility of completely losing any of the alternatives just because it is not currently being actively investigated. In forcing the problem to fit the metaphor, irrational behaviour is promoted—potentially useful knowledge is thrown away—just because that is how the metaphorical system behaves. As discussed in section 1.3.3 above, there are good reasons why the biological system has a limited population size, and cannot retain a complete history of all past "candidate solutions", but these reasons do not transfer well into the engineering domain.

1.4.2 Exploration and Exploitation

The conflict between exploration and exploitation of the space was noted by Holland [58]. His distinction appears to be:

Exploitation. Re-use of a highly-fit, previously-tried solution.

Exploration. Use of any solution not previously tried.

Without exploration, no new knowledge can be gained which may then be exploited, and so the system cannot improve its performance, yet in exploring new solutions the possiblity of using poor-quality solutions which bring down overall performance is admitted.

The above definitions make sense in the context of "on-line" adaptive systems such as Holland considered. They do not transfer directly into the domain of optimization (at least of static functions), where on-line performance is not critical, and re-evaluation of a previously evaluated solution conveys no benefit. In the context of GAs in design and optimization, the terms appear to take on slightly different meanings. Exploitation is viewed as the use of known highly-fit points (usually by small variations in a local search) to derive even fitter points. Exploration is viewed as an attempt to obtain good coverage of the whole space, in the hope of locating promising regions in which to exploit. Goldberg and Voessner's [42] distinction between the search for "targets" and the search for "basins of attraction" captures the flavour of the two concepts well.

Exploration and exploitation will be returned to in chapters 2 and 3. It will suffice here to make two points:

First, exploitation is not equivalent to convergence. A rational use of existing knowledge might be to abandon the current optimum region once it has been thoroughly investigated, in favour of another region which may have been less well investigated so far, and seem reasonably likely to contain highly fit points. Thus, neither is exploitation equivalent to local search: exploitatory behaviour may manifest much of the time as local search, but large steps in the space, based on global knowledge of the problem, may sometimes be the most exploitative action to take.

Secondly, exploration is not equivalent to random search. There is an aim behind exploration—in terms of extending the current state of knowledge in such a way as to enable successful future exploitation. Given an appropriate representation of the problem and current knowledge of the design space, it should be feasible to locate a single point, or set of points, whose evaluation is judged maximally exploratory—likely to yield the maximum utility in terms of knowledge useful for later exploitation.

Thus, if optimization can be approached explicitly in terms of capturing, representing, and using as much relevant knowledge as possible, then we may expect both exploration and exploitation to be deterministic behaviours.

1.4.3 Convergence

Classical optimization techniques tend to converge. Given that most are gradient-following (whether the gradient be calculated or estimated), and assume a smooth unimodal function, this is not surprising. Convergence proofs are important in assessing and comparing the efficacy of different classical optimizers.

Vose [115] points out that for a stochastic optimizer such as the GA, a slightly different definition of convergence is required from that usually adopted in classical optimization, and presents a definition in terms of the time to locate the search (excluding the noise effects of the genetic operators) in a neighbourhood of the design space of a given size.

Other research (see [12], [13], [51], [67], [110]) has assessed, and attempted proofs for, the convergence properties of GAs.

One question which does not appear to have been asked is whether convergence is actually a desirable property of an optimization algorigthm. Most seem implicitly to take the same line as Rudolph [101], who regards convergence to the global optimum as time tends to infinity as a minimal requirement for the behaviour of a stochastic optimizer.

Consider, for a moment, the behaviour of a converged optimization process. Either no new points in the space are being sampled at all, or the new points being sampled are clustered around a well-investigated local optimum, and not generating any significant new information about the design space. This is reasonable behaviour for an optimization algorithm *only* under the assumption of unimodality of the space. Under this assumption, if a stationary point has been identified, and the region around it investigated fully enough to establish that it is an extremum, then optimization may halt since the global optimum has been found. Deb [26] cites failure to guarantee to locate the global optimum as a disadvantage of classical optimization techniques: but this is precisely what such techniques typically *do* guarantee. However, they are explicit about the situations (i.e. the range of optimization problems) to which this guarantee applies.

For complex spaces where no assumption of unimodality can be made, convergence is a positively undesirable behaviour. A converged algorithm may have found a good solution, but in a complex space this is not generally certain to be the optimum. There are other uninvestigated points throughout the space which still retain the possibility of being better than the current optimum: the optimization algorithm should be investigating these, not clustering its efforts around the current best. Even if no better solution is found, an increase in confidence that the global optimum has been located should result. Syrjakow and Szczerbicka [109] list the increase in success with every stage as a basic requirement for multi-stage optmization processes—why should not continuous improvement, whether of the solution itself or our confidence in it, not be expected of any adaptive optimizer? It is not *premature* convergence which is the problem for GAs: it is any convergent behaviour at all.

1.4.4 Constrained Optimization

Most engineering optimization problems are constrained ([75], [74]). Optimization in the presence of constraints is an area of major challenge facing the evolutionary computation field. Michalewicz [77] gives an excellent overview of current approaches to handling constraints.

Perhaps the ideal approach is to adopt an encoding and associated genetic operators which do not permit the construction of solutions which violate the constraints. However, this approach requires a degree of knowledge about the nature of the constraints which seems unlikely to be available in many practical engineering contexts. Bäck [5] advocates the use of repair operators, finding their use an improvement on the use of penalty functions, but these are also likely to require a degree of knowledge of the nature of the constraints which will often not be available.

The prevalent constraint-handling method is through the use of penalty functions, a technique borrowed from classical optimization ([92], [14],[57], [108]), in which a function of the degree to which an individual solution violates the constraints is used to reduce that individual's fitness. The major difficulty with the use of constraints is summed up by Pearce [90]: "If the effect of a constraint violation is to increase the fitness to a level that compensates for the incurred penalty, then this [the constraint violation] will occur." That is, trade-offs may occur between the objective and the constraint penalties which permit the propagation of solutions which violate the constraints to an unacceptable level. Schoenauer and Xanthakis [105] conclude that there is no general solution to the problem of adjusting the relative weights of the objective and the various penalties to ensure a successful optimization.

Richardson et al [98] warn against too sharp penalization, since GAs need to retain the use of the partial information represented in those solutions which violate constraints. One widely adopted approach to achieving successful constrained optimization is to vary the weights given to penalty functions over the course of the optimization. Kim and Myung [65] monitor constraint violation during optimization, but only apply penalties if the best individual in the population violates the constraints, and then apply them gradually, to steer search back into the feasible region of the space. Powell and Skolnick [88] separate optimization into two phases: the search for and characterization of feasible regions of the space, followed by optimization within the feasible regions—the latter being performed by a numerical method, which is claimed to be better at following constraint boundaries than the GA used in the first phase. Rasheed and Hirsh [93] adopt a heuristic of trying to maintain equality of weight between objective function and constraint violation, by adjusting penalty weights such that the overall fitnesses of the best individual in the population (regardless of constraint violation) and of the individual which least violates constraints are equal. Paredis [81] applies co-evolution to constrained optimization, with a population of constraints which evolve (by changing the strength with which they are applied, not their actual nature) alongside the population of solutions. Finally, others approach constrained optimization as a multiobjective optimization problem, in which the degree of violation of each constraint is treated as a variable to be optimized alongside the objective function (e.g. [8],

[11], [15], [45], [72]).

1.5 Conclusion

1.5.1 Summary

The foregoing discussion has been harsh on the use of genetic algorithms. It should be emphasized that this is not intended as a critique of all uses of GAs: in particular, classifier systems, embedded controllers, applications such as the evolutionary robotics described by Harvey [53], and design systems in which there is dynamic interaction between the engineer and an ongoing evolutionary process (e.g. [83], [44]) do not come within the scope of the criticisms made. Such cases are much closer than are optimization problems to the on-line adaptive systems originally envisaged by Holland as the contexts in which GAs would be useful, and more convincingly analogous to the evolutionary metaphor.

The objections raised are to the use of evolution as a metaphor for optimization problems which may be characterized as the solution of a single problem, off-line, and in which the design space and the fitness function are static. The fundamental objection which appears to lead to all of the problems described above is that the natural system which provides the metaphor for genetic optimization is *not* a problem-solving technique: complex behaviour and inscrutability should not be confused with underlying purpose. From the adaptation of the characteristics of biological evolution, an "optimization" technique results which:

1. Does not explicitly represent the goal of the optimization.
2. Is not set up to make maximum use of the available knowledge.

If a problem-solver does not represent its goal, how can it be expected to represent multiple goals, as required for constrained (or multiobjective) optimization problems? How can it be expected to recognize that achieving the goal would be better served by abandoning current efforts in favour of a less thoroughly investigated area of the search space? How can it be expected to know which areas have not been thoroughly investigated if it discards the very information on which such knowledge must be based?

The use of the evolutionary metaphor for optimization is misguided. In adopting it, we place another layer of complexity between ourselves and what is really important: the nature of the problem in hand, and the specific knowledge we can bring to bear for its solution.

1.5.2 Aims

Culberson [19] considers there to be "... no reason to believe that a genetic algorithm will be of any more general use than any other approach to optimization." I have outlined in this chapter some reasons for believing that we might in fact expect it to be worse than many other possible approaches.

The question investigated in this thesis is therefore: Can effective optimization be achieved by a purely problem-centred approach, based explicitly on making the maximum possible use of the available knowledge of the domain and the problem, without resorting to metaphor?

In the light of the discussion presented in this chapter, such an approach to optimization may be expected:

1. Explicitly to represent the goal of the optimization.

2. Not to discard any information generated during the optimization.

3. To use its representation of the goal, together with knowledge of the problem and the set of evaluations so far performed to select the next point for evaluation.

4. To perform such selections deterministically, and singly (not according to any "generational" scheme).

5. Not to comprise a dynamical system: to have behaviour dependent on the current state of knowledge about the problem, but not on the order in which that knowledge was obtained.

1.5.3 Scope and Limitations

The consideration of optimization in this thesis will be restricted to those in problems in which:

- The evaluation of any design is performed by computer simulation, and no tractable mathematical formulation of the evaluation function is available.
- Design evaluations are computationally expensive. Rasheed and Hirsh [93] note that typical engineering problems involve expensive evaluation functions, while Rao [92] observes that as available computing power increases, the complexity of the models which engineers wish to optimize also increases. It will be assumed that the computational expense associated with any optimization technique will be dominated by that of the evaluation function.
- The function to be optimized is a real-valued function of real-valued parameters, and is static and non-noisy.

- The aim of the optimization is to locate one or more points within the design space which exhibit specific properties, expressed in terms of the values of the evaluation function. Practical investigation will focus on satisficing problems—the location of a point with a performance which exceeds a given threshold.

- The efficiency of an optimization process is judged purely in terms of the number of evaluations performed before the aim is achieved.

Chapter 2

A Conceptual Model of Optimization

2.1 Introduction

In chapter 1, reasons for doubting the applicability of evolutionary metaphors to problems of optimization of engineering designs were advanced. The objections made were founded on the adoption of the *knowledge principle*: that problem-solving power stems from bringing to bear on the problem as much problem-specific knowledge as possible. Genetic algorithms were criticised as being poor vehicles for the expression and use of such knowledge for optimization problems.

In this chapter, an alternative approach to optimization is set out. The knowledge brought to bear on the problem is treated as central, and is hypothesized to be embodied in a model of the surface being optimized. It is proposed that such a model may be considered to underlie the operation of every optimization algorithm, being implicit in assumptions about the surface which must be met for use of the algorithm to be successful. Consideration is then given to how such a model, if it were to be made explicit, might be used to achieve robust, successful optimization.

2.1.1 Terminology & Notation

The following terminology and notation is used in this and following chapters:

Design space, $\mathcal{X}$. The space of possible designs over which optimization is being performed. Elements of this space will be referred to interchangeably as *designs* or *points*, and denoted x.

Performance space, $\mathcal{F}$. The space in which the measures of design performance which are of interest are expressed. Elements of this space are denoted f.

Performance (Evaluation) function, F. A mapping such that each element of $\mathcal{X}$ maps to exactly one element of $\mathcal{F}$. This is the function used to determine the expected performance of any design, and corresponds to the fitness function in genetic algorithms (and, like the fitness function, is not a *function* in the strict sense). Denoted $F : \mathcal{X} \to \mathcal{F}$.

Goal. A boolean property which may be possessed by points in the design space, and which is expressed in terms of the absolute or relative values of the performance function at points in the design space.

Goal and performance surfaces. Associated with the representation of the design space, $\mathcal{X}$, is some form of measure of the similarity of designs, which lends a structure to the space, in that it allows the determination of how "close" one design is to another, and the definition of neighbourhoods of similar designs. For a binary-coded space, such a structure might be the binary hypercube, with the

Hamming distance between designs as the measure of similarity; for a real-coded design space, the structure might be $\Re^n$, with the Euler norm as the measure of similarity.

The variation of the value of the performance function over this structuring of the design space will be termed the *performance surface*, and the corresponding variation of the goal the *goal surface*.

Optimization aim. The aim of optimization will be taken to be the location of a single design which satisfies a particular goal (or—as discussed below—for which there exists a very high level of confidence that it satisfies the goal).

Optimization efficiency. In line with the approach adopted in section 1.5.3, the "efficiency" of an optimization refers to the number of evaluations performed before the aim of the optimization is achieved.

2.2 Assumptions and Models

2.2.1 Rationale: No Free Lunch

Wolpert and Macready [119] derive some "no free lunch" theorems for search. The basic theorem establishes that the efficiency of any optimization algorithm is identical to that of any other, when averaged across all possible performance surfaces relating the design and performance spaces in question. As a corollary, the average efficiency of any algorithm is identical to that of a random search process. This may at first appear a surprising theorem, in the light of two empirical observations:

1. Optimization algorithms are successful. In particular domains and on particular

problems, specific algorithms are found to be much more efficient than random search.

2. Different algorithms are observed to exhibit very different levels of efficiency when applied within the same domain.

The reason that there is no conflict between the no free lunch theorems and these observations is that the theorems are expressed in terms of average efficiency across all possible performance surfaces. This permits an algorithm to exhibit high efficiency on some of the possible surfaces, as long as a correspondingly low efficiency is displayed on others.

Robust success of an optimization algorithm in a particular domain must therefore be indicative of a correlation between the set of surfaces for which the algorithm's efficiency is better than average and the set of surfaces which are experienced in practice in optimization problems within the domain in question [35].

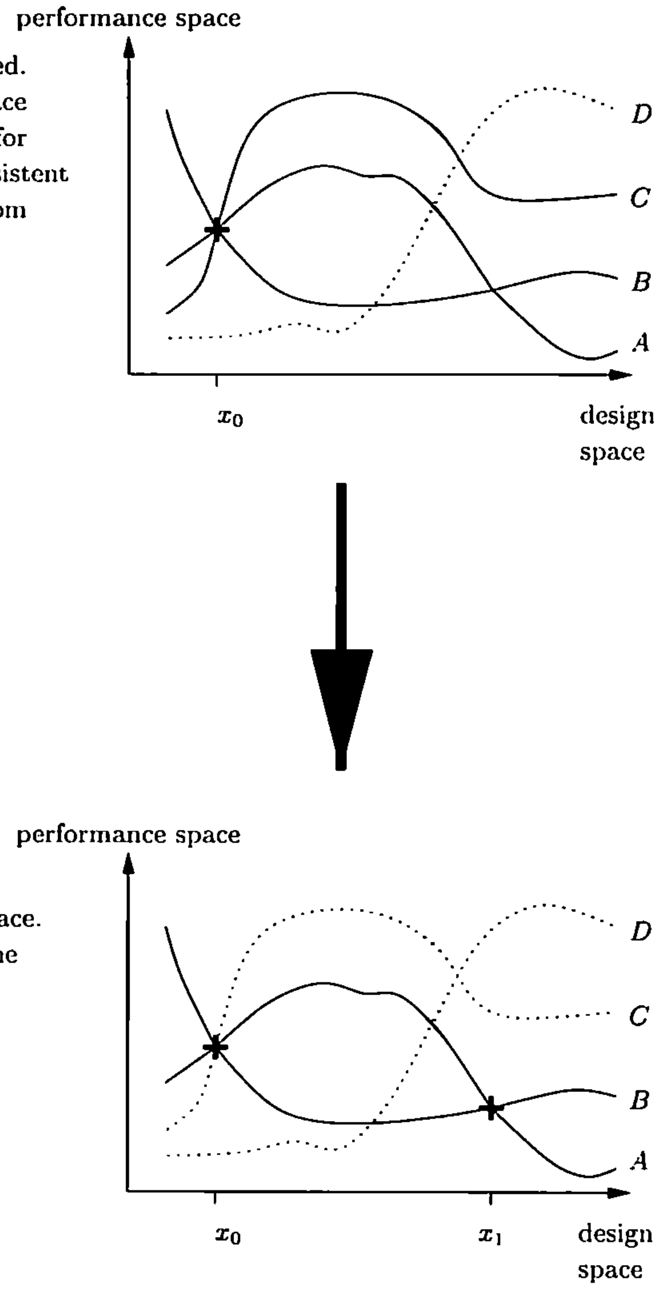
In this thesis, the assumption is made that for any optimization algorithm of practical interest there exist systematic, structural characteristics of surfaces, the presence or absence of which distinguishes the surfaces on which an algorithm performs efficiently from those on which it performs poorly. The exhibition of these structural characteristics may then be viewed as being a prerequisite for surfaces on which the algorithm may be successfully and efficiently used.

2.2.2 Assumptions and Models

For a typical engineering design optimization problem, the size of the design space and lack of a tractable mathematical description of the performance function mean that there is not sufficient knowledge available *a priori* to determine whether the surface satisfies the structural prerequisites for any particular algorithm. The use of an algorithm in the expectation of efficient optimization of a particular surface therefore represents the adoption of an *assumption* that the relevant structural requirements are met.

Therefore, in selecting an optimization algorithm for a specific design task, one is in effect stating that one expects the surface to be optimized to be one of the set which display the relevant structural properties for that algorithm to be successful—that is, one is committing to a *model* of the performance surface. The term *model* is not here intended to convey a construct which assigns a fixed value for the performance of every point in the design space, such as the quadratic surface models used in some classical optimization methods. Rather, it is used in its more abstract mathematical sense—the model is an expression of some specific aspects of the system modelled, which allows predictions to be made about certain properties and behaviours, while falling short of being a complete representation of every detail of the system. In the present case, the model of the surface is envisaged as specifying the set of surfaces which are regarded as possible candidates for the true fitness surface being optimized.

Consider how such a model may be adjusted during optimization. Before any points have been evaluated, the model includes as possibilities all the surfaces relating the design and fitness spaces which satisfy the algorithm's basic assumptions. Upon evaluation of the first point selected, however, many of these possible surfaces will typically be found to conflict with the datum obtained: these must then be discarded as candidates for the true surface. As the optimization proceeds, each evaluation has a similar effect, pruning the set of possible surfaces described by the model. This process is illustrated in figure 2.1.



A single point, x_0 has been evaluated. The model of the performance surface permits A,B and C as possibilities for the true surface, since they are consistent with the datum. D is eliminated from consideration, since it conflicts with the known datum.

x_1 is chosen as the next point for evaluation

A and B are retained as possible candidates for the performance surface. C is discarded as it conflicts with the new datum.

Figure 2.1: *An illustration of the effect of successive design evaluations in pruning the set of surfaces permited by the model of the performance surface. Solid lines show surfaces retained as possibilities within the model; dotted lines show surfaces which conflict with observed data, and are therefore excluded from the model.*

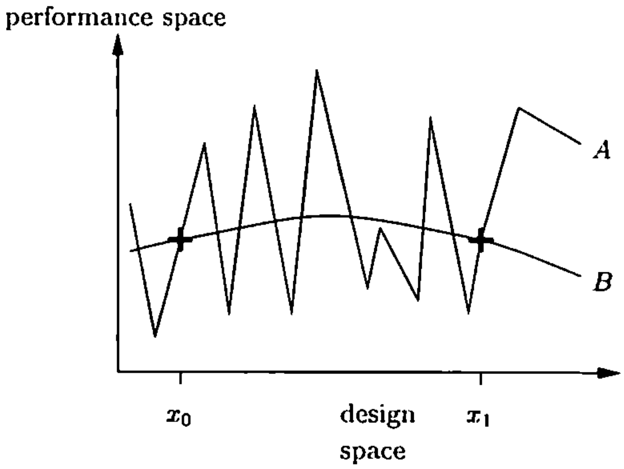


Figure 2.2: *Modelling the relative plausibility of surfaces. Both the surfaces shown are strictly consistent with the known data points. However, experience may lead us to consider one more likely than the other.*

2.2.3 Relative Plausibility Models

Given the experience-based nature of knowledge about most design domains, one is unlikely to be able to be precise in specifying the assumptions made about a problem—one may, for example, expect a surface to be continuous *almost everywhere*, or consider it to be *unlikely* that any point will be found with a performance which exceeds a particular value. The model.of the performance surface may therefore specify the relative plausibilities of the different possible surfaces which are retained as valid possiblities. In figure 2.2, for example, if during previous investigations within the design domain only smooth and continuous surfaces have been encountered, surface B may be considered more likely than surface A to be the actual performance surface, although both are strictly consistent with the known data, and may be regarded as possible.

2.2.4 Implicit Assumptions and Models

The argument developed above suggests a framework for considering the operation of all optimization algorithms. By the selection of an algorithm, the designer commits himself to the assumptions about the performance surface which are fundamental to that algorithm's efficient operation. These assumptions correspond to a model of the possible performance surfaces; a model which is then refined and revised as the optimization progresses, by the incorporation of the effect of knowledge of the performances of the points evaluated.

With most classical optimization techniques, the requirements which the objective function must satisfy in order for optimization to be successful are explicitly specified. These conditions usually include unimodality, and often continuity or smoothness, while some techniques make more stringent restrictions such as that contours in the space be elliptical. As has been stated above, the lack of knowledge about the nature of the performance surfaces in practical engineering optimization problems means that such conditions in effect become assumptions about the surface when the algorithm is used in such a context.

With adaptive optimization techniques such as the genetic algorithm, the situation is not as clear. It is not readily apparent what are the structural characteristics of a surface which predispose it to being successfully optimized by any particular variant of genetic algorithm. The fact that we are unable to determine or express the assumptions or model underlying an algorithm should not, however, be taken as evidence that no such assumptions or model exist: in such cases, the assumptions, and the models constructed therefrom, are implicit within the mathematical and dynamical structure of the algorithm, and may thus be difficult to identify.

2.2.5 Explicit Assumptions and Models

The no free lunch theorem leads us to a view of optimization in which optimization algorithms implicitly embody a model of the surfaces to which they are applied. This model, being a synthesis of the a priori assumptions made about the surface and the data developed during the optimization process, serves as the central repository of available knowledge during optimization.

Wolpert and Macready [119] conclude from their work developing the no free lunch theorem that to approach optimization by specifying an algorithm, and then considering the nature of the surfaces to which the algorithm may successfully be applied, is to approach from the wrong direction. It is the nature of the problem in hand—the surface being optimized—which is central, and from which the appropriate algorithm should ideally be derived.

In the light of the above discussion, it would seem that optimization might sensibly be approached by first identifying what is known about the nature of the surface to be optimized (or what one is prepared to assume to be true), then using this knowledge to construct an explicit model of the surface. The model so constructed (as the repository of all available knowledge about the surface) can then be used as the basis for the decision about which point to evaluate next. As successive evaluations are fed back into the model, it is to be hoped that the increasing available knowledge about the surface would enable the decisions made to be more effective, leading eventually to the satisfaction of the aim of the optimization.

It is clear that at the heart of any model to be used for optimization in this way is uncertainty. In maintaining multiple surfaces as possible candidates for the true surface being optimized, the model captures the uncertainty extant about the true nature of the surface. We may speak of the model as combining the fundamental assumptions made about the surface with the data generated so far during the optimization process, to express the current *beliefs* held about the nature of the surface.

2.3 Goals

It may not be possible to determine with certainty whether a given point satisfies certain classes of goal. In such a situation, the aim of the optimization is better expressed as being to reach a state in which a high level of certainty exists that the goal has been achieved—the situations where it is possible to satisfy a goal with complete certainty are then special cases of this more general formulation.

A satisficing goal, for example, may be achieved with complete certainty: if the performance of a design exceeds the target threshold, then the design achieves the goal. By contrast, in the absence of knowledge of the mathematical nature of the performance function, the global optimum is unlikely to be locatable with complete certainty, even though there may be a high level of confidence—based on extensive investigation of the surface—that the optimum has truly been found.

These examples are two extremes of a spectrum of goals, distinguished by the amount of knowledge of the performance surface—beyond simply the performance of the single point in question—which is required in order to determine the satisfaction or otherwise of the goal by any particular design. The goal of locating a design with low sensitivity to perturbation lies somewhere between these extremes: only a finite number of points in the region around the design can be evaluated, so in order to conclude that the design shows low sensitivity, assumptions must be made about how knowledge of these points' performances constrain the possible performance values for the whole region. These assumptions are precisely those captured in the model of the performance surface: so we may see that a model of the satisfaction or otherwise of the optimization goal over the design space—the *goal surface* for the problem—may be constructed from the model of the performance surface. Like the performance surface model, the model of the goal surface captures current beliefs and uncertainty about the satisfaction or otherwise of the optimization goal across the design space.

We may expect both the model of the performance surface and that of the goal surface to be useful for optimization. The performance surface model captures the current state of belief about the nature of the performance surface, but it is the goal surface model which relates directly to the aim of the current optimization, and which is likely to form the main basis for decisions about which point should be evaluated next, in order to further this aim.

2.4 Optimization Strategies

Assuming performance surface and goal surface models as outlined above to be available, consideration needs to be given to how, in practice, the knowledge they embody may be employed for optimization. This will be addressed in chapter 3 for the form of probabilistic model developed there: it will suffice here to point out that distinct strategies can be identified, corresponding to exploitation and exploration (as discussed in section 1.4.2):

Exploitation. Select points for evaluation to which the model of the goal surface attributes a high certainty that they achieve the goal.

Exploration. Select points for evaluation which are expected to reduce the level of uncertainty in the model of the goal surface, making future exploitation of the model easier.

A minimum requirement for a selection strategy is that it be *rational*, in the sense of "economic rationality" as described by Lane et al [68]. That is, that the point selected for evaluation should be selected on the basis of a calculation of the value of the consequences associated with its evaluation. For exploitation, the calculation of value would be the level of expectation that the point selected will satisfy the goal: for exploration it would be the a measure of the reduction in uncertainty which might be expected to result from the evaluation.

2.5 Discussion

2.5.1 Summary

The above analysis has outlined a conceptual model of optimization for use both in considering the operation of existing algorithms and for developing new ones. The no free lunch theorem has been used to provide theoretical support for the knowledge principle espoused in chapter 1: that it is problem-specific knowledge which is the key to solving optimization problems.

The conceptual model developed distinguishes between a number of key components of an optimization process:

- The aim and corresponding goal of the optimization.
- The underlying knowledge possessed, or assumptions made about the surface to be optimized.

- The set of evaluations so far performed on the particular optimization problem currently in-hand.
- The construction of models of current beliefs and uncertainty about the nature of the performance and goal surfaces by combining the underlying assumptions with the current set of evaluations.
- The use of the models for optimization, via distinct strategies for selecting points for evaluation, which may be applied to achieve either exploitation or exploration.

The next section revisits some aspects of genetic algorithms in the light of the presented model of optimization and this separation of optimization components.

2.5.2 Genetic Algorithms Revisited

Problem, not Algorithm

There appears to be a reluctance in the GA optimization community to consider the full import of Wolpert and Macready's no free lunch theorem. The first question to be asked is not "what are the consequences for evolutionary optimization?" but "what are the consequences for the field of optimization as a whole?" As Wolpert and Macready point out, the fact that no generally superior optimizer can exist places the specific optimization problem, not the generic optimization algorithm, at centre-stage: one should attempt to work from the nature of the problem towards an algorithm which might solve it. Investigations into particular forms of algorithm and the kinds of problem to which they may be applied are only useful to the extent that they assist with the reverse form of reasoning—from the nature of the problem to a suitable algorithm.

Efficiency *or* Robustness

The pursuit of an optimization technique which is robustly efficient across all problem domains is, according to the no free lunch theorem, a wild goose chase. Grefenstette's [47] hope that genetic optimizers would turn out to be a "powerful weak method" can be seen to be unfounded (with, admittedly, a great deal of hindsight). Deb's [26] desire for an algorithm which is generally robust across different classes of problem is unlikely to be satisfied—unless significant exploitable commonalities between the structures of the classes of problem in question can be identified. This may not be a vain hope (see section 2.5.3 below), but it is through study of the problems, not of any particular optimization algorithms, that such commonalities may be identified and exploited. Horn et al's [59] conclusion that the GA is a robust method is questionable: the finding that a GA can solve "longpath" problems which are tough for hill-climbers merely begs the question of where the "no free lunch trade-off" occurs—which are the unimodal, hill-climber-easy problems which prove tough for GAs?

The GA and Problem-Specific Knowledge

The efficiency with which any given algorithm solves a given problem can be seen to be dependent on the problem-specific knowledge which the algorithm enbodies—in the form of assumptions made about the nature of the surface being optimized. Claims such as that "Blind search strategies do not use information about the problem domain." [36] or that "...prior information [about the problem] is not essential [for a specific optimization algorithm]." [87] do not stand up under the no free lunch theorem.

Kargupta [62] points out that "in the absence of any analytic information about the objective function structure, a BBO [Black Box Optimization] algorithm must guess based on samples it takes from the search space", but also that such guesses are taken based only on consideration of "a certain finite set of features that defines the bias of the process." That is, in the absence of analytical knowledge to underpin optimization, an algorithm must assume that the surface is one of the set towards which it is "biased", and proceed accordingly.

In this context, it is best to regard Holland's schema theorem not so much as an explanation of *how* GAs work, but as a requirement which must be satisfied by a fitness function before a GA can be satisfactorily employed to optimize it [22]. The requirement is that the encoding for the space, together with the fitness function, must result in the existence of building blocks with a degree of independence in their effect on fitness (in biological terminology, having only low or medium *epistasis*—see [20]), and which can be combined effectively by the crossover operator in use. In this respect, the exploitation of schemata in the GA parallels the use of estimated gradients in classical optimization: each represents a fundamental assumption about the nature of the relationships between the performance of different designs in the design space. It is the exploitation of these relationships on which successful optimization relies.

A particular genetic algorithm is located within the space of GAs by the nature of its problem representation, operators, and the value of any controlling parameters. Establishing the relationships between these features and the types of problem on which the GA will be efficient is a recurring theme of research. Ronald [99] advocates remapping the encoding in use if better building blocks or a decrease in epistasis results, which corresponds to attempting to restructure the surface being optimized to better match the assumptions underlying GA optimization. Bäck [3] shows that the optimal mutation rate to use depends heavily on the simple question of whether the surface is unimodal or multimodal. Manderick et al [73] show strong relationships between the correlation coefficients of GA operators on a given fitness landscape and the success of a GA in optimizing that landscape. The tuning of GA parameters represents adjusting the assumptions implicit in the algorithm better to match the problem or class of problems in hand.

In recommending that any black box optimizer (BBO) should "quantify its bias", Kargupta and Goldberg [63] are effectively stating that the assumptions underlying the operation of any algorithm—and hence the set of surfaces on which it can be used—should be made explicit, as is the case for classical optimizers. The difficulty for the GA is that these assumptions are obfuscated by the very structure of the algorithm, and distributed across all of the operators involved, plus the problem representation used. Every crossover, mutation and selection operator, along with each of the various representations which may be used, has its specific bias: the bias of the overall algorithm is determined by the interaction of all these individual biases. This is perhaps the reason why most research into the practical application of GAs has an empirical flavour—different variations being tried, and reasons behind their relative success or failure then being sought.

The success of many such variations can be understood by asking the simple question "does the variation allow the use of more problem-specific knowledge than was previously the case (or better use of that which the algorithm already employs)?" It is argued in section 1.3.2, for example, that the use of a real coding allows better use of problem-specific knowledge in a real-valued space. Michalewicz's [76] approach of varying the mutation rate of different bits on the genome during GA optimization makes use of knowledge not previously available to the GA: that the genome consists of distinct sections, each representing a single parameter, and where each parameter comprises bits with varying significance. Paredis's [81] approach of co-evolving constraints makes use of the knowledge that there *are* constraints, which are distinct from the fitness function—knowledge not available to an algorithm which just receives fitness values which are already penalized for constraint violation. Rasheed and Hirsh [93] construct genetic operators for real-valued spaces which are clearly based on assumptions of continuity and on the exploitation of local gradients. Many other variants are explicitly based on the use of existing heuristics within a particular domain, or for a particular class of problem (e.g. [2]).

Exploitation and Exploration

In the same way that the assumptions about the fitness surface underlying GA operation are distributed across the operators and representation used, so too is the distinction between exploitative and exploratory behaviour. The model of optimization presented in this chapter promises to separate the two behaviours into distinct strategies, which seems likely to make balancing between them more straightforward.

Crossover may exhibit both exploratory and exploitatory behaviour, depending on the similarities between the parents selected to breed and the sites on the genome chosen for crossover—both typically stochastic choices. A mutation of a binary chromosome may have either an exploitative or an exploratory effect, depending on the degree of disruption to the phenotype represented by the allele mutated—again, a stochastic choice. The adjustment of GA parameters such as the mutation and crossover rate give only a very indirect level of control over which behaviour is promoted. Furthermore, such control tends to be exercised on a fairly coarse-grained level—typically setting parameters to try to exploit either exploration or exploitation for an entire population breeding cycle, rather than choosing the most appropriate behaviour for each individual selection decision.

In neither crossover nor mutation is exploitative or exploratory behaviour rational, with respect to the overall state of knowledge of the problem. Either operator may generate a point for evaluation which not only has been evaluated before, but is still present in the current population. Selection of such a point for evaluation has no value for either exploitation or exploration.

Assumptions and Deception

Deb et al [29] describe how to construct surfaces which are "maximally deceptive" to genetic algorithms. These surfaces exhibit local features which tend to divert genetic search away from the true global optimum. Kallel and Naudts [61] describe how to construct a *longpath* problem for a GA: a problem which uses a series of "landmarks" to lead the search astray, resulting in extremely inefficient performance in reaching the global optimum.

That such problems will exist is an inevitable consequence of the no free lunch theorem. Deceptive surfaces will exist for any algorithm; the real question is whether they are likely to appear in practice within the domain in which the algorithm is applied.

As Venturini [113],[114] and Liepins and Vose [70] make clear, the genetic algorithm can readily be modified so that deceptive surfaces of the type identified by Deb et al no longer prove deceptive. However, to adopt such an approach has two important consequences:

1. Efficiency in optimizing non-deceptive functions will be reduced (even if only marginally), since evaluations must be expended in order to check that the surface is not deceptive.

2. A different set of surfaces are now deceptive. As Venturini discovered, checking

for maximally deceptive surfaces does not help to eliminate more mildly deceptive ones.

The efficiency of optimization on deceptive surfaces may thus be increased, but only at the expense of reducing the efficiency on previously non-deceptive ones. Under the constraints identified by the no free lunch theorem, the average performance over all functions must remain constant.

So one must return to the nature of the optimization problem in hand: which fitness surfaces are regarded as likely to be encountered, and which as less likely? If one believes that some of the surfaces one will wish to optimize in a domain may be GA-deceptive, then one must either modify the GA to be used to take account of this, or employ another optimization method. Kallel and Naudts [61] hit the nail on the head by questioning whether any real-world problems might exhibit longpath structure.

From this discussion, we may draw a more general conclusion: the assumptions made about a surface cannot be checked as part of the optimization process. To attempt to do so involves checking for specific kinds of deviation from the assumptions, which is equivalent to saying that there is a class of non-conforming surfaces which we deem more likely to arise in practice than other such classes: we are changing our assumptions about the nature of the surface, rather than checking the existing ones.

2.5.3 Concluding Remarks

Saved by the Real World?

The consequences of the no free lunch theorem paint a bleak picture for the development of a widely applicable, generally robust optimizer. One hope, however, still remains:

that classes of optimization problem encountered in practice in the real world may, in fact, share significant properties which may predispose them towards successful optimization by a specific type of optimization algorithm.

Underpinning all optimization efforts is a single assumption: that relationships exist between the performances exhibited by different points in the design space. Without such relationships, any optimization would degenerate to random search. With such relationships, knowledge of the performances of a limited set of designs allows predictions to be made about the performances expected of other designs. By expanding the set of designs whose performances are known in a principled way, such knowledge may be investigated and refined in pursuit of the aim of an optimization task. Reeves and Wright [97] capture this fundamental property of optimization by considering the process as one of experiment design, in which hypotheses are made about the nature of the surface, and experiments (evaluation of points) performed in order to test these hypotheses.

In fact, this fundamental assumption is even stronger: it is that similar designs are more likely to exhibit similar performances than dissimilar designs (Rana and Whitley [91], for example, claim that "optima are often surrounded by relatively good points"). This assumption is the basis of gradient-based methods, as well as fundamental to the applicability of the schema theorem, although each typically uses a different measure of "similarity". The power of the relationships in the space which such a similarity measure allows to be expressed is the factor underlying the differing efficiencies of optimizations using different representations. Jones and Forrest [60] point to the centrality of the concept of "distance" in the consideration of genetic landscapes, in which the genetic operators employed define the "neighbourhood" of any design. The whole concept of searching for "fit regions" or "feasible regions" of a space is predicated on the assumption that similar points (i.e. those within a particular neighbourhood) tend to have similar performances.

For real-valued engineering design spaces defined by parameters which represent physical quantities, we might go further: such spaces *tend* to be continuous. If a design calls for a 5Ω resistor, or a 30mm diameter piston, in order to achieve a given performance, then the performance from using a 5.01Ω resistor, or a 29.9mm diameter piston is likely to be close to that desired (whether it is acceptably close is a different matter). There may, of course, be discontinuities (in an analogue-digital converter, a particular resistance value might mark the point at which a given voltage switches from being classified as 01 to 10; for some value of diameter, the piston will no longer enter the bore for which it is intended), but such discontinuities are likely to be local, and not characteristic of the majority of the design space.

Whether such continuity is an inherent characteristic of physical systems, or merely of the subset with which engineers have chosen to work over the years is a moot point. However, it may tentatively be suggested that an optimization algorithm based on the assumption that most of the design space is continuous, while allowing for occasional discontinuities, might be generally successful in optimizing such "physical system" design spaces.

Conclusion

The key to successful design optimization is the application of appropriate knowledge of the problem. This chapter has proposed that such knowledge underlies the operation of every optimization algorithm, but that it is not always explicit, nor its nature necessarily readily identifiable. In applying any algorithm to a given problem, the knowledge used by that algorithm for successful optimization becomes assumptions made about the nature of the problem.

It has been suggested that optimization may be approached by identifying explicitly the assumptions one is prepared to make about the nature of the optimization problem in hand, and using those assumptions, together with the data generated by evaluations performed during the optimization process, to build models of the performance and goal surfaces for the problem. Such models may then be used by exploitative and exploratory strategies to achieve effective optimization.

Chapter 3

Probability-Based Optimization

Chapter 2 presented a conceptual model of optimization in which assumptions about the nature of the problem are combined with the current set of available evaluations to generate models of the current state of belief about the natures of the performance and goal surfaces. These models then form the basis for successive decisions about which point in the design space to evaluate next. This chapter proceeds to cast the model of chapter 2 in more formal terms, using probabilistic models of the surfaces under a Bayesian view of probability.

Section 3.1 introduces the necessary concepts of Bayesian probability theory. Section 3.2 indicates how those concepts may be equated to elements of the model of optimization presented in the previous chapter, and addresses the construction of probabilistic models of the optimization goal over the design space, while Section 3.3 discusses some possible strategies which may employ such models for optimization. Finally, section 3.4 discusses the expected properties of such an optimization process, in the light of the discussion in chapters 1 and 2.

3.1 Bayesian Probability

3.1.1 The Nature of Probability

Classical probability theory defines probability as the limiting relative frequency of an event as the number of independent identical trials of that event tends towards infinity. The theory is therefore concerned with the long-term aggregate behaviour of random variables, and may not validly be used to make predictions about the results of any individual trial.

By contrast, probabilities in a Bayesian context are treated as measures of the relative plausibility of hypotheses. For an hypothesis U, the probability, $P(U)$, of the hypothesis is a measure of belief concerning the truth or falsehood of U. It is central to this view of probability that U is certainly either true or false, but there is insufficient information available to determine with certainty which value is correct; although there may exist evidence which affects one's judgement of the plausibility of U. Since in the Bayesian view probabilities are treated as measures of the relative plausibility of hypotheses, not as descriptions of limiting frequencies of particular types of event, it is valid to use Bayesian probabilities as an aid for considering single events in a way which is not acceptable under the classical view of probability (see [18] for a discussion).

One consequence of the view of probability taken in Bayesian probability theory is that all probabilities are conditional. The conditioning hypotheses express the nature of the system with which the probability is concerned, and may include constraints imposed by nature, by assumptions made about the system, or by the methods used for sampling the system to obtain observational data. This approach is directly analogous to the standard approach to logical inference—any logical deductions can only be asserted dependent upon their axioms, and are unable to give any information about the truth or falsehood of those axioms. Thus there is always a "given" clause to Bayesian probabilities, expressing the "axiomatic" assumptions upon which the assessment of probability is based.

3.1.2 Bayes' Theorem

The fundamental theorem of Bayesian probability is Bayes' Theorem:

$$P(U \mid DA) = P(U \mid A)\frac{P(D \mid UA)}{P(D \mid A)} \tag{3.1}$$

Where:

- A is the background information defining the nature of the system of interest, and any assumptions which may be made or methods used to gather data.
- D is some observation of characteristics of the system (an hypothesis known to be true).
- U is an hypothesis about some aspect of the system.
- $P(Q \mid R)$ is the conditional probability of some hypothesis, Q, given some hypotheses asserted by R.

Individual terms in Bayes' theorem are conventionally given specific names:

Prior probability $P(U \mid A)$**.** This is the assessed probability of the hypothesis U being true before any data is observed. Its value will depend upon the nature of the system and one's assumptions about the system, and it is therefore conditioned upon A.

Global likelihood $P(D \mid A)$. This is the probability, given only the information expressed in A, that the data actually observed would occur. It does not depend on the hypothesis U.

Likelihood function $P(D \mid UA)$. This is the probability that the data observed would occur if the hypotheses asserted by both A and U were true. The term *likelihood function* is used when the quantity is considered as a function of U; when considered as a function of D, this quantity is known as the *sampling distribution*.

Posterior probability $P(U \mid DA)$. This is the probability of the hypothesis U after taking into account both the hypotheses asserted in A and the data D which was actually observed.

The canonical use of Bayes' theorem is the assessment of the posterior probability of an hypothesis, given both the observed data and the background information expressed in A.

3.1.3 Sequential Belief Revision

A major application of Bayes' theorem in the field of machine learning is in the sequential revision of beliefs, in the light of successive items of evidence becoming available. In this application, current beliefs about the plausibility of some hypothesis, U, are expressed by the assessed probability of the truth of U, $P(U)$. Before any data become available, one may already have beliefs or opinions about U, which are expressed in the selection of the prior probability $P(U \mid A)$ (again, A represents any assumptions which may be made about the nature of the system in question).

As data items, $D_1, \ldots D_t$, become available, which affect one's beliefs about U, the probabilistic representation of those beliefs may be updated by repeated application of

Bayes' Theorem to take into account the new evidence:

$$P\left(U \mid D^{t}A\right) = \frac{P\left(D_t \mid UD^{t-1}A\right)P\left(U \mid D^{t-1}A\right)}{P\left(D_t \mid D^{t-1}A\right)} \tag{3.2}$$

Where D^t represents the conjunction of the data $D_1, \ldots D_t$.

One difficulty in applying this technique lies in the specification of the *conditioned likelihoods*, $P\left(D_t \mid UD^{t-1}A\right)$. Bernardo [7] (which may be consulted for an in-depth treatment of Bayesian modelling and inference) notes that in practice, and especially if t becomes large, it becomes necessary to adopt simplifying assumptions if belief revision by this method is to be tractable: simplifying assumptions such as, for example, the conditional independence of $D_1, D_2, \ldots D_t$, given U.

3.1.4 The Maximum Entropy Principle

An important issue in the use of Bayes' theorem is the generation of prior probability distributions. Specifically: if the hypotheses in A are not expressed in such a form as to determine a unique prior probability distribution for the hypothesis U, how is a suitable prior distribution to be generated?

This question has in the past posed a considerable problem for the Bayesian approach, since the same problem could be approached using a range of different priors, leading to different solutions. However, the *principle of maximum entropy* is now widely recognized as a method which may be used in a consistent way to generate priors for any combination of testable information in A.

The principle of maximum entropy mandates the use of a prior distribution which conforms to the testable information in A, while maximising (over the space of all such conforming distributions) the Shannon entropy of the distribution, which is defined by:

$$H = -\sum_{i=1}^{N} P_i \log_2 P_i \tag{3.3}$$

for a finite distribution over N exclusive, exhaustive alternatives, and by:

$$H = -\int_{\Theta} p(\theta) \log_2 p(\theta) d\theta \tag{3.4}$$

for a continuous probability density p over the parameter $\theta \in \Theta$.

The Shannon entropy is used in this application as a measure of the uncertainty associated with the probability distribution, so that by maximising entropy, the resulting distribution is left maximally non-committal with respect to all characteristics except those constrained by A.

The priors generated by the maximum entropy principle are often referred to as *uninformative priors*, or (more accurately) *least informative priors.*

3.2 Probabilistic Optimization

The concepts of Bayesian probability theory described above map in a pleasing way onto the conceptual model of optimization presented in chapter 2.

First, Bayesian probability theory requires explicit consideration of underlying assumptions about the nature of the system in question, and that such assumptions be taken into account when calculating probabilities. This parallels the proposal that the underlying assumptions made about a surface in order to optimize it should be made explicit.

Secondly, the Bayesian approach treats probability distributions as encapsulations of beliefs about hypotheses, given the underlying assumptions made, plus some relevant data. This mirrors the suggested construction of models of beliefs about the nature of the surface being optimized, based on the a priori assumptions made about the space, plus currently available data (the latter in the form of the set of evaluations performed so far).

Furthermore, every point in the design space being optimized has a single fitness value, and either achieves the optimization goal or does not. The reason for optimizing is that one is uncertain about which points do, and which do not, meet the goal. The concept of "repeated identical trials" simply does not make sense in this context: for any point, the first "trial" determines its value with certainty. This echoes the Bayesian view of probability as being a measure of belief about some parameter which is fixed, but about the true value of which one is uncertain, as opposed to the classical view of probability as a limiting relative frequency of a random variable.

For these reasons, Bayesian probability provides an appropriate representation in which to express models of beliefs about fitness surfaces, for use in optimization as outlined in chapter 2. The concepts of chapter 2 are expressed below in the terms of Bayesian probability.

Optimization Goal

The aim of optimization as addressed in this thesis is to locate one or more points within a design space which satisfy a particular goal. $U(x)$ represents the hypothesis that point $x \in \mathcal{X}$ satisfies the optimization goal, where $\mathcal{X}$ is the space of possible designs to be optimized, and U is expressed in terms of the values assumed by the performance surface relating $\mathcal{X}$ and $\mathcal{F}$ (the space of performance values).

Assumptions

The assumptions made about the space in order to be able to optimize it correspond to the Bayesian "background information", A.

Evidence

The points evaluated so far during the optimization process are used as the successive items of evidence D_i in Bayesian belief revision (see equation 3.2). t represents the number of evaluations performed so far, so that at time t, we have evidence consisting of $D_1 \ldots D_t$, where D_i represents the fact that $F(x_i) = f_i$, x_i being the ith point chosen for evaluation.

Goal Surface Model Form

The model of the goal surface takes the form of a function, over the design space $\mathcal{X}$, which yields the probability that a point $x \in \mathcal{X}$ satisfies the optimization goal (i.e. that the hypothesis $U(x)$ is true). The model after t evaluations have occurred, M_t, is generated based on the set of a priori assumptions, A, which have been adopted about the space, and the conjunction of all additional data which have so far become available during the optimization, D^t:

$$M_t(x) = P\left(U(x) \mid D^t A\right) \tag{3.5}$$

Model Generation

In order to generate the model M_t, the data provided by successive evaluations, $D_1 \ldots D_t$, may be used as items of evidence in a process of Bayesian belief revision, updating the beliefs about the goal surface as encapsulated in the model.

Bayes' theorem (equation 3.1) then gives:

$$M_t (x) = P \left(U (x) \mid D^t A \right) = \frac{P (D_t \mid U (x) D^{t-1} A) M_{t-1} (x)}{P (D_t \mid D^{t-1} A)} \tag{3.6}$$

where $M_0 (x) = P (U (x) \mid A)$ is the prior probability for the hypothesis U at x.

Point Selection

During optimization, the choice of the next point to evaluate is made by a selection procedure, S, on the basis of the current model of the goal, M_t, over the space $\mathcal{X}$:

$$x_{t+1} = S (M_t, \mathcal{X}) \tag{3.7}$$

The strategies which might be adopted to implement such a selection procedure are the topic of the following section.

3.3 Rational Optimization

This section considers possible optimization strategies which might be used to determine the form of the procedure, S, for selecting the next point for evaluation based on the current goal surface model, M_t. It is assumed that the basic requirement for any such strategy is that it be *rational* in the sense discussed in section 2.4. Since the model M_t is the repository in which the assumptions and currently available knowledge about the space are embodied, the question becomes: given a probabilistic model of the form proposed above, how should we set about using that model rationally for optimization of the space?

A simple but appealing strategy will initially be proposed below. Subsequent discussion will identify a number of conceptual problems with that strategy, however, from which a proposal for an alternative strategy will then be developed. These strategies will be identified as corresponding, respectively, to exploitative and exploratory behaviour. Finally, in section 3.4, a composite strategy for rational balancing of exploitation and exploration will be suggested.

3.3.1 The Maximum Likelihood Strategy

The Strategy

Consider first a somewhat reduced optimization problem: given a probabilistic model of the goal surface such as that described above, and the opportunity to evaluate only a *single* point, how should the point to evaluate be chosen so as to have the best chance of achieving the goal with that evaluation?

This is a straightforward problem (in that the basis for the selection of the point is clear; actually performing such a selection may not be so straightforward) in which the optimum decision rule [55] is selection by *maximum likelihood*—choose the point which has currently the highest assessed probability of achieving the goal.

This strategy might be employed for all point selections in a longer optimization process; that is, at every time t we select a point x_{t+1} for evaluation such that:

$$M_t\left(x_{t+1}\right) \geq M_t\left(x\right), \forall x \in \mathcal{X} \tag{3.8}$$

This strategy, which will be termed the *maximum likelihood strategy*, is both simple and intuitively appealing. The consideration of its likely behaviour below also suggests other desirable characteristics.

Expected Behaviour

Consider a situation in which a single region of the performance surface is found to exhibit solutions which are close to achieving the goal, but where other regions of the space are largely unexplored. Under the maximum likelihood strategy, we would expect to observe that a series of points within the high-performance region are evaluated. If none of these points are found to achieve the goal, how will the optimization proceed?

Note that the successive evaluation of points within the high-performance region provides an increasing amount of information about the region. This is likely to result in a reduction over time of the probability of achieving the goal given by the model of the goal surface for all points within the region. This takes place in a context in which there are other regions of the surface about which little is known, and which have a correspondingly high level of uncertainty. It would seem likely that there will come a time when the probability that the goal is satisfied at some point within these more lightly constrained regions will exceed the probability for all points within the high-performance region. When this occurs, the maximum likelihood strategy will cause the optimization to shift from the initially promising (but thus far unrewarding) region into another region—one which was initially less promising, but which now looks somewhat more attractive by comparison.

This strategy is purely exploitative: every point for evaluation is chosen because, based on current knowledge, it is the most likely to achieve the goal. No consideration is given to the information which may result from the evaluation.

Criticism

The discussion above overlooks at least two criticisms which may validly be made of the maximum likelihood strategy.

First, consider an optimization the aim of which is to locate the global optimum of the surface. It may be that the point to which the model of the goal surface assigns the highest probability of being the optimum has already been evaluated, but that there remain regions of the surface containing points with a small probability of having better performances than the optimum so far found. At this point, optimization using the maximum likelihood strategy halts, since a point which has already been evaluated is repeatedly selected. In such a situation, it would be preferable to continue with investigation of other regions: there would still be a chance on improving the solution found, if only by increasing the level of confidence that it is the true optimum. In halting, the maximum likelihood strategy is exhibiting convergence as deprecated in section 1.4.3: it is ceasing to return useful information, when there is still useful information which might be obtained from the surface.

The second criticism of maximum likelihood selection stems from the knowledge-centred view of optimization adopted in this thesis. When conducting an optimization, the aim is that the optimization goal should be satisfied the *at the end of the optimization process.* This is not necessarily equivalent to saying that the algorithm should attempt to satisfy the goal at every individual stage in the process. Consider, for example, trying to climb a flight of stairs by continually trying to place one's foot on the topmost tread. Consideraton of such an analogy suggests a possible principle: only the last point evaluated should be selected using the maximum likelihood strategy to attempt to achieve the goal of the optimization; every previous point should be selected by some different strategy, which aims to maximize the chances that the final evaluation will, in fact, achieve the goal. Such a strategy is suggested in the next section.

3.3.2 The Ignorance Minimizing Strategy

The previous section has presented a rationale for treating the initial stages of optimization as a process of developing knowledge about the goal surface for the optimization, with a view to the actual satisfaction of the optimization goal only with the final evaluation of the optimization. To phrase the problem slightly differently: the early stages of optimization should concentrate on reducing as far as possible the uncertainty about the goal surface, so that it is then easier to locate a point which achieves the goal at the end of the optimization.

The use of the Shannon entropy of a probability distribution as a measure of the uncertainty of that distribution, has already been encountered in section 3.1.4. Since the model of the goal surface which underpins the optimization is a probabilistic one, entropy calculated from the model may be used to quantify uncertainty about whether points in the design space meet the goal or not.

The entropy (see equation 3.3) at x for the probabilistic model M_t (after t evaluations) is given by:

$$H_t(x) = -M_t(x)\log_2(M_t(x)) - (1 - M_t(x))\log_2(1 - M_t(x)) \tag{3.9}$$

I further define the *ignorance* about the goal surface at time t as:

$$I_t = \int_X H_t(x)\,dx \tag{3.10}$$

This quantity expresses a measure of the total uncertainty, over the whole surface, about the satisfaction or otherwise of the optimization goal. In order to reduce uncertainty about the location of points which satisfy the optimization goal, an optimization strategy should therefore attempt to select a point x_{t+1} for evaluation so as to minimize the ignorance I_{t+1}.

The exact effect which the evaluation of any given point will have on the model of the goal surface, M_t, and hence on the ignorance about the surface depends on the performance value which is actually obtained from the evaluation, and hence cannot with certainty be predicted prior to the evaluation's being performed. However, if a model of the probability distribution for the performance of each point is available, then the expected ignorance, EI_{t+1} after the evalution of any point x_{t+1} can be determined.

The strategy proposed, which will be termed the *ignorance minimizing strategy* is therefore to select for evaluation a point x_{t+1} for which the expected ignorance of the goal surface following the point's evaluation, EI_{t+1} is minimized.

Expected Behaviour

Where the maximum likelihood strategy is purely exploitative, the ignorance minimizing strategy is purely exploratory. A point is not chosen for evaluation because it is expected to achieve the goal, but rather for the information it is likely to yield about the locations of other points that will. Regions with very low probability of achieving the goal will tend to be avoided, as will regions with very high probability, in favour of regions with probabilities approaching 0.5. In the case of a satisficing goal, or constraint, for example, the strategy can be expected to focus more evaluations around the boundaries of feasible regions than in the interior or exterior.

For static performance functions, no reduction in uncertainty can ever result from the re-evaluation of a point the performance of which is already known, so we may expect the ignorance minimizing strategy not to exhibit the convergent behaviour described above for the maximum likelihood strategy. Similarly, as more evaluations are placed within a given region of the space, the uncertainty associated with all points in that region is expected to decrease, and the ignorance minimizing strategy to move to other regions of the space which have been less thoroughly investigated.

3.4 Discussion

3.4.1 Overview

The problem of optimization has been approached above from the perspective of the objections raised to the evolutionary metaphor in chapter 1 and the alternative approach outlined in chapter 2.

The optimization method which results from this analysis consists of two distinct components:

1. The use of the Bayesian belief revision method to construct probabilistic models of the satisfaction or otherwise of the optimization goal and the value of the performance function over the design space, based on the set of evaluations performed so far and the a priori assumptions about the space which one is prepared to adopt.
2. The application of an optimization strategy to use the models so constructed to select the next point for evaluation at each stage.

Two strategies have been identified as rational approaches to the second of these components:

1. The *maximum likelihood strategy*, in which the point selected for evaluation is the one believed most likely to achieve the goal.
2. The *ignorance minimizing strategy*, in which the point selected for evaluation is the one which is expected maximally to reduce the current total uncertainty (ignorance) concerning the satisfaction of the goal over the design space.

3.4.2 The Composite Strategy

Criticisms of the maximum likelihood strategy have been discussed above. At least one valid criticism of the ignorance minimizing strategy may also be made, as follows. In situations where the goal may be achieved with complete certainty, the ignorance minimizing strategy will studiously avoid regions of the space containing points with very high probabilities of achieving the goal. Evaluation of one of these points is very likely to achieve the goal, and the optimization could then stop: the ignorance minimizing strategy would not seem likely, then, to lead to efficient optimization.

The crucial point here is that the two strategies identified are not suggested as being appropriate for use on their own. Rather, an overall *composite strategy* is proposed, in which the ignorance minimizing strategy is applied first, in order to develop information about the goal surface which can then be used by the maximum likelihood strategy in an attempt to satisfy the goal. The arguments by which the different strategies were developed above in fact suggested that all but the last point for evaluation should be chosen using the ignorance minimizing strategy, with only the final evaluation selected by the maximum likelihood strategy.

The composite strategy having been developed in response to the objections to the evolutionary metaphor for optimization advanced in chapter 1, the remainder of this chapter discusses its expected behaviour in terms of those objections.

Transparency, Tuning, Determinism and Dynamics

There is no metaphor being used to understand the optimization process. The problem in hand is addressed directly, in terms of the knowledge available to be applied to its solution, and how this knowledge may be used rationally to achieve the aim of the optimization. This has three important consequences, discussed below.

First, there is no tuning of the optimization mechanism necessary. Indeed, such tuning is not even possible: there are no parameters relating to either the maximum likelihood strategy or the ignorance minimizing strategy by adjusting which the behaviour of the optimization process can be altered. Rather, as suggested by the discussion in section 2.5.2 tuning has been relocated to where it belongs: in the expression of knowledge or assumptions about the space.

Secondly, the optimization method is deterministic. It was suggested in chapter 1 that an optimization method which made full use of available knowledge about the space would be fundamentally deterministic, since there would be few cases in which the available knowledge would be perfectly equivocal about the most advantageous point to evaluate next. This has been achieved: there is no random process at the heart of the optimization method proposed. Clear criteria in each case are available to determine the a unique point for evaluation. There is scope for stochastic decision-making only if there are multiple points with the same probability of achieving the goal (for the maximum likelihood strategy), or the same expected effect on the total ignorance of the space (for the ignorance minimizing strategy).

Finally, neither strategy leads to a dynamical system. The construction of the models of the goal and the fitness surface from the set of points so far evaluated is independent of the order in which those evaluations were performed. Coupled with the deterministic nature discussed above, this means that the behaviour of the system is dependent solely on the current state of knowledge about the problem, and not on any artifact of the dynamics of the optimization method.

Population, Evolution and Competition

There is no concept of a "current population". The results of all evaluations performed may be retained, and taken into account in the construction of the models. This property is essential to the exploratory nature of the ignorance minimizing strategy, since to be able to explore the space rationally it is necessary to retain a record of which regions have already been investigated, whether or not they proved to have high performance.

The selection of the next point for evaluation is not made on the basis of competition between the current set of evaluations, based on their relative performances, but rather on a comparison in which every unevaluated point in the design space is treated equally, and judged against a measure of the value expected to result from its evaluation. Because every point in the space is treated as a possible candidate at each stage, the optimization is not "evolutionary" in the sense deprecated in section 1.3.7. Because the measure of value relates directly to the goal of the optimization, rather than just the relative performances of the set of evaluations made so far, the convergence to non-satisfactory regions expected to result from the competitive nature of the GA (see section 1.3.8) is not expected to occur.

Knowledge Use and Goal Representation

The analysis in this chapter has centred on the use of knowledge of the problem in hand as the fundamental driving power behind the optimization process. The success and performance of the strategies presented is likely to depend on the accuracy and constraining power of the knowledge applied to the construction of the models used.

In the analysis above, this knowledge is encapsulated in the background information, A, used in Bayesian belief revision. No suggestion has thus far been made of the form such knowledge might take, nor any evidence advanced that the forms of knowledge typically available for optimization problems will be easily incorporated into the model construction procedures. However, use of the Bayesian approach to integrate different forms of knowledge into a consistent structure of belief about some system is a recurring theme of research in the area. The approach thus seems as promising as any as a potential means for combining different kinds of knowledge into a single formalism which can then be employed for optimization. There are two places in the proposed model of optimization at which knowledge of the problem may be taken into account:

- Generation of the prior probability $P(U(x) \mid A)$. The prior used can take into account beliefs relating to a particular area of the space, or the space as a whole. For example, a belief that a particular class of designs is unlikely to exceed a particular fitness value, or that the design space as a whole will have a given mean performance, might potentially be incorporated into the prior.
- The procedure for generating the successive conditional likelihoods. This incorporates knowledge about how the performance values of different points in the space are likely to relate to each other, which might incorporate observed trends, domain heuristics, and mathematical approximations.

These two options would seem to cover a wide range of possible types of domain knowledge.

Exploration, Exploitation and Convergence

In developing the two optimization strategies described, the distinction between exploitation and exploration has been brought to the fore. Indeed, the two behaviours have been entirely separated, and a distinct strategy been identified for each.

The above analysis suggests strongly that exploration and exploitation are appropriate at different stages during the optimization process. Indeed, if exploration is performed rationally, it might be used for all selections of points to evaluate except the last, and it is only with this last selection that exploitation becomes necessary. The exploitatory strategy may on occasion lead to convergence, but the exploratory strategy will not—it will tend away from regions of the space which have been thoroughly investigated, in pursuit of regions of higher uncertainty. Since the composite strategy allows the exploitatory strategy to be applied only for a single evaluation, it will never have the opportunity to display convergent behaviour, so the overall composite strategy will be non-convergent.

Chapter 4

Optimization Strategy Implementation

4.1 Introduction

Chapter 3 proposed a framework for optimization based on the construction of Bayesian probabilistic models of the performance surface being optimized and of the satisfaction of the optimization goal over the design space. Three strategies were proposed by which optimization might be approached using such models: the *maximum likelihood* strategy, the *ignorance minimizing* strategy, and the *composite* strategy, the last being a combination of the first two.

Chapter 5 reports on experimentation performed to investigate the properties of the different optimization strategies proposed. The investigation described in chapter 5 employs very simple spaces, goals and models. This chapter prepares the ground by deriving the models used, and the procedures for selecting points for evaluation using those models, for both the ignorance minimizing and maximum likelihood strategies.

Section 4.2 describes the assumptions made about the nature of the optimization problem. Section 4.3 describes the model of the performance surface derived from those assumptions, and section 4.4 the corresponding model of the goal surface. Section 4.5 describes the implementation of the maximum likelihood and ignorance minimizing strategies using the given form of model.

4.2 Basic Assumptions

Design and Performance Spaces. Both the design and performance spaces are one-dimensional and real-valued. The design space, $\mathcal{X}$, is a continuous closed section of the real axis: $[x_{min}, x_{max}]$. The performance space, $\mathcal{F}$, is the space of real numbers.

Optimization Goal. The goal of the optimization is to locate a point with a performance which exceeds a given threshold, f^* (a satisficing optimization problem).

Prior Knowledge. Before the start of the optimization, no knowledge or expectation exists about the possible performance values anywhere in the space.

Uniform Continuity. The assumption used to build the models of the performance and goal surfaces is that the performance surface is *uniformly continuous* almost everywhere. Formally, uniform continuity requires that for every $\epsilon > 0$, there exist a $\delta > 0$ such that $|F(x_1) - F(x_2)| < \epsilon$ for all $x_1, x_2 \in \mathcal{X}$, $|x_2 - x_1| < \delta$, where δ depends only on ϵ, and not on x_1 or x_2. (This assumption is slightly weakened by allowing it to apply *almost* everywhere. This retains the possibility of local discontinuities in the performance surface, while retaining the overall belief that the surface is broadly "well-behaved".)

"Occlusion". A corollary of uniform continuity in a one-dimensional design space I shall call the assumption of "occlusion". That is, for any three points $x_1, x_2, x_3 \in \mathcal{X}$ such that $x_1 < x_2 < x_3$, $F(x_1)$ and $F(x_3)$ are conditionally independent, given $F(x_2)$. That is, information provided by the evaluation of one point does not "pass through" another evaluated point to affect the model of the performance of points beyond it. The model of the performance of any point in the space is dependent only on the nearest evaluated points above and below the point of interest.

The assumption of no prior knowledge means that the prior probability density function for the performance of any point is a uniform distribution over all possible values. This distribution has no effect on the Bayesian belief revision process, and so the prior probability will be omitted from the analyses below.

The assumption of uniform continuity forms the background information, A, used in chapter 3 as the basis for generating the models of the performance and goal surfaces. This specific form of background information is employed in the development of the models and strategies in this chapter, and is denoted $\mathcal{A}$.

Some justification for adopting this assumption for optimization of physical systems was given in section 2.5.3.

4.3 The Performance Surface Model

First, the model of the performance surface resulting from the evaluation of a single point is derived. This is then used to derive the model resulting from the combination of the effects of the evaluations of multiple points.

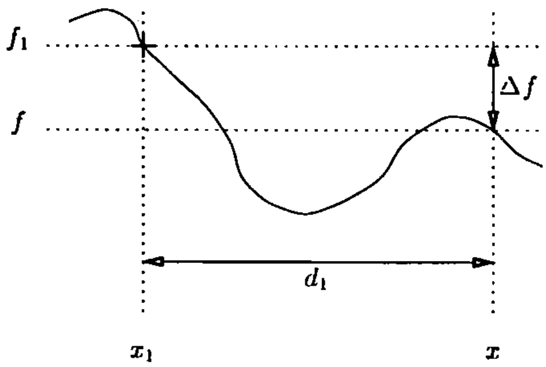


Figure 4.1: The desired form of the performance surface model due to the evaluation of a single point. We wish to determine the probability density for the performance f at x, given the evaluation $F(x_1) = f_1$.

4.3.1 Model from a Single Evaluation

Refer to figure 4.1. We wish to determine $p(f \mid D_1\mathcal{A})$, the probability density function for the unknown performance value f, at x, given the known performance f_1 at x_1 (expressed as the item of evidence D_1), $\mathcal{A}$ represents the underlying assumptions made about the nature of the surface—in this case, the assumption of uniform continuity.

Let $\Delta f = f - f_1$, so that $p(\Delta f \mid D_1\mathcal{A})$ is the probability density function for the deviation, Δf, of the performance at point x from that at x_1. Under the assumption of uniform continuity, $p(\Delta f \mid D_1\mathcal{A})$ depends only on the distance, d_1, between the two points.

The case for adopting a normal distribution as the form of $p(\Delta f \mid D_1\mathcal{A})$ is argued below. Consideration is then given to the nature of the dependence of the parameters of that normal distribution on the separation d_1.

The Form of $p(\Delta f \mid D_1 \mathcal{A})$ is Normal

Uniform continuity suggests that as distance from an evaluated point increases, larger deviations of the performance surface from the performance of that point become more likely, and values close to the performance of the evaluated point become correspondingly less likely. We may thus characterize $p(\Delta f \mid D_1 \mathcal{A})$ as being of variance which increases with d_1. Having no reason to believe a rise in performance to be either more or less likely than a fall, we may also conclude that the distribution is symmetrical, and therefore has a mean of zero. If the mean and variance of a distribution be specified, then the entropy-maximizing distribution (see section 3.1.4) is the normal distribution with the specified mean and variance. For this reason, it seems reasonable to conclude that the form of $p(\Delta f \mid D_1 \mathcal{A})$ is normal, with zero mean and variance a monotonically increasing function of d_1.

This argument closely follows that used [71] in favour of assuming the noise on a random variable to be Gaussian if no other information about the nature of the noise is available: we might regard the performance of an evaluated point in a continuous space as a noisy estimate of the performances of neighbouring points, where we expect the level of noise present in that estimate to increase with distance from the evaluated point.

The Dependence of $p(\Delta f \mid D_1 \mathcal{A})$ on d_1

Refer to figure 4.2, which shows an additional point, x', somewhere between x_1 and x. The "occlusion" corollary of the uniform continuity assumption dictates the conditional independence of f_1 and f, given f'. That is,

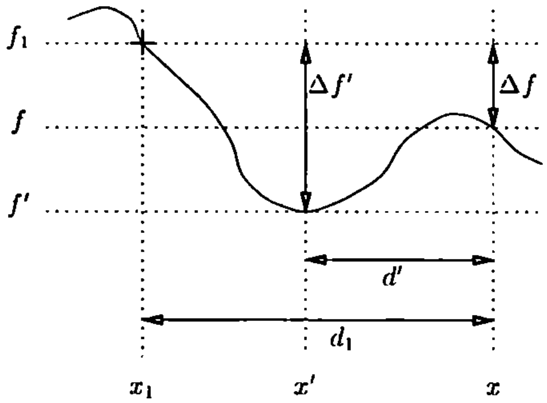


Figure 4.2: *The dependence on the model of the performance surface due to a single evaluation on the distance from the evaluated point. The distributions for f and f_1 are conditionally independent, given f'.*

$$p\,(\Delta f \mid D_1\mathcal{A}) = \int_{-\infty}^{\infty} p\,(\Delta f' \mid D_1\mathcal{A})\,p\,(\Delta f \mid D'\mathcal{A})\,d\Delta f'$$

where D' is the hypothetical datum that $F(x') = f_1 - \Delta f'$. This expression represents the convolution of two normal distributions, the result of which is a normal distribution with variance equal to the sum of the variances of the two convolved distributions. We may therefore conclude that the variance of $p\,(\Delta f \mid D_1\mathcal{A})$ depends linearly on d_1:

$$\mathrm{Var}\,(p\,(\Delta f \mid D_1\mathcal{A})) = \alpha d_1$$

for some appropriately specified constant parameter α.

The Performance Surface Model from a Single Evaluation

Given a single evaluation D_1, which conveys the fact that $F(x_1) = f_1$, the probability density function for the performance f of any point $x \in \mathcal{X}$ is given by:

$$p(f \mid D_1 \mathcal{A}) = \phi(f_1, \alpha d_1, f) \tag{4.1}$$

where d_1 is the separation in the design space between x and x_1, $\phi(\mu, \sigma^2, z)$ is the normal density function of the parameter z, with mean μ and variance σ^2, and α is a constant which parameterizes the assumption of uniform continuity.

4.3.2 Model from Multiple Evaluations

Refer to figure 4.3. We wish to determine $p(f \mid D^t \mathcal{A})$, the probability density function for the unknown performance value f, at x, given a number of evaluations as evidence: $D_1 \ldots D_t$, where D_i represents the fact that $F(x_i) = f_i$. Under the "occlusion" corollary to the assumption of uniform continuity, $p(f \mid D^t \mathcal{A})$ depends only on the two evaluated points which are nearest to x on either side. Without loss of generality, we shall label these points x_1 and x_2. Application of Bayes' Theorem yields:

$$p\left(f \mid D^t \mathcal{A}\right) = p\left(f \mid D^2 \mathcal{A}\right) = p(f \mid \mathcal{A}) \frac{p(D^2 \mid f\mathcal{A})}{p(D^2 \mid \mathcal{A})}$$

$$= \frac{p(f \mid \mathcal{A})\, p(D_1 \mid D_2 f \mathcal{A})\, p(D_2 \mid f\mathcal{A})}{p(D_1 \mid D_2 \mathcal{A})\, p(D_2 \mid \mathcal{A})}$$

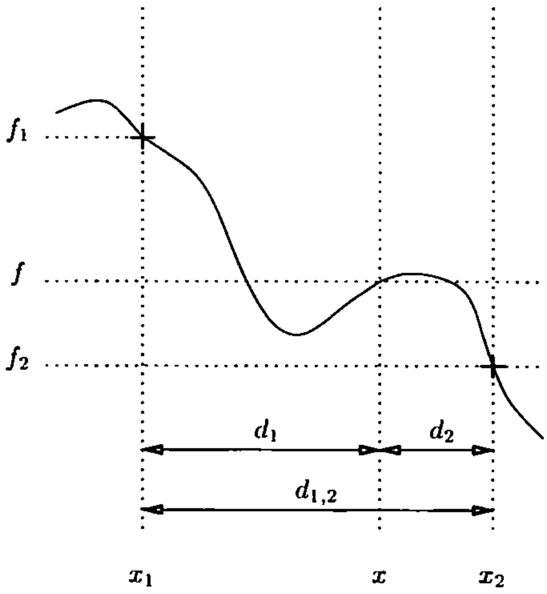


Figure 4.3: *The desired form of the performance surface model due to the evaluation of a multiple points. We wish to determine the probability density for the performance f at x, given the evaluations $F(x_1) = f_1$,$F(x_2) = f_2$.*

"Occlusion" dictates that $p(D_1 \mid D_2 f \mathcal{A}) = p(D_1 \mid f\mathcal{A})$, while under the assumption of no prior knowledge $p(f \mid \mathcal{A})$ and $p(D_2 \mid \mathcal{A})$ are equivalent uniform distributions, so we have:

$$p\left(f \mid D^t \mathcal{A}\right) = \frac{p(D_1 \mid f\mathcal{A})\, p(D_2 \mid f\mathcal{A})}{p(D_1 \mid D_2 \mathcal{A})}$$

Each term in this expression is an example of the model due to a single known point, given in equation 4.1, so we have:

$$p\left(f \mid D^t \mathcal{A}\right) = \frac{\phi(f_1, \alpha d_1, f)\, \phi(f_2, \alpha d_2, f)}{\phi(f_2, \alpha(d_1 + d_2), f_1)}$$

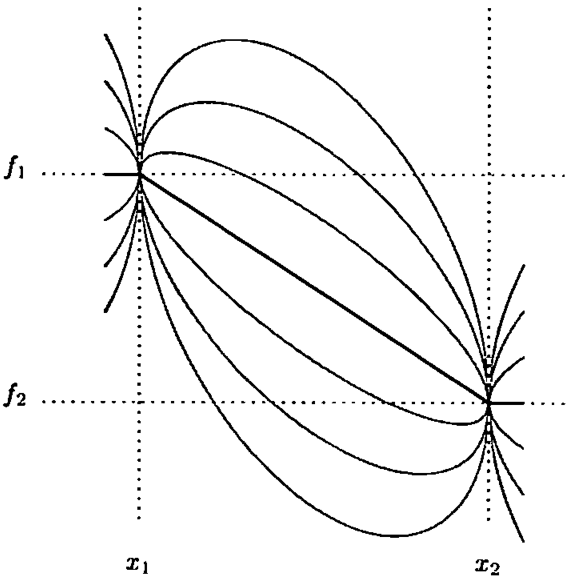


Figure 4.4: The form of the model of a one-dimensional surface produced by the assumption of uniform continuity . At every point x in the space, the probability density function predicted for the performance, f is Gaussian: the bold line shows the mean of the distribution, the fainter lines show ± 1, 2, and 3 standard deviations. Points x_1 and x_2 have been evaluated, yielding performance values of f_1 and f_2, respectively.

which simplifies to:

$$p\left(f \mid D^t \mathcal{A}\right) = \phi\left(\frac{f_1 d_2 + f_2 d_1}{d_1 + d_2}, \alpha \frac{d_1 d_2}{d_1 + d_2}, f\right) \tag{4.2}$$

The performance surface model thus gives a normal distribution as the probability density function for the performance of any point, with the parameters of the distribution dependent only on the nearest evaluated point on either side. The mean of the distribution is linearly interpolated from the two known performance values, and the variance is given by $\alpha \frac{d_1 d_2}{d_1 + d_2}$. Figure 4.4 illustrates the form that this model of the performance surface takes.

For any point x, as depicted in figure 4.3, the probability density for the value of the performance surface at x is given by equation 4.2, with x_1 and x_2 being the evaluated points which are nearest to x on either side. In the case where no points have been evaluated to one side of x, the probability density is given by equation 4.1.

4.4 The Goal Surface Model

(Refer again to figure 4.3 for the notation used.) As discussed in section 3.2, the goal surface model provides for every point x a measure of the probability (given the current state of knowledge about the surface) that the optimization goal is achieved at x. Recall that the goal being addressed is the location of a point x such that $F(x) \geq f^*$, and we denote the hypothesis that $F(x) \geq f^*$ as $U(x)$. For convenience, we also define $\mathcal{U}(x, \tau)$ as the hypothesis that $F(x) = \tau$.

Equation 3.6 shows how to derive the overall goal surface model based on the combination of the effects of all points so far evaluated. As for the derivation of the performance surface model above, the "occlusion" corollary of the assumption of uniform continuity means that for any x, only the effects of the nearest evaluated point on each side of x (x_1 and x_2) need be considered. Equation 3.6 then gives the goal surface model as:

$$M_t(x) = P(U(x) \mid D^2\mathcal{A}) = \frac{P(U(x) \mid D_1\mathcal{A})P(D_2 \mid U(x)D_1\mathcal{A})}{P(D_2 \mid D_1\mathcal{A})}$$

Since D_2 and $D_1U(x)$ are conditionally independent, given $\mathcal{U}(x,\tau)$, this becomes:

$$M_t(x) = \frac{P(U(x) \mid D_1\mathcal{A})}{P(D_2 \mid D_1\mathcal{A})} \int_{-\infty}^{\infty} p(D_2 \mid \mathcal{U}(x,\tau)\mathcal{A}) p(\mathcal{U}(x,\tau) \mid D_1U(x)\mathcal{A}) d\tau$$

Applying Bayes' theorem to the last term in the integral:

$$M_t(x) = \frac{P(U(x) \mid D_1\mathcal{A})}{P(D_2 \mid D_1\mathcal{A})} \int_{-\infty}^{\infty} \frac{p(D_2 \mid \mathcal{U}(x,\tau)\mathcal{A}) p(\mathcal{U}(x,\tau) \mid D_1\mathcal{A}) p(U(x) \mid \mathcal{U}(x,\tau)D_1\mathcal{A})}{P(U(x) \mid D_1\mathcal{A})} d\tau$$

Since D_1 and $U(x)$ are conditionally independent, given $\mathcal{U}(x,\tau)$:

$$M_t(x) = \frac{P(U(x) \mid D_1\mathcal{A})}{P(D_2 \mid D_1\mathcal{A})} \int_{-\infty}^{\infty} \frac{p(D_2 \mid \mathcal{U}(x,\tau)\mathcal{A}) p(\mathcal{U}(x,\tau) \mid D_1\mathcal{A}) p(U(x) \mid \mathcal{U}(x,\tau)\mathcal{A})}{P(U(x) \mid D_1\mathcal{A})} d\tau$$

It is clear from the definitions of $U(x)$ and $\mathcal{U}(x,\tau)$ that $P(U(x) \mid \mathcal{U}(x,\tau)\mathcal{A}) = 0$ for $\tau < f^*$, and $= 1$ for $\tau \geq f^*$, which allows its effect on the expression to be absorbed into the limits of integration, giving:

$$M_t(x) = \frac{P(U(x) \mid D_1\mathcal{A})}{P(D_2 \mid D_1\mathcal{A})} \int_{f^*}^{\infty} \frac{p(D_2 \mid \mathcal{U}(x,\tau)\mathcal{A}) p(\mathcal{U}(x,\tau) \mid D_1\mathcal{A})}{P(U(x) \mid D_1\mathcal{A})} d\tau$$

$$M_t(x) = \frac{1}{P(D_2 \mid D_1\mathcal{A})} \int_{f^*}^{\infty} p(D_2 \mid \mathcal{U}(x,\tau)\mathcal{A}) p(\mathcal{U}(x,\tau) \mid D_1\mathcal{A}) d\tau$$

Each of the probability terms in this expression can be determined from the equation for the performance surface model due to a single evaluation (equation 4.1):

$$M_t(x) = \frac{1}{\phi(f_1, \alpha(d_1+d_2), f_2)} \int_{f^*}^{\infty} \phi(\tau, \alpha d_2, f_2) \phi(f_1, \alpha d_1, \tau) d\tau$$

Making use of the properties of the normal distribution that

$$\phi\left(\mu, \sigma^2, \tau\right) \phi\left(m, s^2, \tau\right) = \phi\left(\mu, \sigma^2 + s^2, m\right) \phi\left(\frac{\mu s^2 + m\sigma^2}{s^2+\sigma^2}, \frac{s^2\sigma^2}{s^2+\sigma^2}, \tau\right)$$

and that $\phi(\mu, \sigma^2, \tau) = \phi(\tau, \sigma^2, \mu)$ gives:

$$M_t(x) = \int_{f^*}^{\infty} \phi\left(\frac{f_1 d_2 + f_2 d_1}{d_1 + d_2}, \alpha\frac{d_1 d_2}{d_1 + d_2}, \tau\right) d\tau$$

or (substituting $d_2 = d_{1,2} - d_1$):

$$M_t(x) = P\left(U(x) \mid D^2\mathcal{A}\right) = 1 - \Phi\left(\frac{f^* - \frac{f_1(d_{1,2}-d_1)+f_2 d_1}{d_{1,2}}}{\sqrt{\alpha\frac{d_1(d_{1,2}-d_1)}{d_{1,2}}}}\right) \tag{4.3}$$

where $\Phi(z)$ is the cumulative normal distribution, $\int_{-\infty}^{z} \phi(0, 1, \tau)\, d\tau$.

The Goal Surface Model

For any point x, as depicted in figure 4.3, the probabilistic goal surface model is given by equation 4.3, with x_1 and x_2 being the evaluated points which are nearest to x on either side. In the case where no points have been evaluated to one side of x, so that evidence is available only from the single evaluation x_1, then the goal surface model is readily determined from equation 4.1 to be:

$$M_t(x) = 1 - \Phi\left(\frac{f^* - f_1}{\sqrt{\alpha d_1}}\right)$$

4.5 Optimization Strategy Implementations

Both the maximum likelihood and the ignorance minimizing strategies require the selection of a point for evaluation based on a calculation of the value of that evaluation to the optimization process. For the maximum likelihood strategy, the measure of value of any point is the assessed probability that the point achieves the goal; for the ignorance minimizing strategy, it is the expected reduction in the ignorance of the goal surface which will result from the point's evaluation. In both cases, the calculation of the measure of value can be made entirely from consideration of the interval between two adjacent evaluated points which contains the point in question.

Both strategies may therefore be approached in the same manner:

1. Partition the design space into intervals defined by the current set of evaluations.
2. For each interval, locate the point within the interval which maximizes the measure of value, and calculate the measure of value for that point.
3. Select for evaluation the point with the highest value located during stage 2, considering all intervals.

The sections below describe how stage 2 above was implemented for each strategy. Recall that the goal addressed is a satisficing one: the location of a point with a performance value which equals or exceeds a given target, f^*.

4.5.1 The Maximum Likelihood Strategy

The maximum likelihood strategy requires the location of the point within each interval at which the assessed probability that the point exceeds the goal is maximized: that is, the value of d_1 at which the expression in equation 4.3 is maximized. Equivalently, we wish to determine d_1 so as to minimize

$$\Phi\left(\frac{f^* - \frac{f_1(d_{1,2}-d_1)+f_2 d_1}{d_{1,2}}}{\sqrt{\alpha\frac{d_1(d_{1,2}-d_1)}{d_{1,2}}}}\right)$$

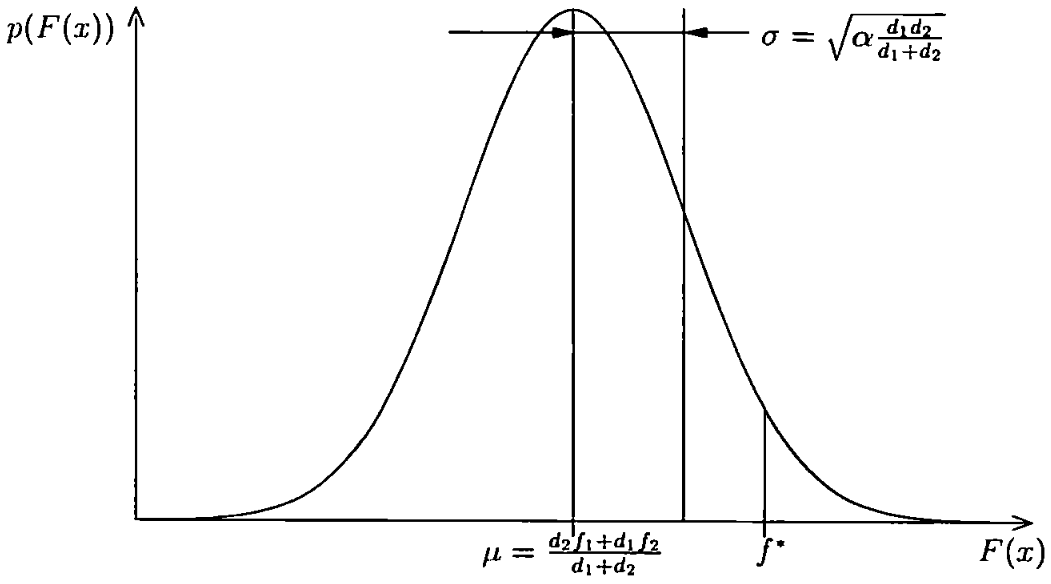


Figure 4.5: The form of the probability density for the performance of a point within an interval such as that illustrated in figure 4.3. The density function is normal, with mean μ and standard deviation σ as shown. The shaded area represents the probability that $F(x) > f^*$.

the value of which is equal to the area of the shaded region in figure 4.5. To maximize this area, we need to minimze the number of standard deviations by which f^* exceeds the mean of the distribution, $\frac{f_1(d_{1,2}-d_1)+f_2d_1}{d_{1,2}}$. That is, we need to locate d_1 so as to minimize

$$\frac{f^* - \frac{f_1(d_{1,2}-d_1)+f_2d_1}{d_1+d_2}}{\sqrt{\alpha\frac{d_1d_2}{d_{1,2}}}}$$

(where $d_1 + d_2 = d_{1,2}$ is constant).

Rearranging this expression gives:

$$\frac{(f^* - f_2)\,d_1 + (f^* - f_1)\,(d_{1,2} - d_1)}{\sqrt{d_1\,(d_{1,2} - d_1)}}\frac{1}{\sqrt{\alpha d_{1,2}}}$$

Substituting for convenience $d_1 = \xi d_{1,2}$, we wish to determine ξ so as to minimize

$$\frac{(f^* - f_1) + (f_1 - f_2)\,\xi}{(\xi - \xi^2)^{\frac{1}{2}}}\frac{1}{\sqrt{\alpha d_{1,2}}}$$

Setting the derivative with respect to ξ of this expression equal to zero yields:

$$\left((f_1 - f_2)\left(\xi - \xi^2\right)^{\frac{1}{2}} - \frac{1}{2}(f^* - f_1 + f_1\xi - f_2\xi)\left(\xi - \xi^2\right)^{-\frac{1}{2}}(1 - 2\xi)\right)\frac{1}{\sqrt{\alpha d_{1,2}}} = 0$$

Multiplying through by $\sqrt{\alpha d_{1,2}\,(\xi - \xi^2)}$:

$$2(f_1 - f_2)\left(\xi - \xi^2\right) - ((f^* - f_1) + (f_1 - f_2)\xi)(1 - 2\xi) = 0$$

Which simplifies to:

$$\xi(2f^* - f_1 - f_2) = f^* - f_1$$

or:

$$\xi = \frac{f^* - f_1}{2f^* - f_1 - f_2}$$

Resubstituting for ξ gives a stationary point within the interval at:

$$d_1 = \frac{f^* - f_1}{2f^* - f_1 - f_2}d_{1,2} \tag{4.4}$$

That this stationary point represents a maximum for the probability value in question can be seen from the form of the model of the interval shown in figure 4.4.

Note that the location of the maximum likelihood point within the interval does not depend on the value of α.

4.5.2 The Ignorance Minimizing Strategy

The Ignorance of an Interval

Referring again to figure 4.3, let $U(x)$ be the hypothesis that $F(x) \geq f^*$, and $d_{1,2} = d_1 + d_2$ the total length of the interval. The model of the goal surface is given by the probability that the performance of x achieves the goal (see equation 4.3):

$$M_\iota(x) = P\left(U(x) \mid D^2\mathcal{A}\right) = 1 - \Phi\left(\frac{f^* - \frac{f_1(d_{1,2}-d_1)+f_2d_1}{d_{1,2}}}{\sqrt{\alpha\frac{d_1(d_{1,2}-d_1)}{d_{1,2}}}}\right)$$

where $\Phi(z)$ is the cumulative normal distribution, $\int_{-\infty}^{z} \phi(0,1,\tau)\, d\tau$.

For convenience, define:

$$\mathcal{H}(q) = -q\log_2 q - (1-q)\log_2(1-q)$$

Then the Shannon entropy of the goal surface model at x is given by:

$$H(x) = \mathcal{H}\left(\Phi\left(\frac{f^* - \frac{f_1(d_{1,2}-d_1)+f_2d_1}{d_{1,2}}}{\sqrt{\alpha\frac{d_1(d_{1,2}-d_1)}{d_{1,2}}}}\right)\right)$$

And the total ignorance of the interval $[x_1, x_2]$ by:

$$I = \int_0^{d_{1,2}} \mathcal{H}\left(\Phi\left(\frac{f^* - \frac{f_1(d_{1,2}-d_1)+f_2d_1}{d_{1,2}}}{\sqrt{\alpha\frac{d_1(d_{1,2}-d_1)}{d_{1,2}}}}\right)\right) dd_1$$

Denoting the quantity $\frac{d_1}{d_{1,2}}$ as ξ, rearranging and changing the variable of integration yields:

$$I = d_{1,2} \int_0^1 \mathcal{H}\left(\left(\frac{f^* - f_1}{\sqrt{\alpha\,(d_{1,2})}} + \frac{f_1 - f_2}{\sqrt{\alpha\,(d_{1,2})}}\xi\right)(\xi\,(1-\xi))^{-\frac{1}{2}}\right) d\xi$$

Defining:

$$\mathcal{I}\,(a, b) = \int_0^1 \mathcal{H}\left((a + b\xi)\,(\xi\,(1-\xi))^{-\frac{1}{2}}\right) d\xi$$

the ignorance of the interval $[x_1, x_2]$ is given by:

$$I = d_{1,2}\mathcal{I}\left(\frac{f^* - f_1}{\sqrt{\alpha d_{1,2}}}, \frac{f_1 - f_2}{\sqrt{\alpha d_{1,2}}}\right) \tag{4.5}$$

It can be seen from equation 4.5 that the ignorance of an interval may be expressed as a function of two quantities characteristic of the interval, $\frac{f^*-f_1}{\sqrt{\alpha d_{1,2}}}$, and $\frac{f_1-f_2}{\sqrt{\alpha d_{1,2}}}$, scaled by the length of the interval, $d_{1,2}$. A tabulation of $\mathcal{I}\,(a, b)$ against values of a and b (each in the range $[-100000, 1000000]$, with logarithmic detail: 100 values in $[0, 1]$, 100 in $[0, 10]$, and so on) was generated by numerical integration. This table was then used to interpolate estimates of the function $\mathcal{I}$. Figure 4.6 shows the form of the function $\mathcal{I}$.

The Expected Ignorance Reduction Due to an Evaluation

Assume that the point x is evaluated, giving a performance value of f. The new evaluation splits the interval $[x_1, x_2]$ into two new intervals, $[x_1, x]$ and $[x, x_2]$. The ignorance of these new intervals is then given, respectively, by:

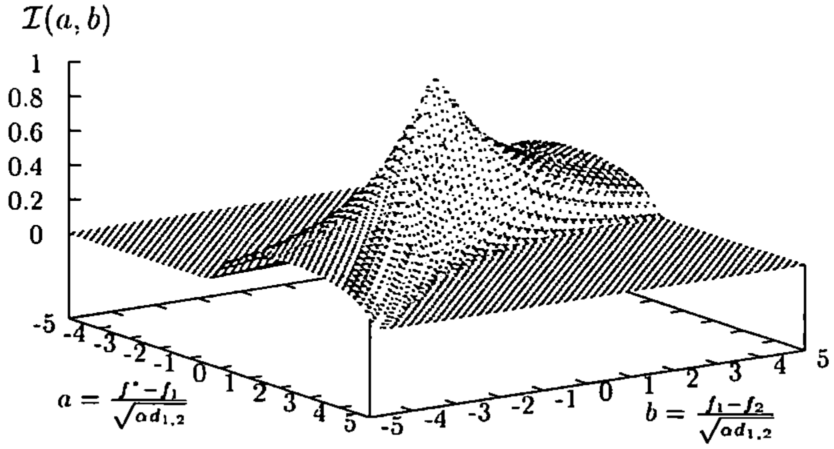


Figure 4.6: *The form of the function $\mathcal{I}$ for calculating the ignorance of an interval.*

$$d_1\mathcal{I}\left(\frac{f^*-f_1}{\sqrt{\alpha d_1}}, \frac{f_1-f}{\sqrt{\alpha d_1}}\right)$$

and

$$d_2\mathcal{I}\left(\frac{f^*-f}{\sqrt{\alpha d_2}}; \frac{f-f_2}{\sqrt{\alpha d_2}}\right)$$

The total expected ignorance for both new intervals following evaluation of x is therefore:

$$d_1\int_{-\infty}^{\infty} p(f)\,\mathcal{I}\left(\frac{f^*-f_1}{\sqrt{\alpha d_1}}; \frac{f_1-f}{\sqrt{\alpha d_1}}\right) df + d_2\int_{-\infty}^{\infty} p(f)\,\mathcal{I}\left(\frac{f^*-f}{\sqrt{\alpha d_2}}, \frac{f-f_2}{\sqrt{\alpha d_2}}\right)$$

and the expected reduction in ignorance resulting from the evaluation:

$$d_{1,2}\mathcal{I}\left(\frac{f^*-f_1}{\sqrt{\alpha d_{1,2}}}, \frac{f_1-f_2}{\sqrt{\alpha d_{1,2}}}\right) - d_1\int_{-\infty}^{\infty} p(f)\,\mathcal{I}\left(\frac{f^*-f_1}{\sqrt{\alpha d_1}}, \frac{f_1-f}{\sqrt{\alpha d_1}}\right) df$$

$$- d_2\int_{-\infty}^{\infty} p(f)\,\mathcal{I}\left(\frac{f^*-f}{\sqrt{\alpha d_2}}, \frac{f-f_2}{\sqrt{\alpha d_2}}\right)$$

For convenience, we adopt performance values normalized with respect to the length of the original interval and the value of α, and without loss of generality adopt f_1 as the origin of the performance measurement:

$$\begin{aligned}
\lambda &= \frac{f}{\sqrt{\alpha d_{1,2}}} \\
\lambda^* &= \frac{f^*}{\sqrt{\alpha d_{1,2}}} \\
\lambda_1 &= \frac{f_1}{\sqrt{\alpha d_{1,2}}} = 0 \\
\lambda_2 &= \frac{f_2}{\sqrt{\alpha d_{1,2}}}
\end{aligned}$$

Which yields an expression for the total expected ignorance reduction resulting from the evaluation of x:

$$d_{1,2}\left(\mathcal{I}(\lambda^*, -\lambda_2) - \xi\int_{-\infty}^{\infty} p(\lambda)\,\mathcal{I}\left(\frac{\lambda^*}{\xi}, -\frac{\lambda}{\xi}\right) d\lambda\right.$$

$$\left. - (1-\xi)\int_{-\infty}^{\infty} p(\lambda)\,\mathcal{I}\left(\frac{\lambda^*-\lambda}{1-\xi}, -\frac{\lambda-\lambda_2}{1-\xi}\right) d\lambda\right) \tag{4.6}$$

Equation 4.6 gives an expression for calculating the expected reduction in ignorance resulting from the evaluation of any point within a particular interval, in terms of two parameters, $\lambda^* = \frac{f^*}{\sqrt{\alpha d_{1,2}}}$ and $\lambda_2 = \frac{f_2}{\sqrt{\alpha d_{1,2}}}$, which are characteristic of the interval, and the value of ξ, which defines the location within the interval of the point evaluated (as a fraction of the interval length, $d_{1,2}$, measured from the point x_1). The expected reduction in ignorance for any interval can be determined by calculating the corresponding reduction for the "similar" interval (having the same values of λ^* and λ_2) of unit length, and scaling by the actual length of the interval, $d_{1,2}$.

Determination of the point of maximum expected ignorance reduction for an interval requires the maximization of the expression in equation 4.6 over $\xi \in [0, 1]$. Tables of the maximum expected ignorance reduction attainable, and of the corresponding maximizing values of ξ, against values of λ^* and λ_2 were generated numerically for intervals of unit length (values of λ^* and λ_2 tabulated were in the range $[-1000, 1000]$, with logarithmic detail as for the function $\mathcal{I}$). Each maximization was performed by simply calculating the expected ignorance reduction at 99 equally-spaced points within the interval. The resulting tables allowed the interpolation of the location of the point of maximum expected ignorance reduction and the value of the expected reduction at that point for the unit interval "similar" to any given interval. The actual location and value for the given interval can then be obtained simply by scaling the results by the length of the interval.

4.5.3 Implementation Notes

Extremes of the Space

The above analysis has described the implementation of methods for locating the point of maximum value within a given interval, and determining the corresponding measure of value, for both the maximum likelihood and ignorance minimizing strategies. In cases where one or both of the extreme points of the design space have not been evaluated, then the interval at that end of the space is defined by one point only. For the maximum likelihood strategy, in this case the maximum value is attained at the unevaluated extreme of the interval. For the ignorance minimizing strategy, an identical procedure to that presented above (but simpler, there being only a single constraining point) was used to generate a table for estimating the point of maximum expected ignorance in such cases.

Implementation Loop

Given the capacity to determine the optimum point for each strategy within any interval, implementation of the strategies reduces to the maintainance of a sorted list of the intervals into which the design space is currently divided, and repeating the following stages until the goal of the optimization is achieved:

1. Select for evaluation the point on the list of intervals with the maximum assessed value. Evaluate it.
2. Remove from the list the details of the existing interval within which the evaluated point lies.
3. Calculate the point of maximum value, and the corresponding value, for both the new intervals formed by the new evaluation, and add these details to the list.

The Model Parameter α

In order to implement the ignorance minimizing strategy, it was necessary to parameterize the assumption of uniform continuity with the model parameter α. In determining the rate of change of the variance of the model of the performance surface with distance from an evaluated point, α is clearly related to expectations about the magnitudes of gradients on the surface. The experimentation in chapter 5 makes some investigation into the sensitivity of the ignorance minimizing strategy to the value chosen for α.

Complexity

The determination of the point of maximum expected ignorance reduction within an interval involves nested integration of analytically intractable expressions. The recognition that the results for an interval of unit length could be tabulated, and then scaled to give the result for any given interval avoided the necessity for expensive numerical integration during the optimization process. The tables took approximately 10 days of computer time to generate, but could subsequently be used across multiple optimization processes, enabling the assessment of an interval using the ignorance maximizing strategy to be achieved very cheaply. Use of the tables was tested against a specific numerical integration for 100 randomly constructed intervals, and found to be acceptably accurate.

Chapter 5

A Comparison of Optimization Strategies

5.1 Introduction

Previous chapters have introduced a viewpoint from which to approach the optimization of design spaces through the construction of probabilistic models of the goal and performance surfaces for the problem. Three strategies have been proposed for carrying out optimization based on such models:

1. The *maximum likelihood* strategy: an exploitative strategy whereby the point in the space with the highest assessed probability of achieving the optimization goal is always selected as the next point for evaluation.

2. The *ignorance minimizing* strategy: an exploratory strategy where the point selected for evaluation is that which is expected maximally to reduce the overall level of uncertainty concerning the locations and distribution of goal-satisfying points across the design space.

3. The *composite* strategy, which consists of applying the ignorance minimizing strategy in the early stages of the search, and switching to the maximum likelihood strategy only for selection of the final point to evaluate.

This chapter describes experimentation aimed at investigating the behaviours of, and qualitative differences between these strategies. The investigation uses simple one-dimensional design and performance spaces, a satisficing goal, and adopts the assumptions, models and optimization strategy implementations developed in chapter 4.

Section 5.2 describes the aims of the experimentation conducted, and the hypotheses investigated, while section 5.3 describes the experiments performed. Results are presented in section 5.4, and discussed in terms of the hypotheses proposed. An overall discussion of the findings of the experimentation is presented in section 5.5.

5.1.1 The Composite Strategy: "Potential" Efficiency

While the composite strategy has been proposed as the most appropriate optimization strategy to use in practice, no principle has yet been suggested for deciding *when* to switch from the ignorance minimizing strategy to the maximum likelihood strategy. Consequently, the experimentation in this chapter addresses the *potential efficiency* of optimization using the composite strategy. For every optimization process run using the ignorance minimizing strategy, the point which *would* have been selected at each stage by the maximum likelihood strategy was recorded. This enables consideration of the *potential efficiency* (also termed *potential time-to-solution* below) of the composite strategy: when *might* a point satisfying the optimization goal have been found, assuming that the decision of when to switch between the ignorance minimizing and maximum likelihood strategies could have been optimally made?

Note that the time-to-solution of the composite strategy has an upper limit determined by the operation of the ignorance minimizing strategy: if a point which achieves the optimization goal is evaluated, then that point has subsequently a probability of 1 of achieving the goal, and will therefore be the next point selected by the maximum likelihood strategy. The potential efficiency for the composite strategy cannot therefore be worse by more than one evaluation than the efficiency of the ignorance minimizing strategy under the same conditions.

5.2 Aims and Hypotheses

5.2.1 Aims

A number of predictions were made in chapter 3 concerning the expected behaviour of the optimization strategies identified. The experimentation described in this chapter was conducted in order to test these predictions. To this end, a number of hypotheses were investigated, relating to aspects of the strategies' possible behaviours. The hypotheses investigated are listed below.

In addition, it was desired to investigate qualitatively any differences which might be observed between the operations of the different strategies, and the level of sensitivity of the ignorance minimizing strategy to the value of the model parameter, α (defined in chapter 4).

5.2.2 Hypotheses

Reliability of Solution

1. Both the maximum likelihood strategy and the ignorance minimizing strategy will reliably find a solution in all cases where a solution is possible.

Both strategies were characterized in chapter 3 as non-convergent, with the exception of the maximum likelihood strategy in cases where the goal has already been achieved. It is therefore expected that until the optimization goal is achieved, both strategies will continue to achieve wide coverage of the surface, leading to the reliable location of a point which achieves the goal in every case in which such a point exists.

Efficiency of Solution

1. The composite strategy will outperform both of the individual strategies of which it is composed, given the same optimization problem.

Neither of the individual strategies, if used in isolation, has the characteristics required for efficient optimization. The maximum likelihood strategy has no exploratory behaviour, and does not set out to develop the information about the goal surface necessary for optimization to be efficient. The ignorance minimizing strategy does develop information in this way, but does not exploit it actually to achieve the goal: it never selects a point for evaluation because that point is likely to achieve the goal, and thus if use of the strategy ever *does* achieve the goal, it does so purely serendipitously. It is the combination of these two strategies represented by the composite strategy which is expected to be the most effective optimizer.

Allocation of Trials and Convergence

In order to investigate the manners in which the different strategies distribute evaluations in the design space over time, some of the optimization experiments run were given goals which were not actually achievable anywhere within the design space. It was felt that these cases would give a better view of the strategies' longer-term behaviour in a "difficult" space. For these cases, the hypotheses made were:

1. Both the maximum likelihood and ignorance minimizing strategies will allocate trials to all regions of the surface, with a greater density in those regions which most nearly attain the goal.
2. More difficult goals (i.e. higher performance targets) will lead to a lowering of the density of trials allocated within higher-performance regions, and a corresponding rise in the density of trials in lower-performing regions.
3. The ignorance minimizing strategy will not produce convergence. The maximum likelihood strategy will converge, but only on an evaluated point which achieves the optimization goal, once such a point is found.
4. All local optima of the multimodal surfaces will be investigated when the goal set for the optimization is not actually achievable.

5.3 Method

In pursuit of the aims described above, a number of optimization processes were run under varying conditions. The factors which were varied were as follows:

Performance Function. Three different performance functions were used, and are shown in figures 5.1–5.3. The functions used were:

$F1$: A unimodal function:

$$F(x) = \sin\left(\frac{3\pi}{2}x^2\right), \qquad 0.0 \leq x \leq 1.0 \tag{5.1}$$

$F2$: A multimodal function with 5 equivalent optima:

$$F(x) = \sin^6(5\pi x), \qquad 0.0 \leq x \leq 1.0 \tag{5.2}$$

$F3$: A multimodal function with a single global optimum and several lower-performance local optima:

$$F(x) = e^{-2\log 2\left(\frac{x-0.1}{0.8}\right)^2} \sin^6(5\pi x), \qquad 0.0 \leq x \leq 1.0 \tag{5.3}$$

$F2$ and $F3$ are similar to the multimodal functions used by Goldberg and Richardson [41] to investigate GA niching mechanisms. $F3$ was used by Bilchev [9] in investigating the optimization of continuous spaces using the ant colony metaphor.

Optimization Goal. A simple satisficing goal was adopted, taking the form of a target performance value, f^*. The aim of each optimization process was thus to locate a point with a performance equal to or exceeding the given target. A range of target values was used, including some which were not in fact attainable. These unachievable goals were included to investigate the longer-term behaviour of the optimization strategies used in a "difficult" space. These cases were expected to be more revealing about the long-term exploratory, exploitative and convergent behaviour of the strategies. All the performance functions used have global optima with a performance value of 1.0. The same set of target perfor-

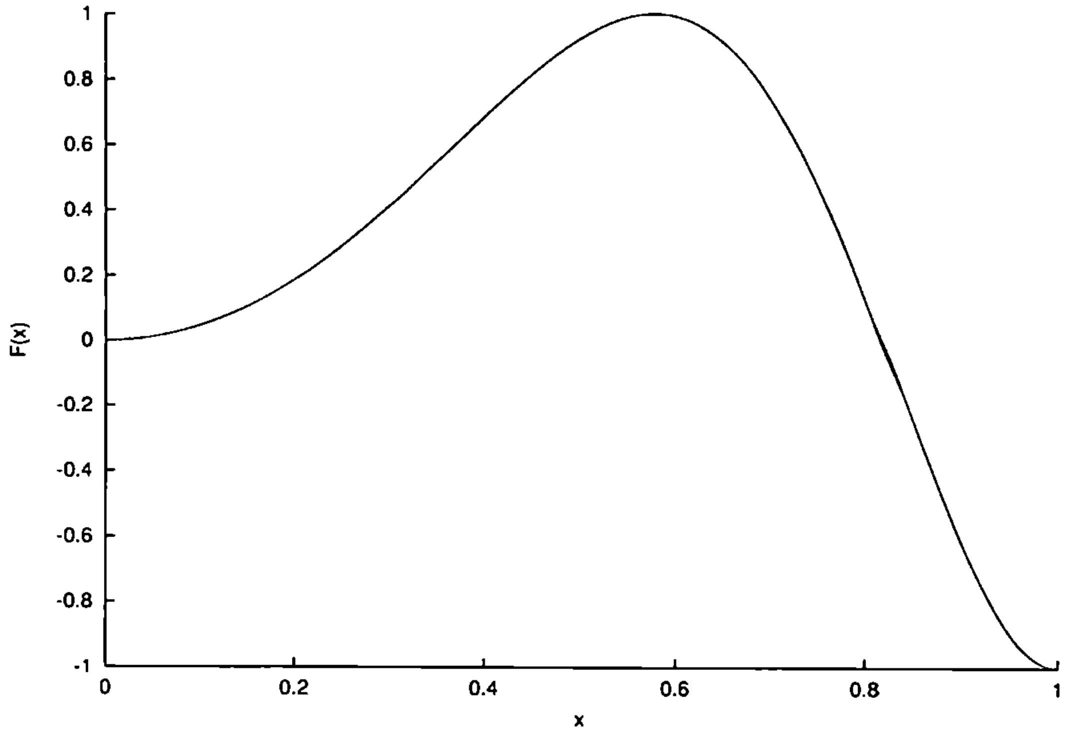


Figure 5.1: *The performance function $F1$: a unimodal function.*

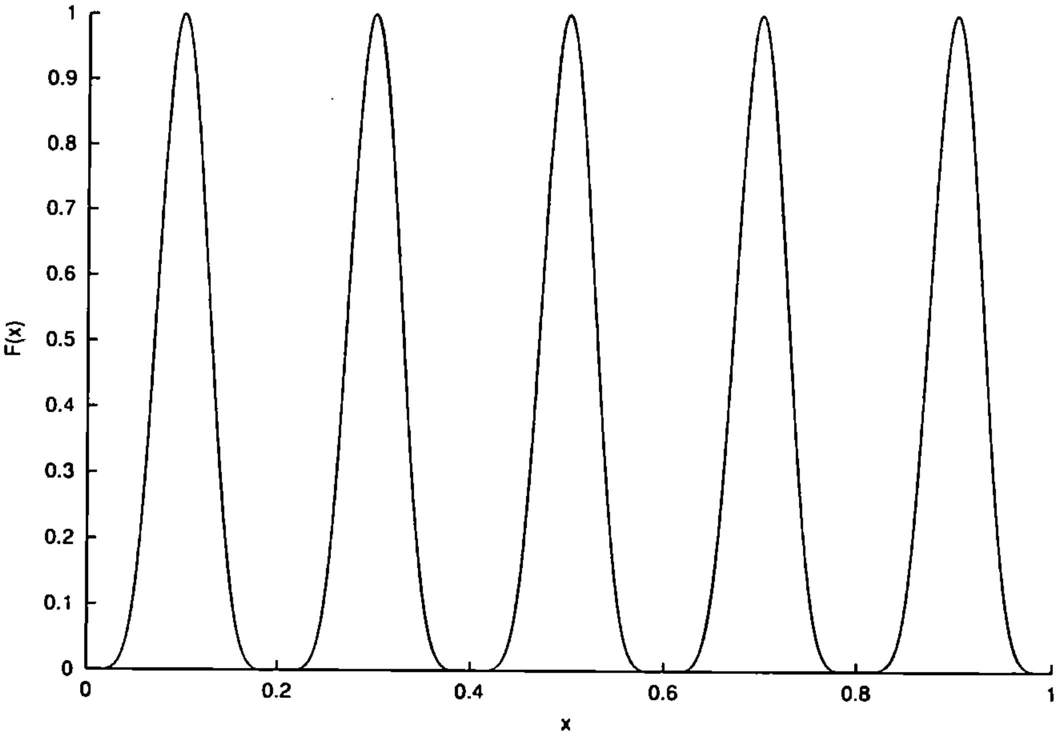


Figure 5.2: *The performance function $F2$: a multimodal function with equal optima.*

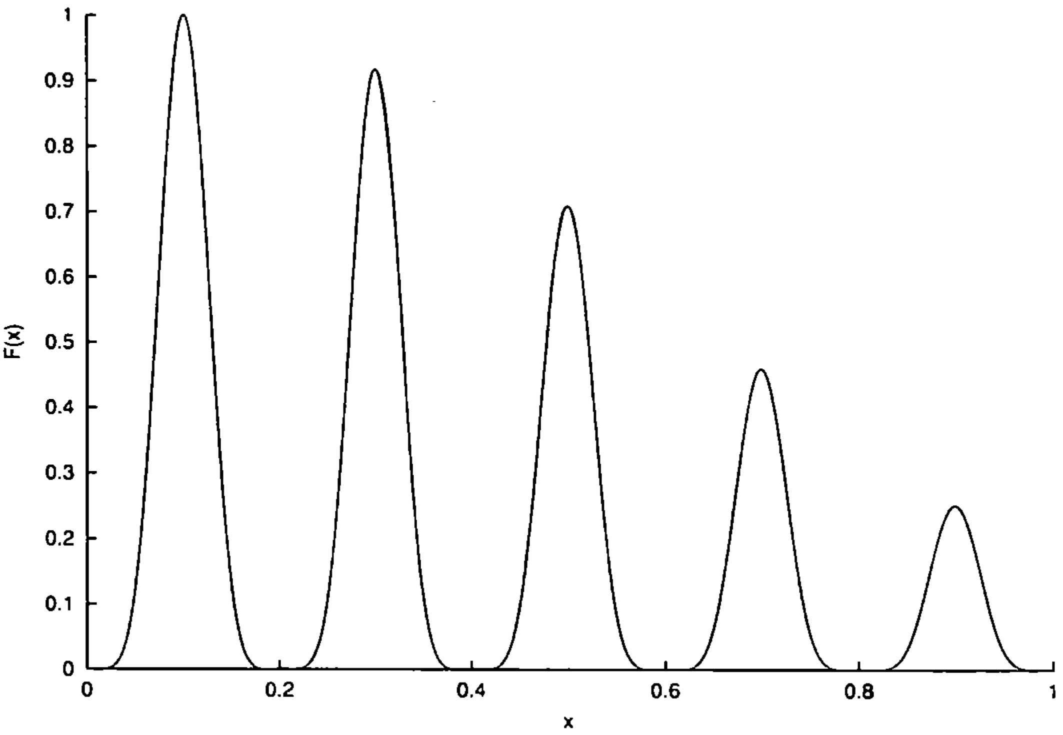


Figure 5.3: The performance function F3: a multimodal function with unequal optima.

mance values was used for each function, namely $f^* = 0.8, 0.9, 0.98, 0.99, 0.999, 1.001, 1.01, 1.1, 1.2$.

Optimization Strategy. For each optimization, either the maximum likelihood strategy or the ignorance minimizing strategy was used. For optimizations conducted using the ignorance minimizing strategy, the point which would have been selected at each stage by the maximum likelihood strategy was also recorded, although not used in the optimization process, to allow investigation of the potential efficiency of the composite strategy, as described above.

Model Parameter, α. Use of the ignorance minimizing strategy required specification of the parameter α for the performance model. This parameter encapsulates the assumptions made when generating the performance surface model about the relationship between the distance between any two points and the possible magnitude of the difference in their performances: that is, α represents the beliefs captured in the model about the expected magnitudes of gradients on the performance surface. For the three surfaces investigated, the maximum gradient magnitude is known, and this value therefore represents a sensible upper bound for α: for $F1$, $\alpha_{max} \approx 10$; for $F2$ and $F3$ $\alpha_{max} \approx 25$. For each function, the values of α investigated were 0.2, 0.4, 0.6, 0.8 and 1.0 times the relevant upper bound: that is, for $F1$, $\alpha = 2, 4, 6, 8, 10$, and for $F2$ and $F3$, $\alpha = 5, 10, 15, 20, 25$.

For each possible combination of function, goal, strategy, and (where the strategy was the ignorance minimizing strategy) α, 10 optimizations were run—in all, 1620 optimization processes.

For each run, the first point for evaluation was chosen uniformly randomly from the design space. While the random choice of a first point is mandated by the maximum likelihood strategy, it can be shown that the ignorance minimizing strategy dictates that the first point evaluated in every case should be the mid-point of the design space. It was decided that it would be more illuminating to view the behaviour of the ignorance minimizing strategy given a range of different starting points. For this reason (and also to remove the possibility of a serendipitous high performance of the strategy on a particular function, based on the strategy starting always from the same point) the first point for evaluation was chosen randomly in every case.

Each optimization proceeded either until the goal was attained, or until 1000 evaluations had been performed. The limit of 1000 evaluations was chosen as being well in excess of the expected number of evaluations required by a random search process to achieve success on the most difficult combination of performance function and feasible goal: $F3$ and $f^* = 0.999$. $F3(x)$ exceeds this target value within approximately the range $(0.098839, 0.101161)$. This represents a fraction of $0.101161 - 0.098839 = 0.002322$ of the design space (the range $[0,1]$). Random search can therefore be expected to take $0.002322^{-1} \approx 430$ evaluations before a point in this region is selected for evaluation. It was therefore thought reasonable to judge an optimization to have failed after 1000 evaluations—somewhat over twice the number expected to be required in random search.

Details of the progress of each optimization were stored in a database, which was then queried to generate the tables and graphs in the results presented below.

5.4 Results

5.4.1 Reliability of Solution

Hypothesis

The hypothesis made concerning the reliability of optimization using the different strategies was:

1. Both the maximum likelihood strategy and the ignorance minimizing strategy will reliably find a solution in all cases where a solution is possible.

Results

Tables 5.1(a–c) show the results obtained for each performance function on trials with achievable goals. It can be seen from these tables that the ignorance-minimizing strategy exhibits 100% reliability: in all trials in which a satisfactory solution was possible,

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 2$	10	10	10	10	10
$\alpha = 4$	10	10	10	10	10
$\alpha = 6$	10	10	10	10	10
$\alpha = 8$	10	10	10	10	10
$\alpha = 10$	10	10	10	10	10
likelihood	10	3	3	3	1

(a) $F1$

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 5$	10	10	10	10	10
$\alpha = 10$	10	10	10	10	10
$\alpha = 15$	10	10	10	10	10
$\alpha = 20$	10	10	10	10	10
$\alpha = 25$	10	10	10	10	10
likelihood	10	10	9	6	2

(b) $F2$

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 5$	10	10	10	10	10
$\alpha = 10$	10	10	10	10	10
$\alpha = 15$	10	10	10	10	10
$\alpha = 20$	10	10	10	10	10
$\alpha = 25$	10	10	10	10	10
likelihood	10	9	8	2	1

(c) $F3$

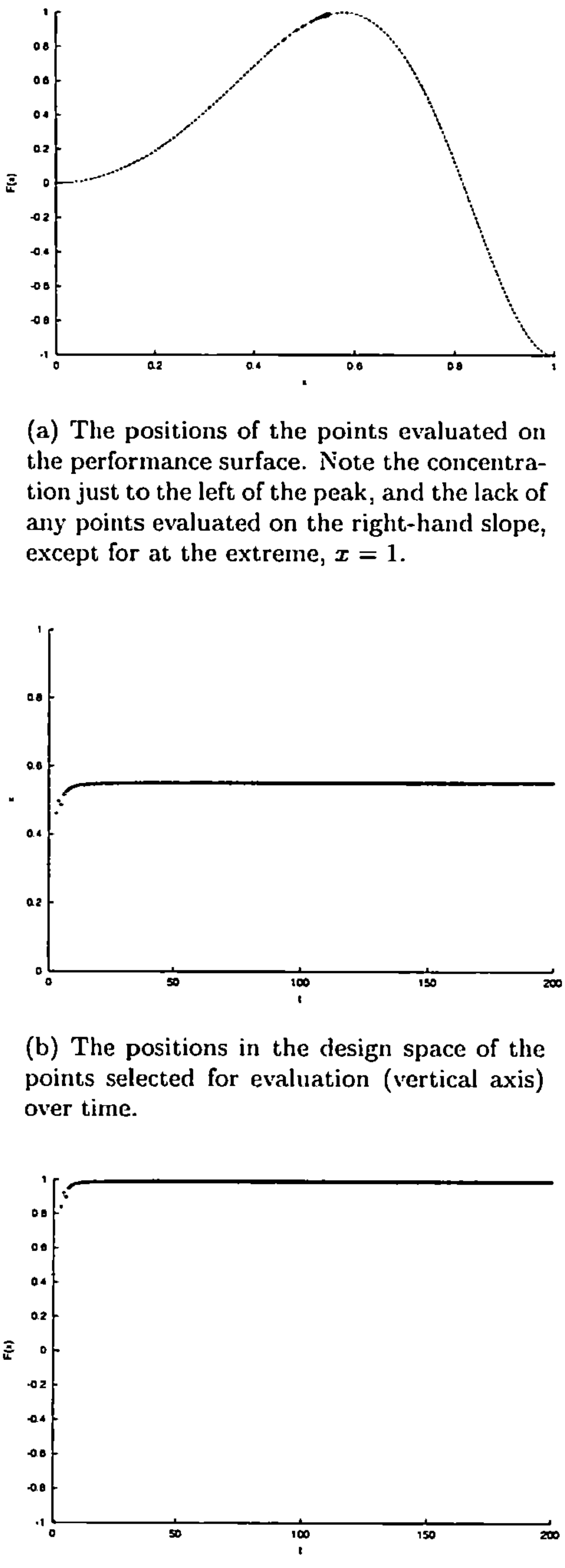
Table 5.1: *Reliability of solution. (a–c) show the results obtained for each performance function. Each table breaks the results down by performance target and optimization strategy, breaking down the ignorance-minimizing strategy results further by the value of the model parameter, α, used. Each cell shows the number of trials, out of a total of 10, in which a satisfactory solution was found. Only cases where the fitness target was attainable are shown.*

one was obtained. By contrast, the maximum likelihood strategy can be seen to fail to find a solution reliably on all functions, for all but the easiest goals (i.e. the lowest target performance values).

Discussion

The failure of the maximum likelihood strategy was surprising. In many cases (the majority of cases for the more difficult goals), no solution was found within the 1000 evaluation limit. Further investigation showed that the optimization in each case had converged at a point within the optimum-containing peak, but below-target performance. That is, the convergence encountered was not even at a local optimum, but at a sub-optimal point on the peak containing the true optimum. In fact, successively selected points assymptotically approached (without ever reaching) the target performance. Figure 5.4 shows a typical case for $F1$.

Further investigation indicated that as successive selections became closer together in the design space, and the performance attained neared the target, the limit of representation for floating-point numbers on the computer architecture used for the experiments was reached, leading to the selection of the same point for evaluation repeatedly. This is a fault of implementation, rather than of the optimization strategy itself, and the results therefore do not indicate whether the target performance would eventually be attained if the floating-point representation limitations did not apply. However, it is clear that even if the target were eventually reached, it would only be after some considerable time: the floating-point effect did not start to take hold in the case shown in figure 5.4 until approximately 150 evaluations had been performed, whereas the maximum number of evaluations taken to reach the goal for the equivalent trials using the ignorance minimizing strategy was 13, the mean 7.34.



(a) The positions of the points evaluated on the performance surface. Note the concentration just to the left of the peak, and the lack of any points evaluated on the right-hand slope, except for at the extreme, $x = 1$.

(b) The positions in the design space of the points selected for evaluation (vertical axis) over time.

(c) The performances of the points selected for evaluation over time.

Figure 5.4: *An illustrative example of the failure of the maximum likelihood strategy to achieve the optimization goal. Results are shown for optimization of $F1$, using the maximum likelihood strategy and given a performance target $f^* = 0.99$. Results for the first 200 evaluations only are shown.*

Each case in which the maximum likelihood strategy failed to achieve the goal was found to share the characteristics of the case shown in figure 5.4. Specifically, convergence occurred within an interval defined by two neighbouring evaluated points which spanned the optimum, where one of the points had low performance while the performance of the other was approaching the target.

Considering the behaviour of the maximum likelihood strategy as defined by equation 4.4, it is clear that as the performance of the better of the two points defining an interval approaches the target performance, then the point selected for evaluation approaches the existing evaluated point. Whether this will lead to asymptotic behaviour such as that observed will depend on the position and performance of the other point defining the interval, and the local gradient of the surface near the better point. However, given the rate of failure experienced even in the experiments for the lower fitness targets, it would seem reasonable to expect such convergence to be not uncommon.

5.4.2 Efficiency of Solution

Hypothesis

The hypothesis relating to the efficiencies of the different strategies stated:

1. The composite strategy will outperform both of the individual strategies of which it is composed, given the same optimization problem.

The failure of the maximum likelihood strategy to find solutions reliably precludes it from being considered "efficient" in any sense, so comparison will only be made between the ignorance minimizing strategy efficiency and the potential efficiency of the composite strategy.

In addition to the above concerns, it was desired to examine the sensitivity of the efficiency of the ignorance minimizing strategy to the value of the model parameter, α.

Results

Tables 5.2 and 5.3 show the mean and maximum time-to-solution, respectively, for the optimization trials using the ignorance minimizing strategy. Tables 5.4 and 5.5 show the same information for the potential time-to-solution for the composite strategy (see section 5.1.1 for the meaning of "potential" time-to-solution).

For each function, figures 5.5–5.7 plot the mean time-to-solution for the maximum ignorance strategy agains the mean potential time-to-solution for the composite strategy.

It is clear from these tables and figures that there is an increase in time-to-solution with increasing performance target. This is entirely what would be expected; more interesting is the apparent consistency with which the potential efficiency of the composite strategy improves upon the efficiency of the ignorance minimizing strategy alone, for functions $F2$ and $F3$. There is no such consistency visible in the results for $F1$, however.

There is no apparent effect of the value used for α on the efficiency of either of the strategies.

Discussion

The reasoning for the hypothesis above was that the composite strategy is the "rational" strategy for optimization, since it integrates exploration and exploitation, while the individual strategies each exhibit only one of these types of behaviour. Any point selected for evaluation by the ignorance minimizing strategy which turns out to achieve

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 2$	4.1	5.3	6.8	5.9	8.2
$\alpha = 4$	2.3	4.4	4.4	7.6	13.4
$\alpha = 6$	3.7	3.5	6.2	7.8	13.5
$\alpha = 8$	3	3.9	7.2	6.5	16.9
$\alpha = 10$	2.5	3.9	6.3	8.9	14.2

(a) $F1$

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 5$	5.8	7	15.3	22	21.5
$\alpha = 10$	6.1	9.6	21.5	19.8	41
$\alpha = 15$	5.2	8.6	16.7	33.6	34.9
$\alpha = 20$	5.6	9.3	18.4	25	36.7
$\alpha = 25$	5.9	9.9	16.7	19.4	40.6

(b) $F2$

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 5$	17.7	20.5	26.2	26	29
$\alpha = 10$	13.1	12.6	20.3	25.9	33.8
$\alpha = 15$	10.8	15.7	22.5	27.1	41.7
$\alpha = 20$	11.4	14.4	25.5	28.7	45.5
$\alpha = 25$	12.7	20.7	24.6	29.1	51.1

(c) $F3$

Table 5.2: *Mean time-to-solution for the ignorance minimizing strategy.*

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 2$	6	7	9	9	15
$\alpha = 4$	4	7	7	10	21
$\alpha = 6$	5	6	9	11	21
$\alpha = 8$	6	6	11	12	24
$\alpha = 10$	6	6	12	13	28

(a) $F1$

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 5$	18	19	22	33	29
$\alpha = 10$	12	23	28	36	52
$\alpha = 15$	11	23	31	40	53
$\alpha = 20$	16	21	30	33	67
$\alpha = 25$	11	22	32	33	52

(b) $F2$

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 5$	25	27	35	34	35
$\alpha = 10$	20	23	25	29	43
$\alpha = 15$	16	24	31	34	50
$\alpha = 20$	24	25	31	37	63
$\alpha = 25$	26	31	35	41	66

(c) $F3$

Table 5.3: *Maximum time-to-solution for the ignorance minimizing strategy.*

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 2$	2.6	4.3	5.2	5.1	7.9
$\alpha = 4$	1.9	4.8	4	6.1	8.1
$\alpha = 6$	1.2	2.8	5.5	6.4	9.4
$\alpha = 8$	2.7	4.3	4.5	5.9	10.1
$\alpha = 10$	1.8	3.8	4.1	6.8	7.7

(a) $F1$

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 5$	2.5	4.6	7.1	10.7	18.7
$\alpha = 10$	5.7	3.9	10.5	13	27.7
$\alpha = 15$	3.7	6	10	18	27.3
$\alpha = 20$	2.6	6.5	18.6	15.5	21.9
$\alpha = 25$	3.6	7.3	10.4	12.6	33.8

(b) $F2$

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 5$	13.7	16.9	22	14.9	24.8
$\alpha = 10$	10.5	9.6	16.4	22.3	21.1
$\alpha = 15$	9.6	10.5	13.9	21.2	31.5
$\alpha = 20$	6.9	12	16.7	20.8	32.6
$\alpha = 25$	7.3	13.1	14.3	16.6	38.7

(c) $F3$

Table 5.4: *Mean potential time-to-solution for the composite strategy.*

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 2$	6	7	9	7	16
$\alpha = 4$	3	7	6	9	14
$\alpha = 6$	2	5	7	9	15
$\alpha = 8$	5	6	8	9	17
$\alpha = 10$	4	6	7	11	13

(a) $F1$

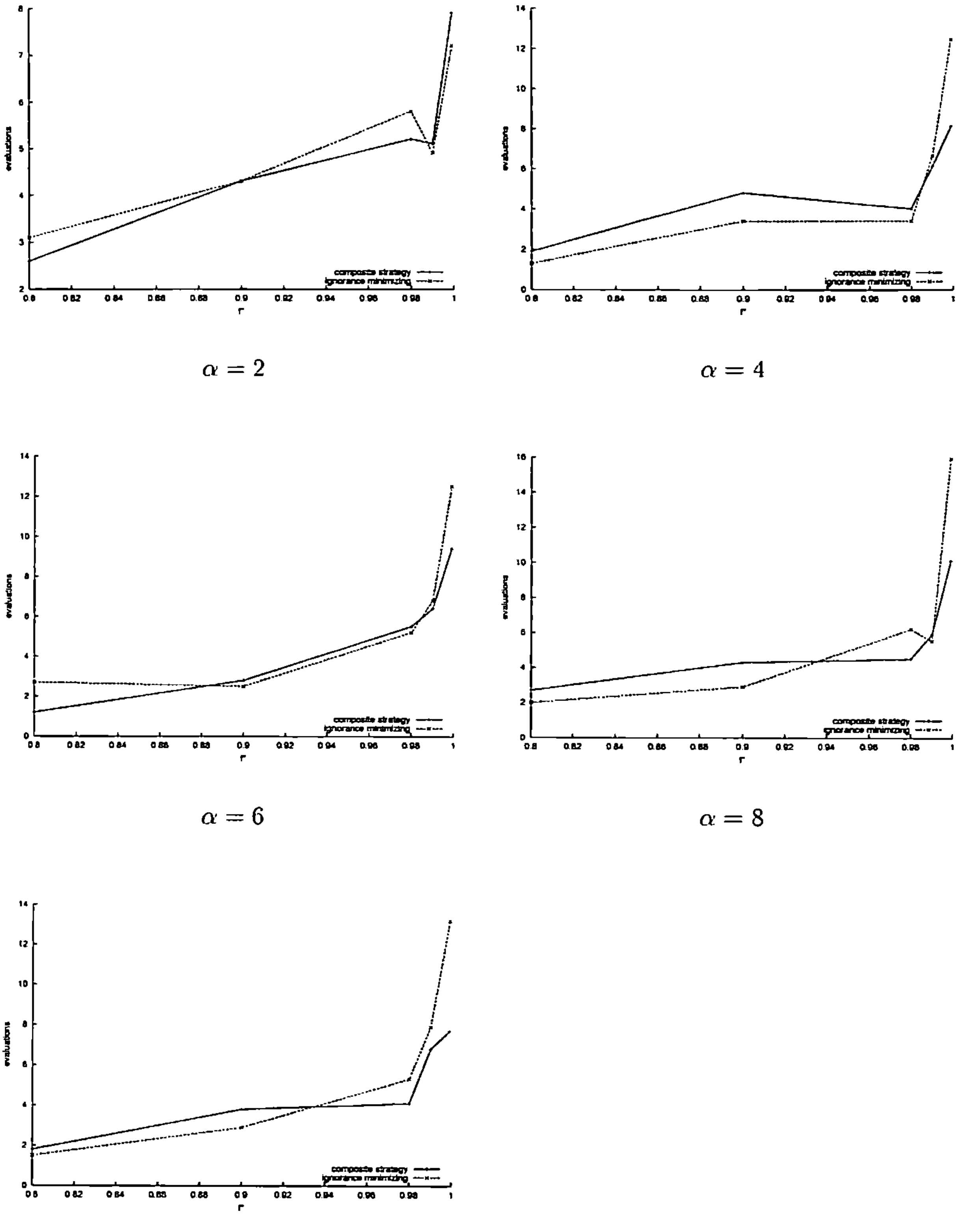
	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 5$	4	8	20	22	27
$\alpha = 10$	13	13	27	28	47
$\alpha = 15$	7	15	32	34	41
$\alpha = 20$	6	12	31	31	66
$\alpha = 25$	10	11	20	31	51

(b) $F2$

	f^*				
	0.8	0.9	0.98	0.99	0.999
$\alpha = 5$	25	25	32	32	31
$\alpha = 10$	13	12	24	29	35
$\alpha = 15$	13	12	27	28	42
$\alpha = 20$	14	19	29	31	48
$\alpha = 25$	15	19	26	28	54

(c) $F3$

Table 5.5: *Maximum potential time-to-solution for the composite strategy.*



$\alpha = 2$

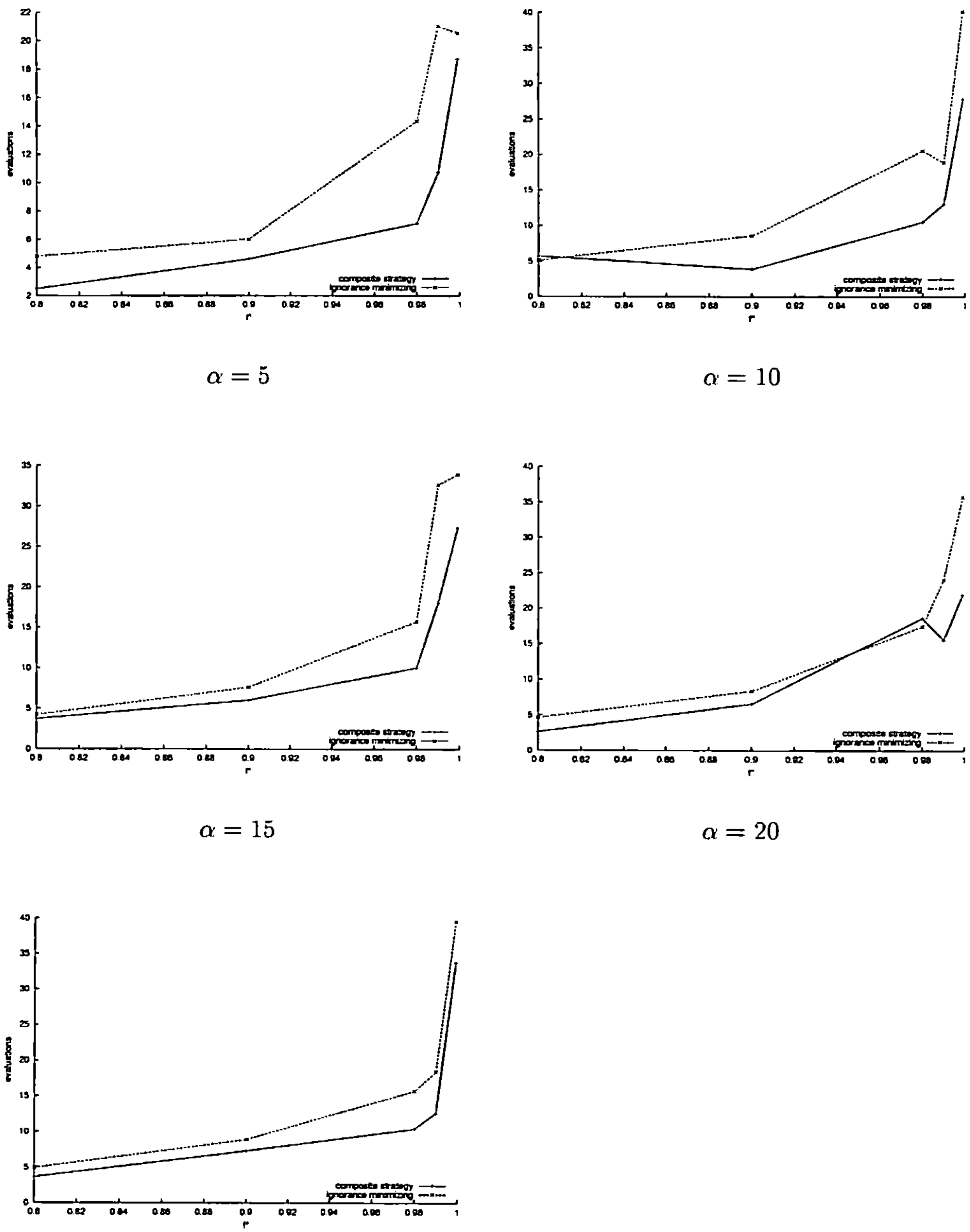
$\alpha = 4$

$\alpha = 6$

$\alpha = 8$

$\alpha = 10$

Figure 5.5: *A comparison of the efficiency of optimizations of $F1$ using the ignorance minimizing strategy and the composite strategy. The performance target f^* is plotted on the horizontal axis, time-to-solution on the vertical. Each point plotted represents the mean time-to-solution for the 10 optimizations conducted for the given values of f^* and α. The composite strategy results show the mean* potential time-to-solution, *as described in section 5.1.1.*



$\alpha = 5$

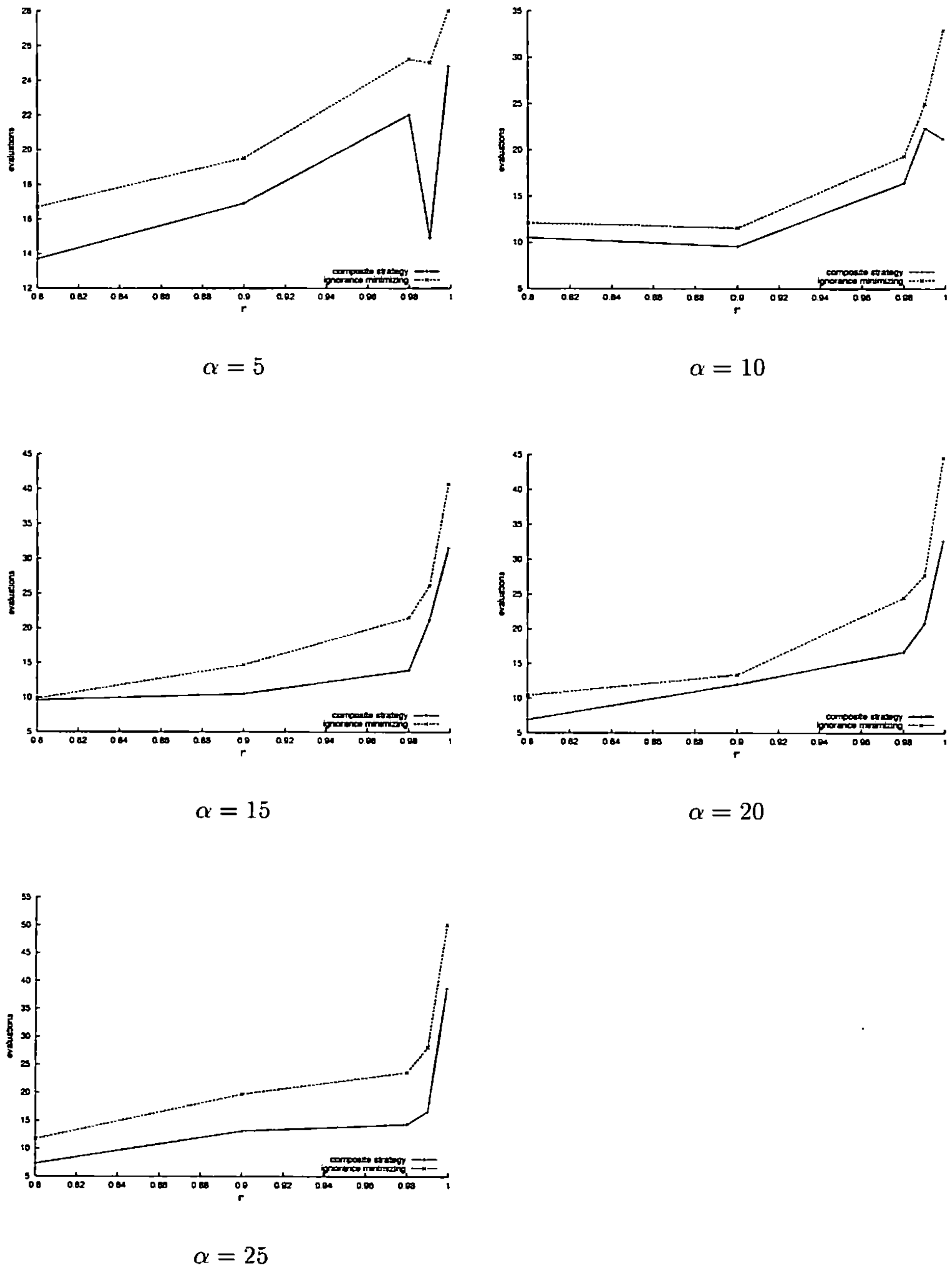
$\alpha = 10$

$\alpha = 15$

$\alpha = 20$

$\alpha = 25$

Figure 5.6: A comparison of the efficiency of optimizations of $F2$ using the ignorance minimizing strategy and the composite strategy. The performance target f^* is plotted on the horizontal axis, time-to-solution on the vertical. Each point plotted represents the mean time-to-solution for the 10 optimizations conducted for the given values of f^* and α. The composite strategy results show the mean potential time-to-solution, *as described in section 5.1.1.*



$\alpha = 5$ $\alpha = 10$

$\alpha = 15$ $\alpha = 20$

$\alpha = 25$

Figure 5.7: *A comparison of the efficiency of optimizations of $F3$ using the ignorance minimizing strategy and the composite strategy. The performance target f^* is plotted on the horizontal axis, time-to-solution on the vertical. Each point plotted represents the mean time-to-solution for the 10 optimizations conducted for the given values of f^* and α. The composite strategy results show the mean* potential time-to-solution, *as described in section 5.1.1.*

the goal does so coincidentally, not by design. We may thus generally expect the model of the goal surface to be refined to sufficient degree for the goal to be achieved by a point selected by the maximum likelihood strategy before such a point is serendipitously located.

This expectation is borne out by the results obtained for $F2$ and $F3$, but not by those for $F1$. However, it is clear that sufficient evaluations need to be performed to build a reasonably accurate model before the reasoning behind this hypothesis can apply. The numbers of evaluations being performed in optimizations of $F1$ are considerably lower than in the cases of the other two functions, so it may well be that a solution is being found serendipitously by the ignorance minimizing strategy before the model is refined enough for one to be found "rationally".

The lack of dependence on α is surprising, but gratifying: since accurate specification of the correct value of α is unlikely to be possible in practical cases, high sensitivity to its value would present a problem.

5.4.3 Allocation of Trials and Convergence

Hypotheses

A number of hypotheses relating to the allocation of trials and convergent behaviour of the maximum likelihood and ignorance minimizing strategies were proposed above:

1. Both the maximum likelihood and ignorance minimizing strategies will allocate trials to all regions of the surface, with a greater density in those regions which most nearly attain the goal.
2. More difficult goals (i.e. higher performance targets) will lead to a lowering of the

density of trials allocated within higher-performance regions, and a corresponding rise in the density of trials in lower-performing regions.

3. The ignorance minimizing strategy will not produce convergence. The maximum likelihood strategy will converge, but only on an evaluated point which achieves the optimization goal, once such a point is found.

4. All local optima of the multimodal surfaces will be investigated when the goal set for the optimization is not actually achievable.

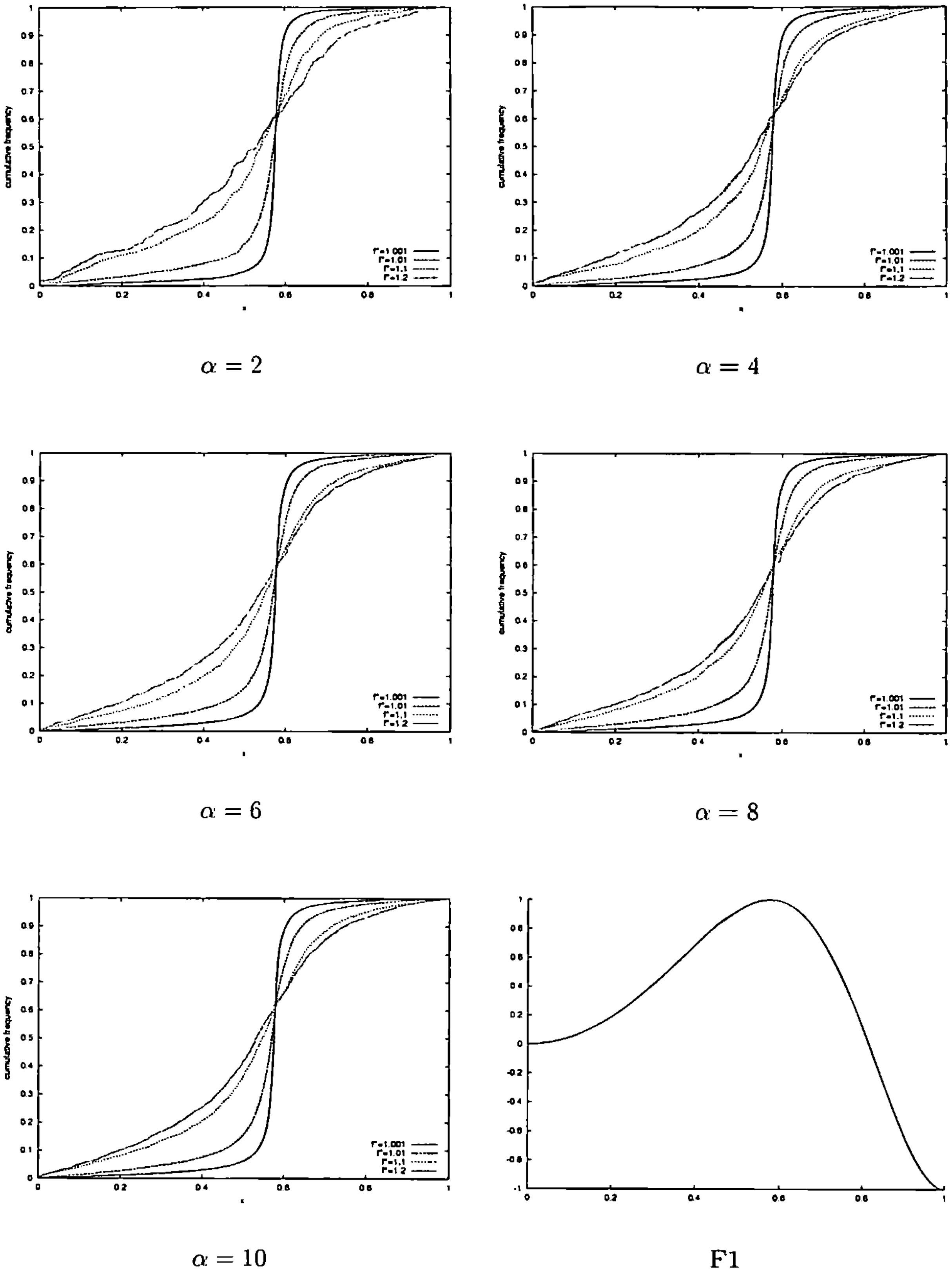
As discussed above, these hypotheses were investigated using the results for the trials in which the goal of the optimization was not achievable anywhere in the design space.

Results

Figures 5.8–5.13 show the cumulative distributions of the locations in the design space of points evaluated for each set of 10 trials for given α and f^* values . (After the standard manner of cumulative distributions, the higher the gradient of the plot, the higher the density of points evaluated in that region. This format turns out to be much clearer than the corresponding density functions, which need to be excessively filtered in order to make the results for different f^* values distinguishable from each other.) For ease of reference, each figure also shows the performance function in question.

Several points are readily apparent from these figures:

1. Very regular and predictable distributions of trials are evident, with the exception of the results for the maximum likelihood strategy when applied to $F2$, the distributions for which appear somewhat erratic for the lower values of f^* (figure 5.12).



$\alpha = 2$

$\alpha = 4$

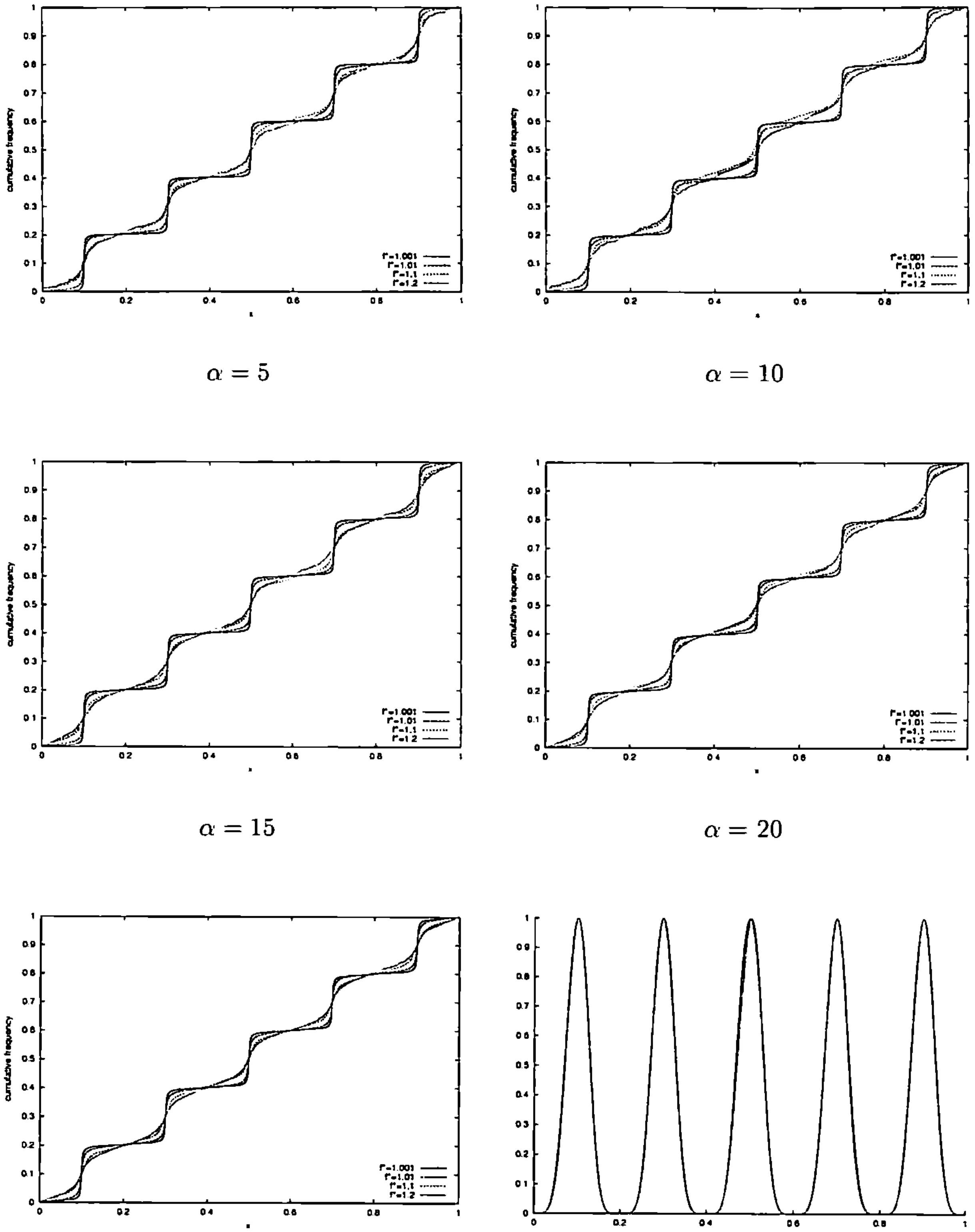
$\alpha = 6$

$\alpha = 8$

$\alpha = 10$

F1

Figure 5.8: *The cumulative distribution of the positions of evaluated points within the design space for optimizations of the function $F1$ using the ignorance minimizing strategy. Location within the design space is plotted on the horizontal axis, cumulative relative frequency of evaluations on the vertical. Each line represents results for the 10 optimizations carried out for a given combination of fitness target, f^* and model parameter, α.*



$\alpha = 5$

$\alpha = 10$

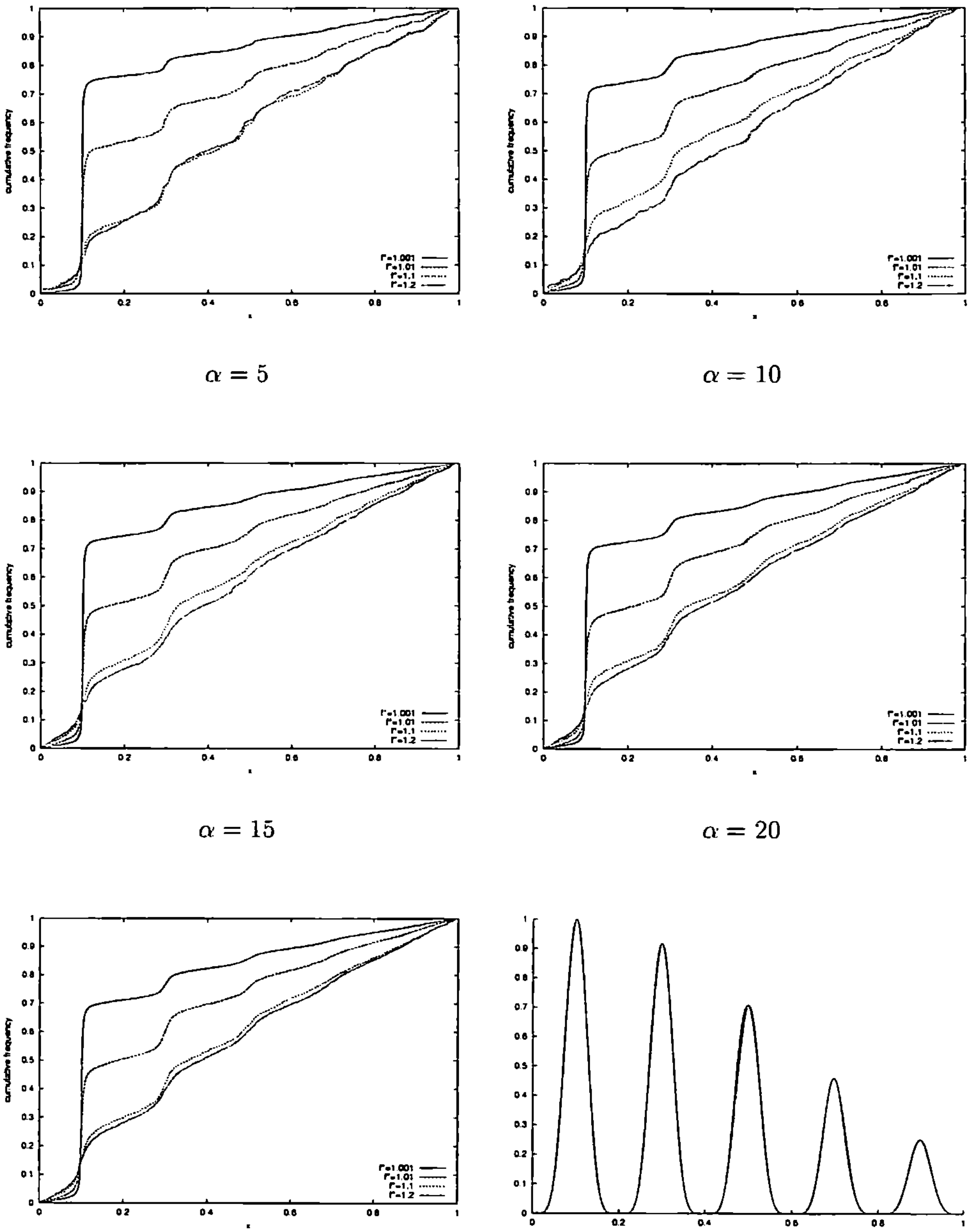
$\alpha = 15$

$\alpha = 20$

$\alpha = 25$

F2

Figure 5.9: *The cumulative distribution of the positions of evaluated points within the design space for optimizations of the function F2 using the ignorance minimizing strategy. Location within the design space is plotted on the horizontal axis, cumulative relative frequency of evaluations on the vertical. Each line represents results for the 10 optimizations carried out for a given combination of fitness target, f^* and model parameter, α.*



$\alpha = 5$

$\alpha = 10$

$\alpha = 15$

$\alpha = 20$

$\alpha = 25$

F3

Figure 5.10: *The cumulative distribution of the positions of evaluated points within the design space for optimizations of the function F3 using the ignorance minimizing strategy. Location within the design space is plotted on the horizontal axis, cumulative relative frequency of evaluations on the vertical. Each line represents results for the 10 optimizations carried out for a given combination of fitness target, f^* and model parameter, α.*

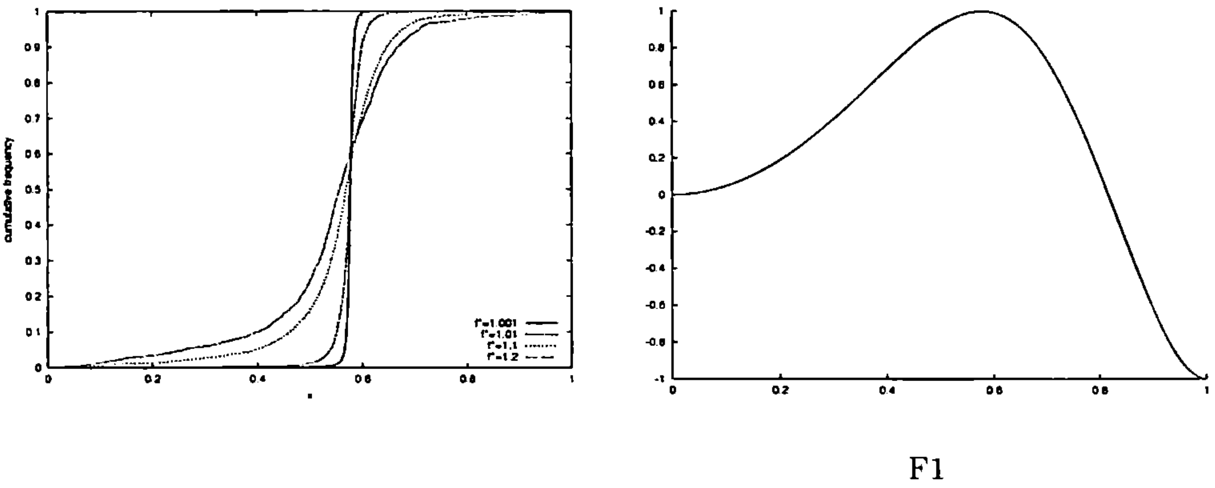


Figure 5.11: The cumulative distribution of the positions of evaluated points within the design space for optimizations of the function $F1$ using the maximum likelihood strategy. Location within the design space is plotted on the horizontal axis, cumulative relative frequency of evaluations on the vertical. Each line represents results for the 10 optimizations carried out for a given fitness target, f^*.

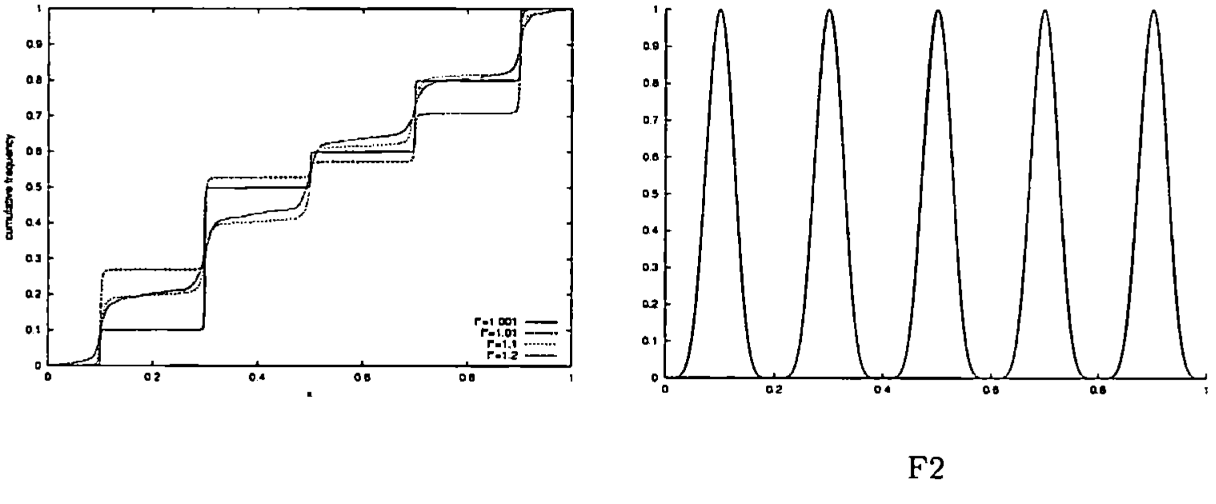
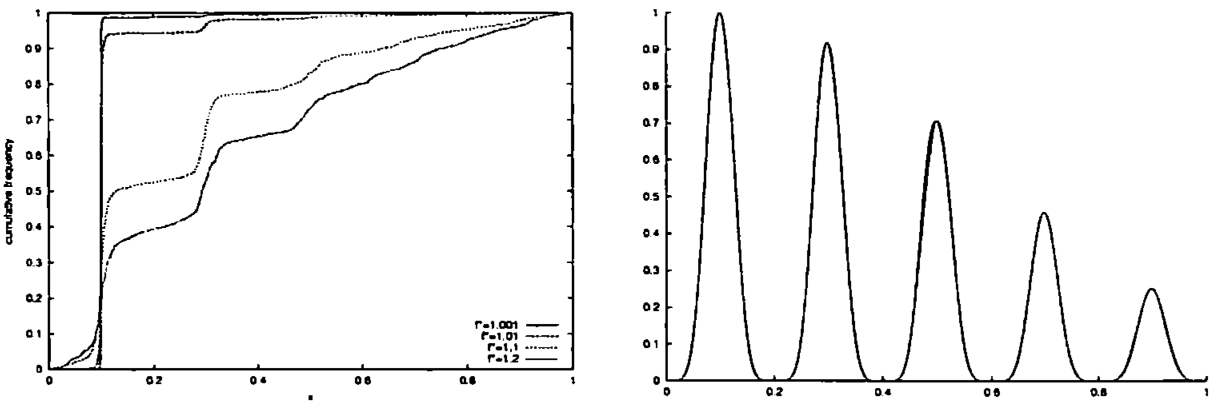


Figure 5.12: The cumulative distribution of the positions of evaluated points within the design space for optimizations of the function $F2$ using the maximum likelihood strategy. Location within the design space is plotted on the horizontal axis, cumulative relative frequency of evaluations on the vertical. Each line represents results for the 10 optimizations carried out for a given fitness target, f^*.



F3

***Figure 5.13:** The cumulative distribution of the positions of evaluated points within the design space for optimizations of the function F3 using the maximum likelihood strategy. Location within the design space is plotted on the horizontal axis, cumulative relative frequency of evaluations on the vertical. Each line represents results for the 10 optimizations carried out for a given fitness target, f^*.*

2. There is a marked and consistent effect from the performance target set for the optimization: the lower the target, the higher the density of points evaluated around the global optimum (and consequently the lower the density evaluated in other areas of the space).

3. There does not appear to be a significant effect from the value of α used in the ignorance minimizing strategy trials, except that the distribution curves become less smooth for smaller α values, particularly where f^* is high.

4. In all cases, the maximum likelihood strategy places a higher density of points close to the global optimum (close to each local optimum, in the case of $F2$) than any of the corresponding cases using the ignorance minimizing strategy.

Figures 5.8–5.13 demonstrate the average behaviour of each strategy. However, this does not allow convincing conclusions to be drawn about each individual optimization process. To this end, figures 5.14–5.19 show some results from single optimization trials. In these figures, the position of the points selected for evaluation is plotted on the vertical axis, time on the horizontal. (The cases shown were not specially selected—they are simply the results for the first run made under each set of conditions. In the case of the ignorance minimizing strategy, results for two values of α only are shown. The results shown are representative of the results obtained in the other runs.)

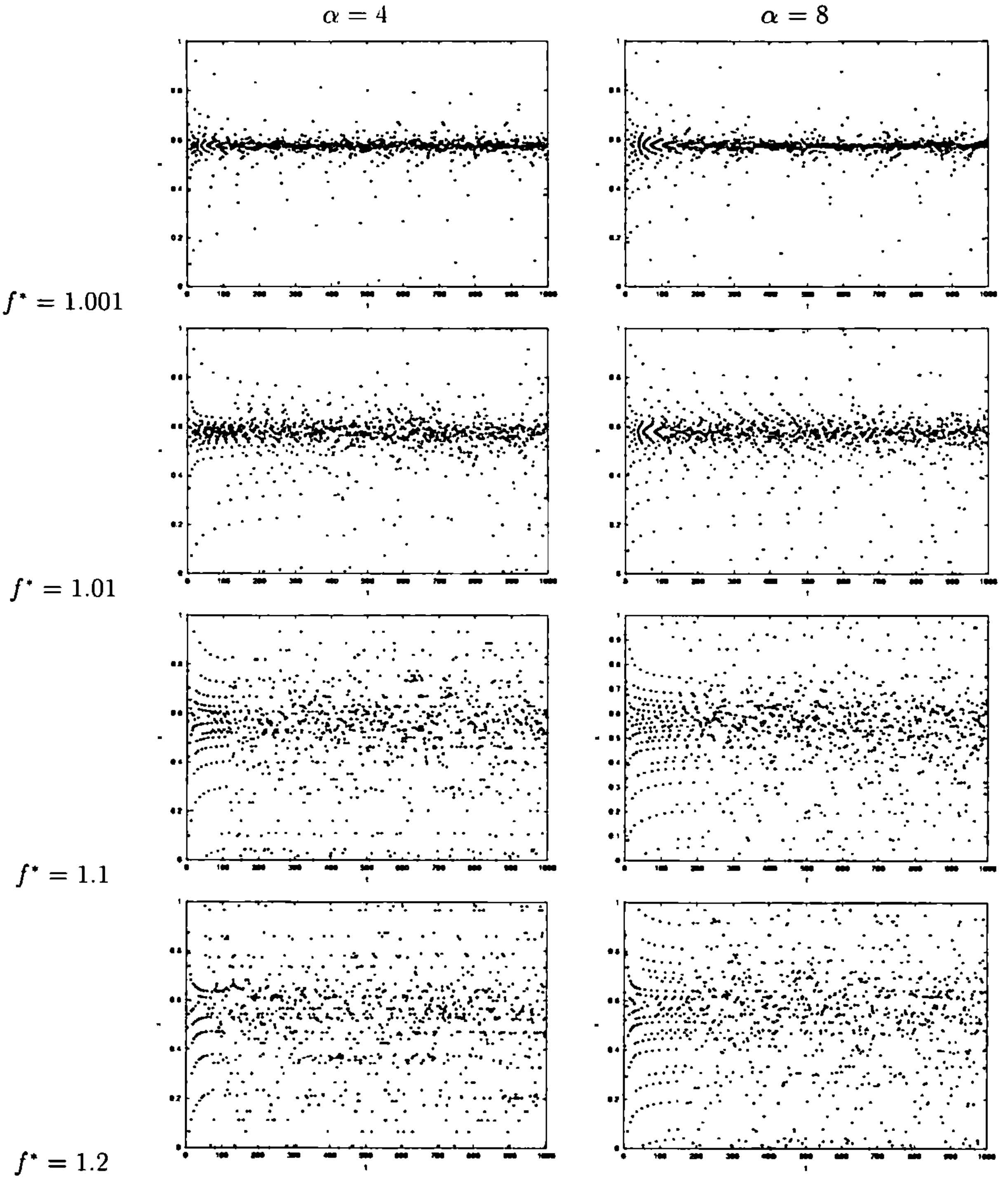


***Figure 5.14:** The positions in the design space of points evaluated for some individual optimizations of $F1$ using the ignorance minimizing strategy. Position in the design space is shown on the vertical axis, time on the horizontal.*

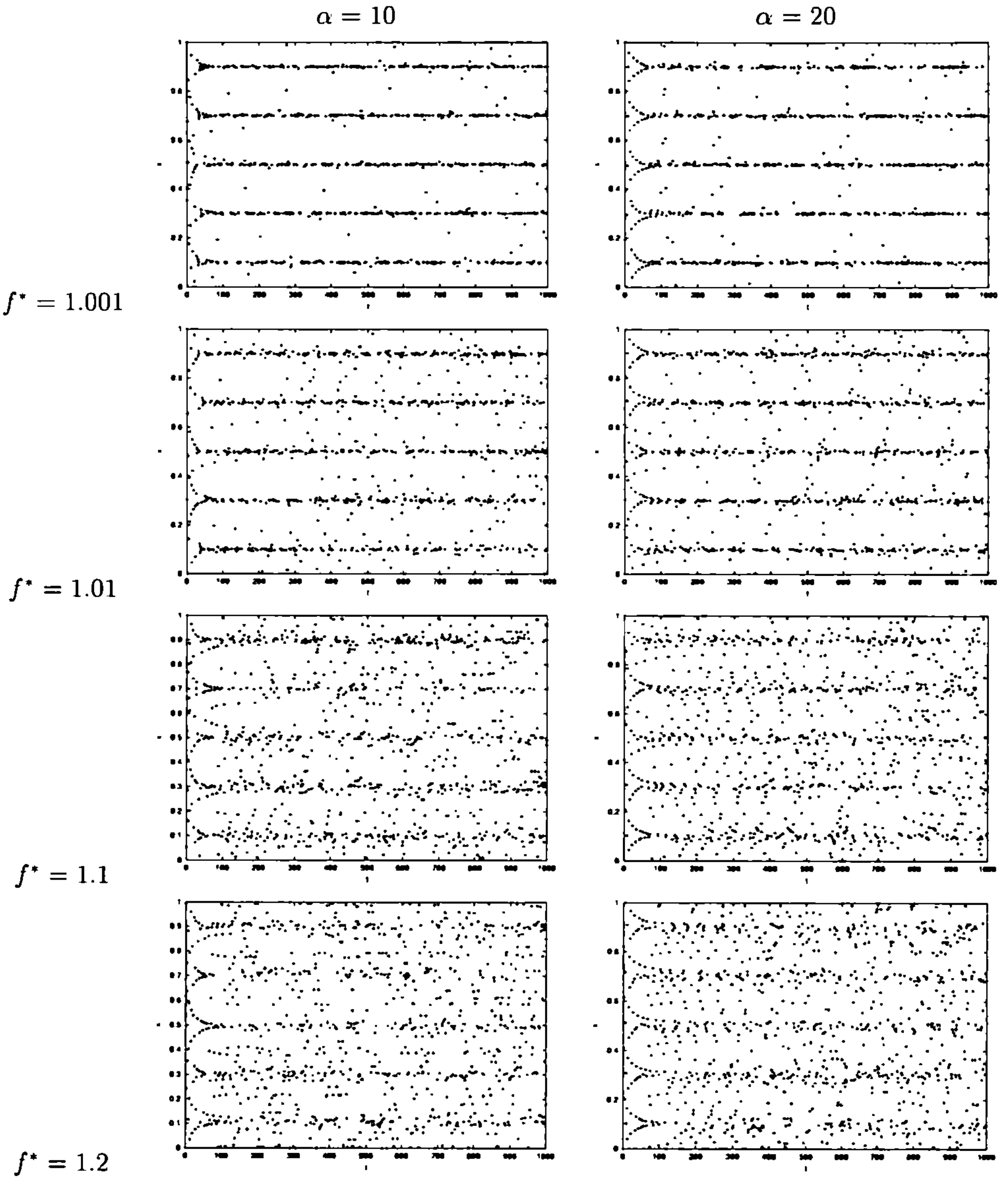


Figure 5.15: *The positions in the design space of points evaluated for some individual optimizations of $F2$ using the ignorance minimizing strategy. Position in the design space is shown on the vertical axis, time on the horizontal.*

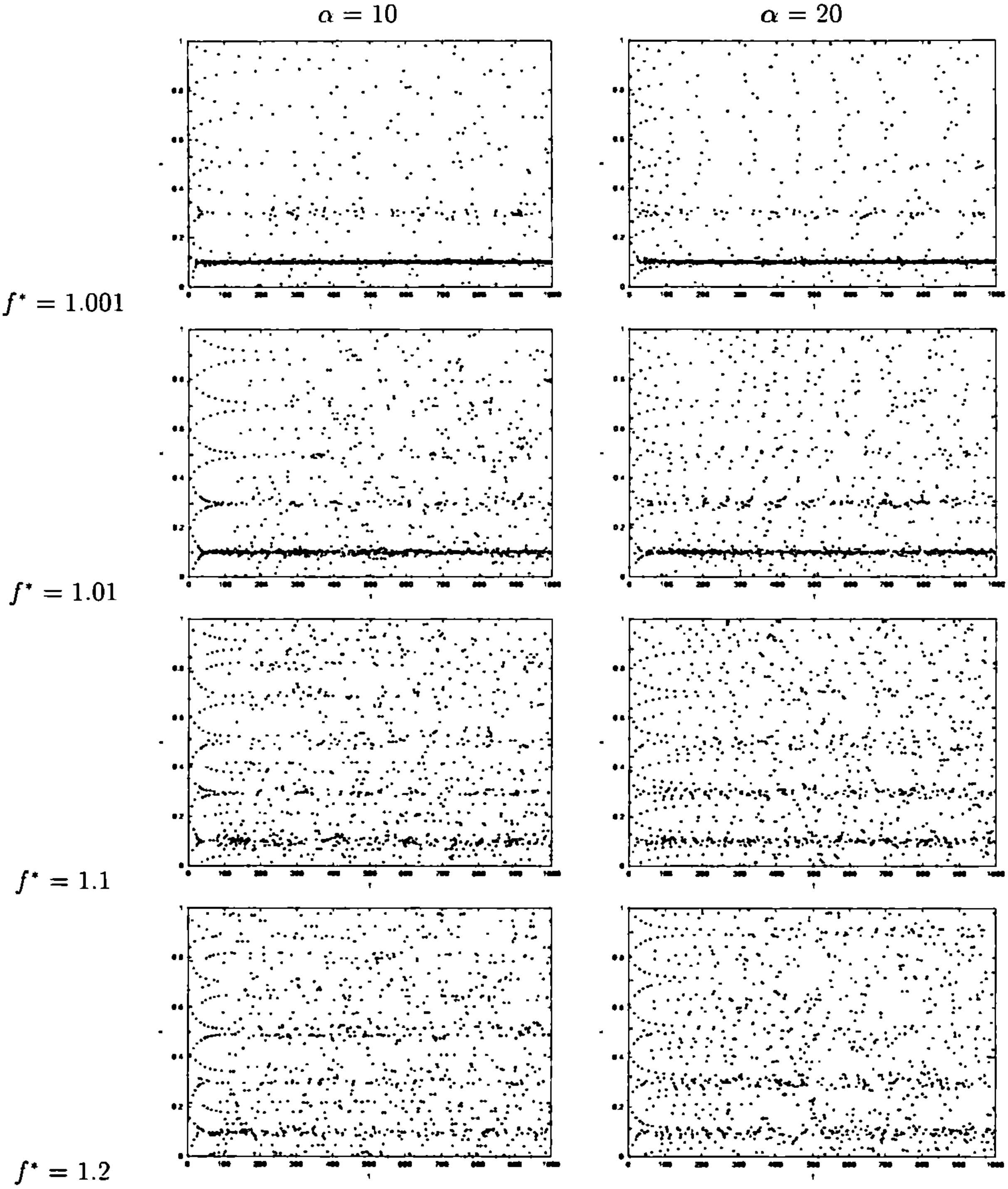
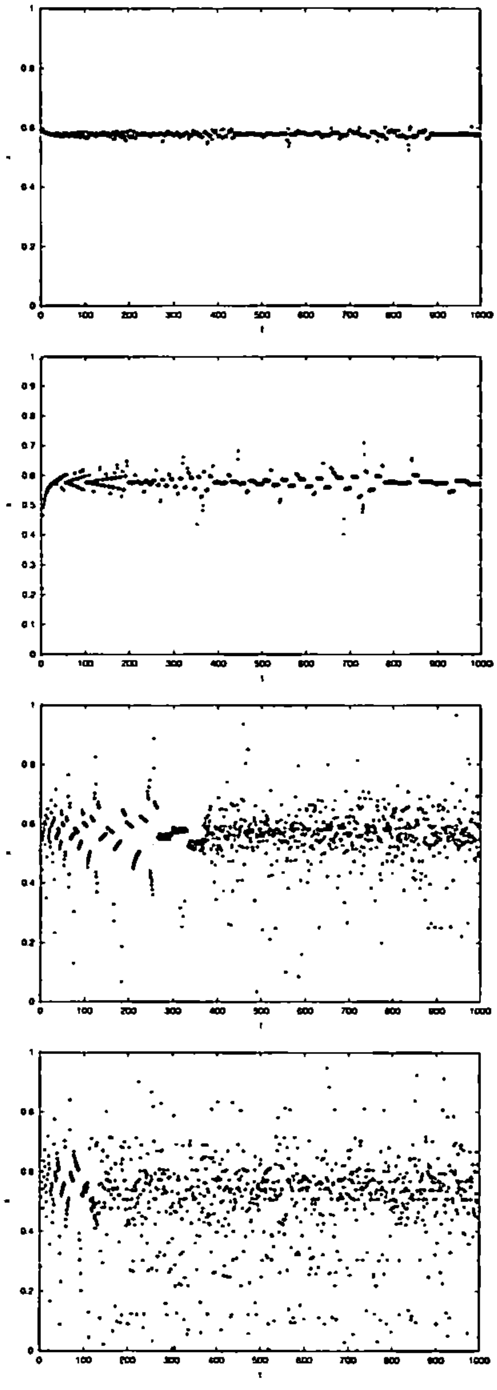


Figure 5.16: *The positions in the design space of points evaluated for some individual optimizations of $F3$ using the ignorance minimizing strategy. Position in the design space is shown on the vertical axis, time on the horizontal.*



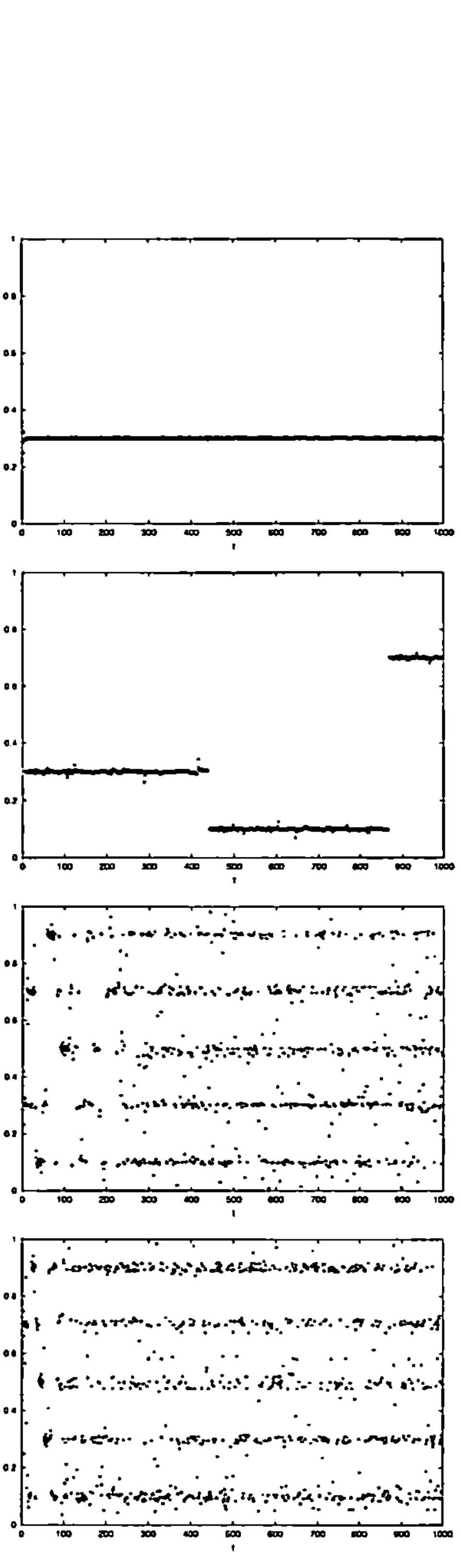
$f^* = 1.001$

$f^* = 1.01$

$f^* = 1.1$

$f^* = 1.2$

Figure 5.17: *The positions in the design space of points evaluated for some individual optimizations of* $F1$ *using the maximum likelihood strategy. Position in the design space is shown on the vertical axis, time on the horizontal.*



$f^* = 1.001$

$f^* = 1.01$

$f^* = 1.1$

$f^* = 1.2$

Figure 5.18: *The positions in the design space of points evaluated for some individual optimizations of F2 using the maximum likelihood strategy. Position in the design space is shown on the vertical axis, time on the horizontal.*

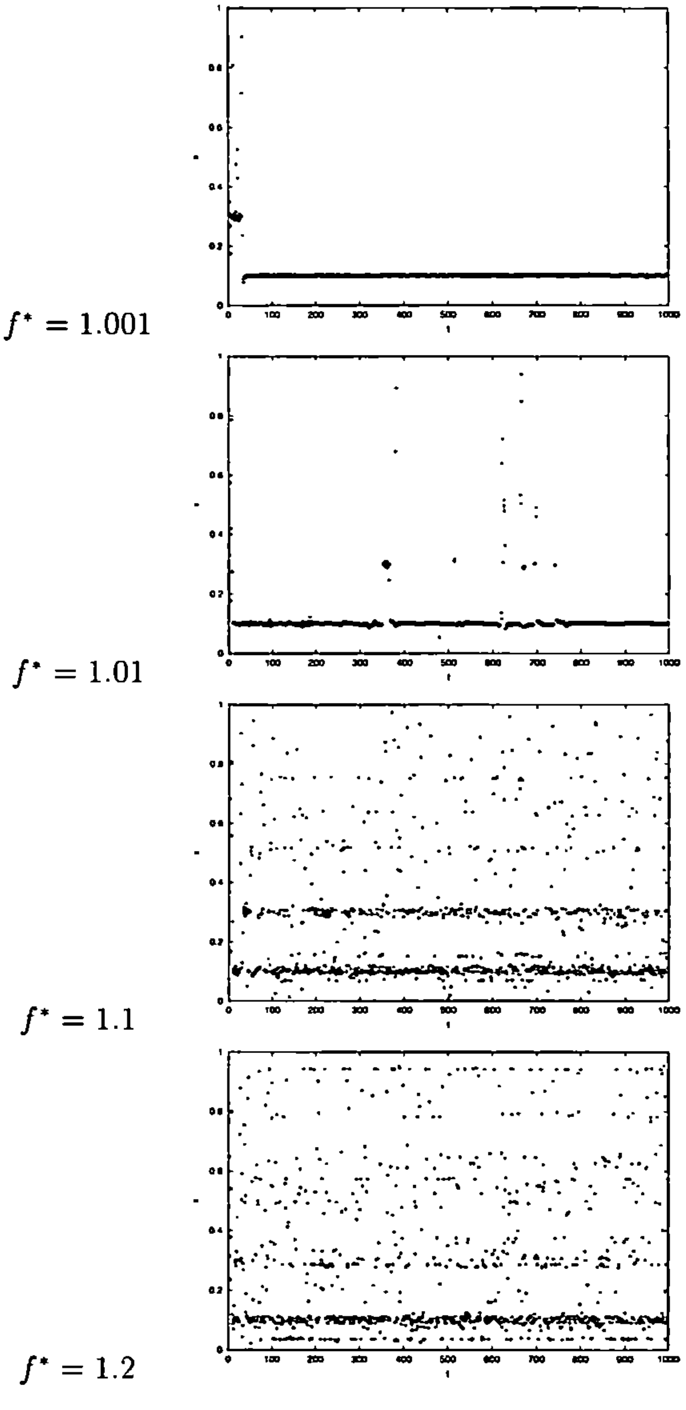


Figure 5.19: *The positions in the design space of points evaluated for some individual optimizations of F3 using the maximum likelihood strategy. Position in the design space is shown on the vertical axis, time on the horizontal.*

Discussion

Allocation of Trials The overall behaviour with respect to the distribution of evaluations over the surface is exactly as hypothesized (hypotheses 1 and 4). Both strategies can be seen to achieve coverage of the entire surface, while concentrating most on the region or regions containing the optimum. In the case of the multimodal surfaces, effort is divided among the peaks according to their performance: all peaks in $F2$ receive a similar density of trials, whereas the density for $F3$ increases according to the performance of the peak in question. The exception is the irregular results for $F2$ under the maximum likelihood strategy for low performance targets, the reason for which will be discussed below.

Hypothesis 2 also is supported: there is a clear effect of higher performance targets in tending to even out the distribution of evaluations across the surface.

Convergence The results presented in figures 5.14, 5.15, and 5.16 show the resistance of the ignorance minimizing strategy to convergence. Although evaluations are concentrated around local optima (and that to an increasing degree as the performance target decreases), yet there is no concentration on any such peak to the exclusion of other areas of the space. On the multimodal surfaces, all optima receive attention, and on $F3$ the more promising peaks receive greater attention than the less promising ones.

The corresponding results for the maximum likelihood strategy, in figures 5.17, 5.18, and 5.19 show the same resistance to convergence, although this is perhaps not so clear for all performance targets, and the claim may require some justification.

Consider first the plots for $f^* = 1.1$ and $f^* = 1.2$. In the early stages (the first 100 or so evaluations), a clear "clustering" effect is visible (except, perhaps, in $F3$'s results for $f^* = 1.2$), as one region of the space is investigated thoroughly before being discarded as attention shifs to another, more promising region. As more and more of the space is investigated and proves unforthcoming, effort shifts back to previously investigated regions for an even more dense search. That this is not convergent behaviour is shown by the readiness with which the optimization switches from one region to another (remembering that there is no random process causing the switch, but a rational decision).

Similar behaviour can be seen in the $F2$ results for $f^* = 1.01$ (figure 5.18). However, since the target in this case is so much closer to the performance values actually attainable, each peak takes much longer to eliminate from the search. In the case shown, the area around the local optimum at $x = 0.3$ takes approximately 430 evaluations to eliminate, whereupon attention shifts to the peak at $x = 0.1$, which is investigated for a very similar period before the optimization moves on again to the next peak. We can expect this pattern to continue until around 2200 evaluations, at which point all 5 peaks will have been investigated to approximately the same extent. Similar behaviour can be seen for $F3$ with $f^* = 1.01$ (figure 5.19), except that the non-global optima are of lower performance, and can be eliminated from consideration much more swiftly, with attention then returning to the area around the global optimum. Nevertheless, as the area around the global optimum becomes less and less likely to contain a satisfactory solution, the sub-optima must be revisited periodically, as can be seen by the significant investigation of the peak containing the local optimum at $x = 0.3$ which occurs at around $t = 350$.

For this reason, we may conclude that the results for $f^* = 1.001$ do not represent convergence. Even though the optimization process apparently fixates on a single peak, this peak will eventually prove unsatisfactory, and attention will shift to other areas of the space which now appear more likely to contain a solution. However, to eliminate each peak when the performance target is so close to the values really attainable would appear to require more than 1000 evaluations.

This observation explains the irregularity of the distributions of trials observed on $F2$ using the maximum likelihood strategy, for the lower performance targets (figure 5.12). For the relatively low number of runs (10) under each set of conditions, the chance effects of which peaks happen to be investigated first in each trial (which will be determined by the random point chosen to start the optimization) become visible. With more runs, or by making each run longer, these irregularities may be expected to be reduced.

We can therefore conclude that hypothesis 3 is supported, and that neither strategy exhibits convergent behaviour in the cases where the goal is not achieveable—despite the convergence shown by the maximum likelihood strategy in cases where the goal is actually achievable. It would appear that as long as there is a finite separation between the target performance and the performance levels actually achievable, then the maximum likelihood strategy will avoid convergence: the separation ensures some distance between successive evaluations, which causes the model in the area of the peak being investigated to be significantly refined by each evaluation, so that eventually the probability for every point on the peak is reduced below that of some point elsewhere on the surface.

5.4.4 Qualitative Differences

In reviewing the data on individual optimization runs, such as that shown in figures 5.14–5.19, an apparent difference between the means of operation of the ignorance minimizing and maximum likelihood strategies was noted.

While both strategies, on average, cover the whole space and place more evaluations in areas with higher fitness, the maximum likelihood strategy appears to accomplish this by dwelling in a particular region for some time, and then moving on to another region, while the ignorance minimizing strategy spreads its evaluations more diversely over time.

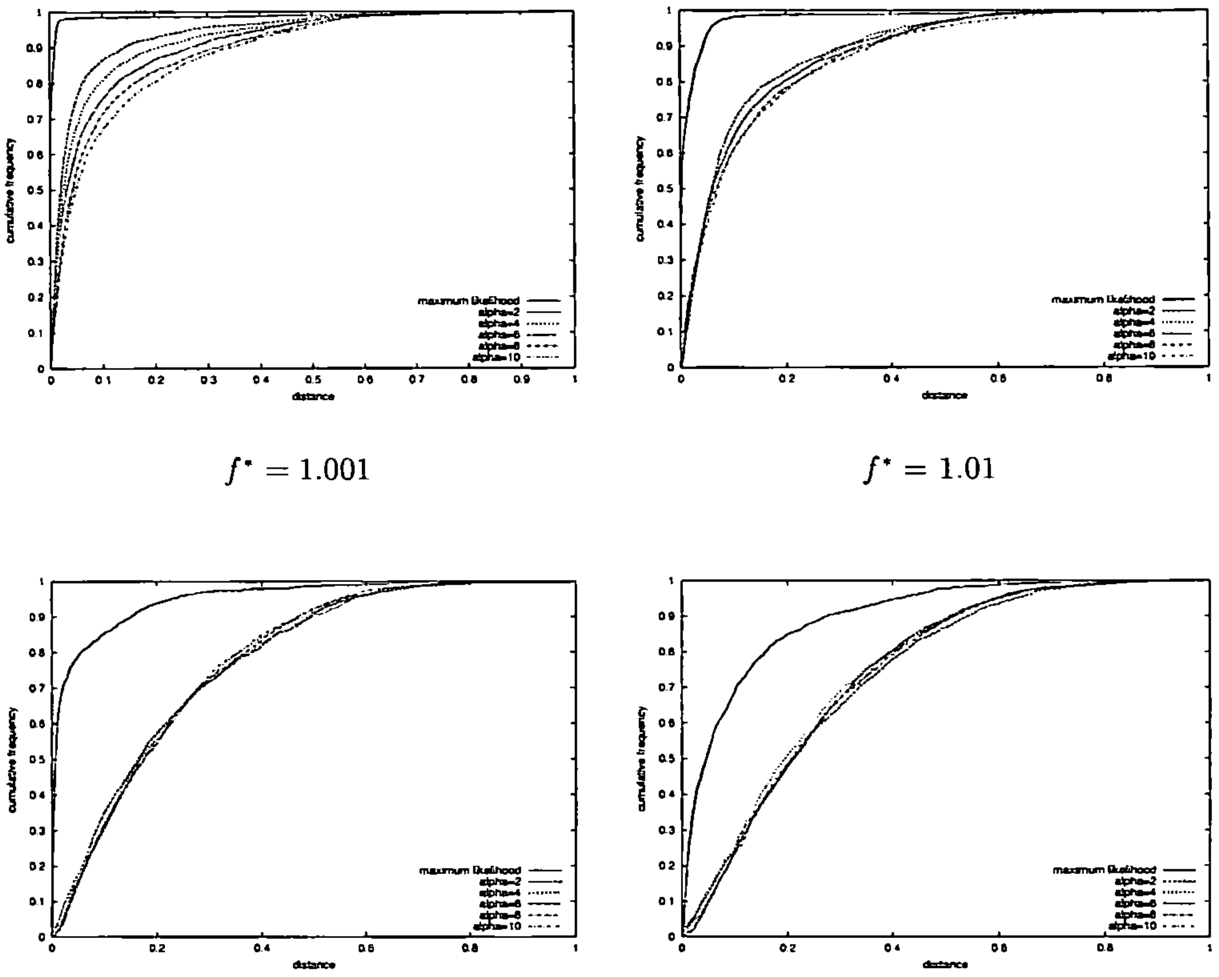
Since this observation was based on results from only single runs, figures 5.20–5.21 are included to show that the observation applies generally. These figures plot the cumulative frequency of the distance in the design space between successive evaluations for the first 200 evaluations, over the 10 trials at the same values of α and f^* represented by each line.

It can be seen from these figures that the maximum likelihood strategy has a marked tendency to take smaller steps between consecutive evaluations than does the ignorance minimizing strategy, which would appear to confirm the tendency observed for the maximum likelihood strategy to investigate particular regions one after the other, while the ignorance minimizing strategy distributes attention over the space more uniformly.

5.5 Discussion

5.5.1 Optimization Strategies

The failure of the maximum likelihood strategy as a successful optimization strategy in its own right was surprising, even though this use has not been advocated. It had been expected that the strategy would produce successful, but inefficient optimization if used alone. The behaviour of the strategy is still rational according to the assumptions



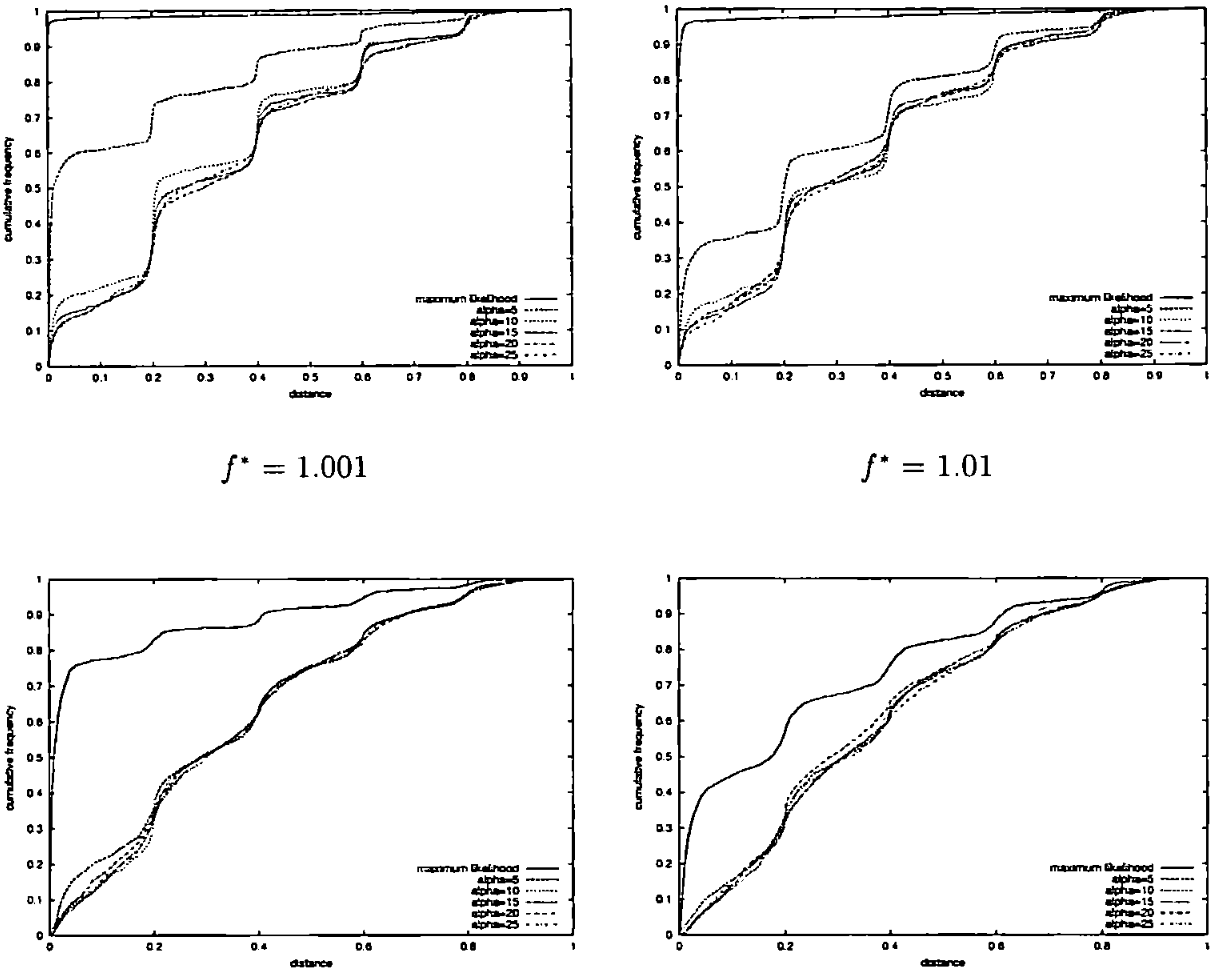
$f^{*} = 1.001$

$f^{*} = 1.01$

$f^{*} = 1.1$

$f^{*} = 1.2$

***Figure 5.20:** The distribution of distances in the design space between successive evaluations during optimizations of $F1$. Distance in the design space is plotted on the horizontal axis, cumulative relative frequency on the vertical. Each plot shows results for the first 200 evaluations in each of 10 trials conducted with the same values of f^{*} and α.*



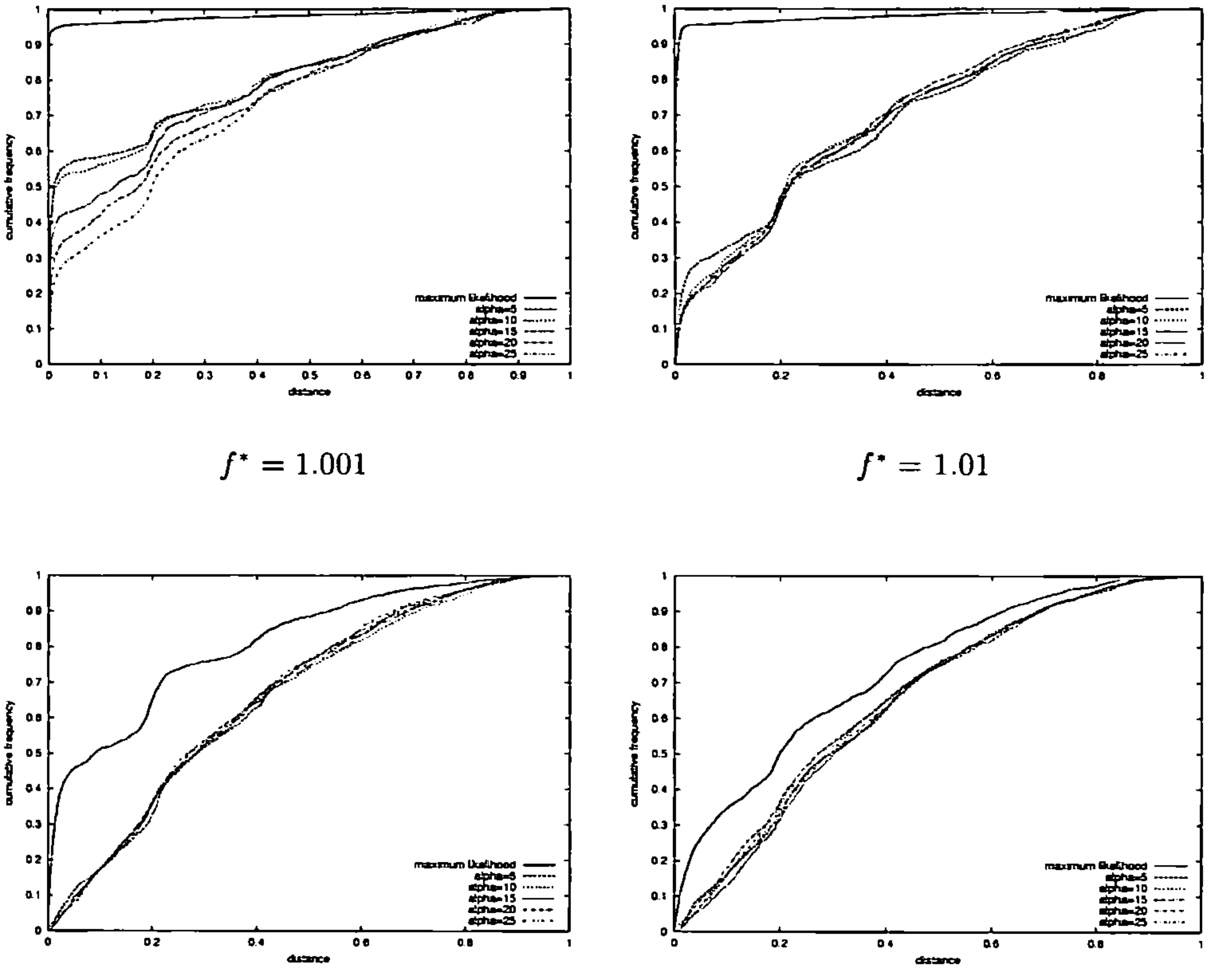
$f^* = 1.001$

$f^* = 1.01$

$f^* = 1.1$

$f^* = 1.2$

***Figure 5.21:** The distribution of distances in the design space between successive evaluations during optimizations of F2. Distance in the design space is plotted on the horizontal axis, cumulative relative frequency on the vertical. Each plot shows results for the first 200 evaluations in each of 10 trials conducted with the same values of f^* and α.*



$f^* = 1.001$

$f^* = 1.01$

$f^* = 1.1$

$f^* = 1.2$

***Figure 5.22:** The distribution of distances in the design space between successive evaluations during optimizations of F3. Distance in the design space is plotted on the horizontal axis, cumulative relative frequency on the vertical. Each plot shows results for the first 200 evaluations in each of 10 trials conducted with the same values of f^* and α.*

made about the space and the purpose for which it was developed, however, and it seems likely that if, for example, an assumption of smoothness was made in addition to that of continuity, then the effect would be a reduction of the convergent behaviour observed.

The convergence of the maximum likelihood strategy serves to reinforce the importance of exploration. When the point judged most likely to achieve the goal fails to do so when evaluated, that is indicative of a mismatch between the current model of the goal surface and the actual surface. Being purely exploitative, the maximum likelihood strategy has no role in the reduction of such a mismatch: that is exploration, and is the province of the ignorance minimizing strategy.

The ignorance minimizing strategy proved successful as an optimization strategy in its own right. However, it achieves the optimization goal only by chance when the point expected to yield most information about the goal surface happens also to exceed the performance target. In the experiments described in this chapter, there was a practical upper limit for the probability of any point's achieving the goal of 0.5. (a consequence of the form of the assumptions and goal adopted). For other assumptions and other goals, this may not be so, and the ignorance minimizing strategy will tend to avoid regions with very high probability to the same extent as those with very low probability, when a single evaluation selected by the maximum likelihood strategy might complete the optimization.

As previously suggested, therefore, a strategy such as the composite strategy which applies first the ignorance minimizing strategy then the maximum likelihood strategy can be expected to outperform either of the strategies used alone. The evidence obtained from the experimentation described in this chapter supports this suggestion. However, a question which has not been addressed is how to decide, when optimizing using the composite strategy, the right moment to switch from the ignorance minimizing strategy to the maximum likelihood strategy. A suggestion for a possible answer to this question will be given in chapter 6.

5.5.2 Effect of the Goal

One of the main criticisms of the use of genetic algorithms advanced in chapter 1 was their inability to represent the goal of the optimization, which was cited as one of the contributory causes of convergence. This criticism is supported by the significant effect which the difficulty of the goal has been observed to have on the optimization process: the further out-of-reach the goal, the less the concentration on the peaks of the surface, and thus the more diverse the search. We might think of this effect as a "diversity pressure"—the complementary force to the selection pressure in evolutionary systems—which pushes search away from regions of the space found unlikely to achieve the goal, rather than towards regions with high expected performance.

The operation of the goal on the behaviour of the optimization can be intuitively understood in terms of human problem-solving: if the requirements of a problem are very close to requirements for which a known solution exists, the tendency is to look for similar solutions, or adjust the existing one to meet the new requirements (an evolutionary process, as defined in section 1.3.7). If, on the other hand, the requirements are considerably beyond what is achieved by existing solutions, there will be an increased tendency towards trying novel approaches—looking elsewhere in the space of possible solutions. It is this latter behaviour—the recognition that even the current best solution is unacceptable, and that areas of the solution space which have not been investigated should be tried—which is missing from the the evolutionary metaphor. It is missing because it requires representation of the goal of the search, not just relative comparisons between candidate solutions (competition), and the propagation of the least poor.

5.5.3 Exploration and Exploitation

Much of the behaviour of the maximum likelihood strategy resembles local search, in that it tends to dwell in a single high-performance region of the surface for some time before moving to another region. That exploitative behaviour is not simply equivalent to local search, however, can be seen by the readiness of the strategy to switch between the peaks of the multimodal surfaces, and the much more diverse search exhibited as the target performance is increased (see figure 5.17).

By contrast, the exploratory strategy shows a much greater tendency to move to a new region of the design space with each selection of a point to evaluate (see figures 5.14–5.16). Allocation of trials over the design space is much more uniformly diverse over time than for the maximum likelihood strategy.

This difference in behaviour of the two strategies can be understood in terms of their different aims as regards the changes in the model resulting from the evaluation of the points they select. Both strategies base their operation on the current model of the goal surface for the optimization, but the ignorance minimizing strategy is explicitly designed to maximize the changes in the feature of the model with which it is concerned (the total uncertainty about the goal). The maximum likelihood strategy, by contrast, does not take account of the changes which are expected to result in the goal surface model due the evaluation of the points it selects. The model is therefore more likely to be little changed after such evaluations, leading to the selection of the next point for evaluation being more likely to be close to the last.

5.5.4 Conclusion

The behaviours of the ignorance minimizing, maximum likelihood, and composite strategies have been broadly found to be in accordance with the predictions made in chapter 3. The exception is that the maximum likelihood strategy has been found to exhibit more convergent behaviour than predicted (although this does not affect its use as envisaged as part of the composite strategy).

The importance of representing the goal of the optimization process has been emphasized, with a very clear effect of the difficulty of the goal on the diversity of the optimization process.

Chapter 6

Discussion and Conclusion

6.1 Overview

Chapter 1 developed a number of criticisms of the adoption of an evolutionary metaphor for the understanding and solution of optimization problems. The criticisms centred around the recognition that biological evolution and natural selection are not goal-driven processes of design, leading to some fundamental failings in attempts to achieve design and problem solving behaviour by using an analogy with biological evolution. Particular targets for criticism were the failure of evolutionary optimizers explicitly to represent the goal of the optimization process, and their poor qualities for representing and using domain- and problem-specific knowledge.

Arguing from the influential no free lunch theorems of Wolpert and Macready [119], chapter 2 suggested an alternative, non-metaphorical framework in which to consider optimization. The framework centres on the use of available knowledge (or reasonable assumptions) about the nature of the problem to construct a model of the performance surface being optimized. This, together with a similar model of the goal of the optimization, can then be used for optimization by the application of rational, distinct strategies for both exploratory and exploitative behaviour.

Chapter 3 presented a mathematical formulation of the conceptual model of chapter 2, in the form of the use of Bayesian belief revision to construct probabilistic models of the goal and performance surfaces for an optimization problem, and further clarified the distinction between exploration and exploitation, suggesting a procedure to achieve each, in the form of the ignorance minimizing and maximum likelihood optimization strategies. The composite strategy was also proposed, being a principled combination of exploratory and exploitative behaviour for achieving effective optimization.

Chapter 4 described an implementation of the ignorance minimizing and maximum likelihood strategies for one-dimensional real-valued design and performance spaces, based on a simple assumption of uniform continuity of the performance surface. Chapter 5 reported on experimentation carried out to investigate the optimization behaviour of the ignorance minimizing, maximum likelihood and composite strategies on some simple test functions, using the implementation of each strategy developed for the purpose in chapter 4.

In this chapter, I summarize the key findings of this thesis, identify some valid criticisms, and suggest potentially interesting directions for future investigation.

6.2 Key Findings

6.2.1 Applicability

The class of problems addressed in this thesis has been restricted to optimization problems consisting of the pursuit of one or more points in a design space which satisfy some goal with respect to a performance surface defined over the space. The design space has been assumed to consist of real-valued continuous parameters, and the performance function to be static and non-noisy. The findings presented are not asserted to hold outside of the domain of problems of this type. Although some of the arguments advanced may be applicable (or adaptable) to other classes of problem, consideration has not been given to such application.

6.2.2 Non-Metaphorical, Knowledge-Centred Optimization

It has been demonstrated that an effective optimization technique can be constructed which addresses the deficiencies of the evolutionary metaphor discussed in chapter 1. Specifically, it:

- Uses no metaphor to obscure the fundamental nature of the optimization problem, centring instead on the use of available knowledge (or acceptable assumptions) to direct the optimization process, and exhibits no hidden biases or intractable dynamics. All knowledge developed during the optimization is retained, and used to guide the process.

- Explicitly represents and pursues the goal of the optimization, allowing the nature of the goal to affect the progress of the search This contrasts with the characterization of evolutionary processes as "competitive"—based only on the relative performances of the current population (see chapter 1). The effect of the nature and difficulty of the optimization goal on the diversity of search behaviour was seen to be significant in the experiments reported in chapter 5.

- Achieves coverage of the space in a principled way, allocating search effort to a region based both on estimates of the performance values achievable in that region and on the level of certainty associated with those estimates.

- Investigates all local optima in a principled fashion, until the goal is achieved, without the need to maintain a set of concurrent "niches".

- Does not converge at all, let alone prematurely.

6.2.3 Exploration and Exploitation

Achieving a successful balance of exploitative and exploratory behaviour is widely recognized as important for successful optimization. Under the approach adopted in this thesis, exploratory and exploitative behaviour can be separated into distinct strategies, and at any point during optimization the next point for evaluation may be selected by either one strategy or the other. This is much finer-grained and more transparent control than is available through adjusting the parameters of a typical GA.

Neither strategy is inherently stochastic: if an appropriate form and level of knowledge is available, then at any stage during optimization unique points can typically be identified which represent the "most exploitative" and "most exploratory" evaluations to perform. Exploitative behaviour is achieved by selecting for evaluation the point judged most likely to achieve the goal, exploratory behaviour by selection based on the expected reduction in the total uncertainty of the goal surface associated with a point's evaluation.

Both are global strategies, in that the selection of a point to evaluate is made based on a comparison of the expected exploratory or exploitative value of all points in the design space: there is no operator bias, or effect of the reachability of points from a current population. Under the assumptions adopted and on the surfaces investigated in chapter 5, exploitation was observed to exhibit greater propensity towards behaviour similar to local search (without being directly equivalent to local search), while exploration promoted a more continually diverse search.

The early stages of optimization are the rightful province of exploration: knowledge needs to be developed about the nature of the goal surface for the problem before exploitatory behaviour can be expected to be successful. Given a rational, deterministic exploratory strategy such as that derived, the role for exploitation in optimization is much reduced; indeed, the development of the composite strategy suggested that only a single exploitative evaluation might be necessary in any optimization. Exploitation remains essential, however: although exploration alone was shown to be a successful strategy on the problems investigated, it seems likely that it would not prove efficient on other problems. In particular, in problems where it is possible to achieve probabilities of a point's achieving the goal in excess of 0.5, exploration is expected to avoid regions with very high probabilities, requiring exploitation in order actually to achieve the goal.

6.2.4 Convergence

It was suggested in chapter 1 that the pursuit of a convergent optimizer makes sense only for problems known (or assumed) to be unimodal. Optimization using the composite strategy suggested in chapter 3 was demonstrated in chapter 5 not to converge. While the exploitative strategy did exhibit convergence, if it is only applied for a single evaluation, as suggested, such convergence will not have opportunity to manifest itself.

Convergence of an optimization is usually taken as an indication of when to terminate the process: when no point has been found for some time which has improved on the best-so-far, the conclusion is drawn that the optimum has been reached. But in order for this conclusion to be supportable, the optimization process must have actually been *looking* for better points and failed to locate any, not just repeatedly evaluating the current optimum, or points very close to it.

In a multimodal space, it is not a convergence of *trials* which is required (i.e. all or most evaluations close to the located optimum), but a convergence of *belief*. Convergence of belief occurs when the point believed most likely to be the optimum does not change significantly for some time, *despite* the fact that non-converged, diverse search has been attempting to locate a better point in other areas of the space, as well as purely locally. Convergence of belief and convergence of trials are identical for gradient-based methods on unimodal surfaces—when no local improvement can be found, the search must have reached the optimum—which fact has led to the confusion of the two concepts. The concepts diverge on multimodal problems, however, when convergence of trials becomes positively undesirable, since it prevents any significant refinement and revision of beliefs about the surface, and the concomitant increase in certainty that the true global optimum has been located.

These considerations lead to a possible answer to a question about the composite strategy which has thus far not been addressed in this thesis: When should the optimization switch from using the ignorance minimizing strategy to using the maximum likelihood strategy? The key to answering this question is the fact that an evaluation made using the maximum likelihood is not expected to have any value of future use to the optimization (since it is not intended necessarily to cause significant changes to the goal surface model), unless it actually achieves the goal. Therefore there ought to be a reasonable level of certainty that the evaluation will successfully achieve the goal before selection of a point for evaluation based on the maximum likelihood strategy should be attempted. That is, the stimulus for switching to the maximum likelihood strategy should be the observance of some degree of convergence of belief, as expressed in the model of the goal surface, concerning the location of a point which achieves the goal. How such convergence might be characterized and detected is not clear.

6.3 Criticisms

6.3.1 Computational Expense

The work reported in this thesis was undertaken under the assumption that the computational cost of the simulations of engineering systems comprising typical performance functions would remain the dominating component of the cost of any optimization method developed. This assumption may well not be valid, since the computational cost of applying the method of optimization developed to realistic problems seems likely to be very high:

- Only one-dimensional design spaces have been investigated. It seems likely that the complexity of constructing and maintaining surface models of a similar nature to those used will be combinatorially explosive as the dimensionality of the design space increases.
- The expense associated with the ignorance minimizing strategy is potentially large, with selection of each point for evaluation requiring the maximizing over the design space of a function involving nested integrals. This would be further exacerbated if the goal of the optimization were to locate the global optimum, rather than the simpler satisficing goal investigated.
- Every selection of a point to evaluate, using either strategy, effectively requires an optimization process over the design space in order to locate the most exploratory

or exploitative point for evaluation.

No rebuttal of these criticisms is offered: they are valid, and pose serious difficulties. Several relevant observations may be made, however, which may go some way to suggesting that these problems are not insuperable:

- The mathematical treatments given in this thesis are not sophisticated (reflecting the lack of formal mathematical background of the investigator). A more sophisticated mathematical approach might make inroads into such complexity issues.

- It was found that most of the calculations necessary to apply the ignorance minimizing strategy to a one-dimensional space could be pre-compiled, enabling the ignorance minimizing point for any interval to be determined by scaling a value retrieved from a table (the table took 10 days of computer time to generate, but thereafter could be re-used for any one-dimensional problem). The overall complexity of the strategy was then equivalent to that of maintaining a sorted list of the intervals in the space.

- The complete surface models did not have to be completely regenerated after each evaluation, but only those parts which were affected by the evaluation.

- Heuristics and approximate methods for implementing each strategy might prove an acceptable compromise with complexity.

- In the final analysis, the trade off between the computational expense of the optimizer and the difficulty of the problem needs to be addressed individually for each problem.

Regardless of the expense associated with implementing such an optimizer, the conceptual framework in which it has been developed may be useful for considering the operation of other optimization algorithms.

6.3.2 Optimization

The usual formulation of optimization tasks—the location of the global optimum of a surface—has not been addressed. Ideally, such problems would be approached in the same way as the satisficing problems addressed in this thesis, but using a different goal surface model—one which expressed for every point in the design space the probability of that point's being the global optimum.

However, the construction of such models would probably add significantly to the computational expense associated with the optimization. A possible alternative approach might be to pursue global optimization via a satisficing optimization in which the target performance is varied to keep it above the performance of the best point found so far, maintaining the pressure for continual improvement. Such an approach would fail to represent the true goal of the optimization within the algorithm, but might be an acceptable compromise with complexity. Given the sensitivity of optimization diversity to the target performance value (see chapter 5), careful control of the target would be necessary. Such control might be based on a judgement, based on the current model of the performance surface, of the level of performance realistically likely to be achievable.

6.4 Future Directions

6.4.1 Practical Application

The work reported in this thesis has been largely of a conceptual nature, consisting of the identification of difficulties with evolutionary approaches to optimization, the outlining of an alternative approach, and some investigations into the nature of optimization strategies which might be adopted under that approach. The starting point, however, was the immensely practical field of engineering design. In order to validate the ideas and approach of this work, they need to be applied to practical examples of engineering design optimization. Considerable development work will be necessary if such application is to become possible, including in particular the addressing of the problems of computational expense outlined in section 6.3.1 above.

Another outstanding question relating to practical application of the work is the nature of design knowledge which actually exists in engineering domains, and whether such knowledge may be readily expressed in the Bayesian model formalism developed. Some justification was presented in section 2.5.3 for the assumption of uniform continuity, from the perspective of optimization problems involving physical quantities, and section 3.4.2 expressed the belief that the Bayesian formalism is as likely as any other to form a good basis for the integration of different forms of design knowledge into the procedure for constructing models of the performance and goal surfaces.

6.4.2 Constrained Optimization

Chapter 1 outlined current evolutionary approaches to problems of constrained optimization, and suggested that the failure of the evolutionary approach explicitly to represent the goal of the optimization was a hindrance to the effective solution of such problems, leading to a tendency to converge in constraint-violation-minimizing, but nonetheless infeasible regions. Since the approach taken in this thesis *does* explicitly represent the goal of the optimization, it ought to be capable of being applied to address this failing of the evolutionary approach.

A constrained optimization problem might be addressed by constructing separate performance and goal surface models for each constraint, then combining all the separate goal surface models into a single model for the overall goal. The overall goal surface model would express the probability for each point in the space of its achieving all the individual goals. Optimization conducted based on the overall goal surface model can then be expected to exhibit the "diversity pressure" noted in section 5.5.2—pressure away from a region found unlikely to achieve the overall goal, even though it be the best performing region found so far, towards less well investigated regions of the surface.

6.5 Conclusion

To question prevailing orthodoxy in any domain is a valuable exercise. The work reported in this thesis takes a contrary view to the current popularity of the evolutionary metaphor for the understanding and conduct of optimization processes in engineering design, and concludes that the metaphor is not only unnecessary, but positively obstructive.

By approaching optimization from a viewpoint which considers the knowledge (or assumptions) applied to the problem as central, an optimization method may be achievable which addresses many of the deficiencies of evolutionary optimizers.

Of particular interest in this work are the characterizations of exploitative and exploratory search behaviour in terms of their use of, and effect on, the current state of belief about the nature of the optimzation problem, and the suggestion that any form of convergent behaviour is undesirable in an optimization method intended for use on potentially multimodal surfaces.

The experimentation carried out has been of a preliminary, investigatory nature, and much further work needs to be done—especially in determining whether the approach is applicable in practice to design optimization problems of realistic size and complexity.

References

[1] Akiko N Aizawa and Benjamin W Wah. Dynamic control of a genetic algorithm in a noisy environment. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 48–55. Morgan Kaufmann, University of Illinois at Urbana-Champaign, 1993, July 17-21.

[2] C J Aldridge, J R McDonald, and S McKee. Unit commitment for power systems using a heuristically augmented genetic algorithm. In *Proceedings of the 2nd International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 97)*, pages 433–438, Glasgow, UK, 1997. Institution of Electrical Engineers.

[3] Thomas Bäck. Optimal mutation rates in genetic search. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 2–8, University of Illinois at Urbana-Champaign, 1993, July 17-21. Morgan Kaufmann.

[4] Thomas Bäck. Evolutionary strategies: An alternative evolutionary algorithm. In J M Alliot, E Lutton, E Ronald, M Schoenauer, and D Snyers, editors, *Artificial Evolution*, pages 3–20, Brest, France, September 1995. Springer Verlag.

[5] Thomas Bäck, Martin Schütz, and Sami Khuri. A comparative study of a penalty function, a repair heuristic, and stochastic operators with the set-covering prob-

lem. In J M Alliot, E Lutton, E Ronald, M Schoenauer, and D Snyers, editors, *Artificial Evolution*, pages 320–332, Brest, France, September 1995. Springer Verlag.

[6] Peter Bentley. Guest editorial: Special issue on evolutionary design. *Artificial Intelligence for Engineering Design Analysis and Manufacturing*, 13:143, 1999.

[7] J M Bernardo and A F M Smith. *Bayesian Theory*. Probability and Mathematical Statistics. John Wiley & Sons, Inc., New York, 1994.

[8] George Bilchev and Ian Parmee. Adaptive search strategies for heavily constrained design spaces. In *Proceedings of the 22nd International Conference CAD'95*, pages 8–13, Yalta, Ukraine, May 1995.

[9] George Bilchev and Ian Parmee. Constrained optimisation with an ant colony search model. In Ian Parmee, editor, *Proceedings of the 2nd International Conference on Adaptive Computing in Engineering Design and Control*, pages 145–151, Plymouth, UK, March 1996. The University of Plymouth.

[10] Bruce Bridgeman. *The Biology of Behaviour and Mind*. John Wiley & Sons, Inc., New York, 1988.

[11] Eduardo Camponogara and Sarosh N Talukdar. A genetic algorithm for constrained and multiobjective optimization. In *Proceedings of the 3NWGA*, August 1997.

[12] Y J Cao and Q H Wu. Convergence analysis of adaptive genetic algorithms. In *Proceedings of the 2nd International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 97)*, pages 85–89, Glasgow, UK, 1997. Institution of Electrical Engineers.

[13] Raphaël Cerf. An asymptotic theory for genetic algorithms. In J M Alliot, E Lutton, E Ronald, M Schoenauer, and D Snyers, editors, *Artificial Evolution*, pages 37–53, Brest, France, September 1995. Springer Verlag.

[14] Edwin K P Chong and H Zak Stanislaw. *An Introduction to Optimization*. John Wiley & Sons, Inc., New York, 1996.

[15] Carlos A Coello Coello. Constraint-handling through a multiobjective optimization technique. In Annie S Wu, editor, *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 117–118, 1999.

[16] David A Coley. *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific Publishers, Singapore, 1999.

[17] Robert J Collins. Artificial evolution and the paradox of sex. In Ray Paton, editor, *Computing with Biological Metaphors*, pages 244–263. Chapman and Hall, London, UK, 1994.

[18] R T Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13, 1946.

[19] Joseph C Culberson. On the futility of blind search: An algorithmic view of 'no free lunch'. *Evolutionary Computation*, 6(2):109–127, 1998.

[20] Yuval Davidor. Epistasis variance: Suitability of a representation to genetic algorithms. *Complex Systems*, 4:369–383, 1990.

[21] Lawrence Davis. Adapting operator probabilities in genetic algorithms. In J David Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 61–69, George Mason University, 1989. Morgan Kaufmann.

[22] Lawrence Davis. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, 1991.

[23] Kenneth De Jong and William Spears. On the state of evolutionary computation. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 618–623, University of Illinois at Urbana-Champaign, 1993, July 17-21. Morgan Kaufmann.

[24] Kenneth A De Jong. *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, 1975.

[25] Kenneth A De Jong. Genetic algorithms are *not* function optimizers. In L Darrell Whitley, editor, *Foundations of Genetic Algorithms*, pages 5–17. Morgan Kaufmann, San Mateo, California, 1993.

[26] Kalyanmoy Deb. Genetic algorithms for function optimization. In Francisco Herrera and Jose Luis Verdegay, editors, *Genetic Algorithms and Soft Computing*, pages 3–29. Springer Verlag, New York, 1996.

[27] Kalyanmoy Deb and Samir Agrawal. Understanding interactions among genetic algorithm parameters. In Wolfgang Banzhaf and Colin Reeves, editors, *Foundations of Genetic Algorithms 5*, pages 265–286. Morgan Kaufmann, San Mateo, California, 1997.

[28] Kalyanmoy Deb and David E Goldberg. An investigation of niche and species formation in genetic function optimization. In J David Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 42–50, George Mason University, 1989. Morgan Kaufmann.

[29] Kalyanmoy Deb, Jeffrey Horn, and David E Goldberg. Multimodal deceptive functions. *Complex Systems*, 7:131–153, 1993.

[30] A Eiben, R Hinterding, and Z Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 1999.

[31] Emanuel Falkenauer. Applying genetic algorithms to real-world problems. In Lawrence Davis, Kenneth De Jong, Michael D Vose, and L Darrell Whitley, editors, *Evolutionary Algorithms*, pages 65–88. Springer Verlag, New York, 1999.

[32] Terence C Fogarty. Varying the probability of mutation in the genetic algorithm. In J David Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 104–109, George Mason University, 1989. Morgan Kaufmann.

[33] David B Fogel. An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks*, 5(1):3–14, January 1994.

[34] David B Fogel. Real-valued vectors. In Thomas Bäck, David B Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, pages C1.3:1–2. Oxford University Press, Oxford, 1997.

[35] David B Fogel. Some recent important foundational results in evolutionary computation. In K Miettinen, P Neittaamäki, M Mäkelä, and J Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, pages 55–71. John Wiley & Sons, Inc., New York, 1999.

[36] Mitsuo Gen and Runwei Cheng. *Genetic Algorithms and Engineering Design*. John Wiley & Sons, Inc., New York, 1997.

[37] David E Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Reading, MA, 1989.

[38] David E Goldberg. Sizing populations for serial and parallel genetic algorithms. In J David Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 70–79, George Mason University, 1989. Morgan Kaufmann.

[39] David E Goldberg. Zen and the art of genetic algorithms. In J David Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 80–85, George Mason University, 1989. Morgan Kaufmann.

[40] David E Goldberg. The existential pleasures of genetic algorithms. IlliGAL Report 94010, Univeristy of Illinois at Urbana-Champaign, 1994.

[41] David E Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In John J Grefenstette, editor, *Proceedings of the 2nd International Conference on Genetic Algorithms*, pages 41–49. Lawrence Erlbaum Associates, 1987.

[42] David E Goldberg and S Voessner. Optimizing global-local search hybrids. In Annie S Wu, editor, *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 220–228, 1999.

[43] David E Goldberg and Liwei Wang. Adaptive niching via coevolutionary sharing. In D Quagliarella, J Periaux, C Poloni, and G Winter, editors, *Genetic Algorithms and Evolutionary Strategy in Engineering and Computer Science*, pages 21–38. John Wiley & Sons, Inc., New York, 1998.

[44] Jeanine Graf. Interactive evolution in engineering design. In Ian Parmee, editor, *Proceedings of the 2nd International Conference on Adaptive Computing in Engineering Design and Control*, pages 297–299, Plymouth, UK, March 1996. The University of Plymouth.

[45] Garrison W Greenwood, Xiaobo (Sharon) Hu, and Joseph G D'Ambrosio. Fitness functions for multiple objective optimization problems: Combining preferences with Pareto rankings. In Richard K Belew and Michael D Vose, editors, *Foundations of Genetic Algorithms 4*, pages 437–454. Morgan Kaufmann, San Mateo, California, 1997.

[46] John J Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):122–128, 1986.

[47] John J Grefenstette. Incorporating problem specific knowledge into genetic algorithms. In Lawrence Davis, editor, *Genetic Algorithms and Simulated Annealing*, pages 42–60. Morgan Kaufmann, San Mateo, California, 1987.

[48] J Haataja. Using genetic algorithms for optimization: Technology transfer in action. In K Miettinen, P Neittaamäki, M Mäkelä, and J Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, pages 3–22. John Wiley & Sons, Inc., New York, 1999.

[49] G R Harik, F G Lobo, and D E Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):287–297, 1999.

[50] Stephen P Harris and Emmanuel C Ifeachor. Automating IIR filter design by genetic algorithm. In *Proceedings of the 1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 95)*, pages 271–275, Sheffield, UK, 1995. Institution of Electrical Engineers.

[51] William E Hart. A stationary point convergence theory for evolutionary algorithms. In Richard K Belew and Michael D Vose, editors, *Foundations of Genetic Algorithms 4*, pages 325–342. Morgan Kaufmann, San Mateo, California, 1997.

[52] I Harvey, P Husbands, D Cliff, A Thompson, and N Jakobi. Evolutionary robotics: The Sussex approach. *Robotics and Autonomous Systems*, 20:205–224, 1997.

[53] Inman Harvey. Cognition is not computation; evolution is not optimisation. In *Proceedings of the 7th International Conference on Artificial Neural Networks (ICANN97)*, New York, October 1997. Springer Verlag.

[54] Randy L Haupt and Sue Ellen Haupt. *Practical Genetic Algorithms*. John Wiley & Sons, Inc., New York, 1998.

[55] R J Henery. Classification. In D Michie, D J Spiegelhalter, and C C Taylor, editors, *Machine Learning, Neural and Statistical Classification*, pages 6–16. Ellis Horwood, New York, 1994.

[56] Robert Hinterding, Zbigniew Michalewicz, and T C Peachey. Self-adaptive genetic algorithm for numeric functions. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature IV*, pages 420–429, Berlin, Germany, September 22–26 1996. Springer Verlag.

[57] Frank Hoffmeister and Joachim Sprave. Problem-independent handling of constraints by use of metric penalty functions. In Lawrence J Fogel, Peter J Angeline, and Thomas Bäck, editors, *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 289–294, Cambridge, Massachusetts, 1996. The MIT Press.

[58] John H Holland. *Adaptation in Natural and Artificial Systems*. The MIT Press, Cambridge, Massachusetts, 1993.

[59] Jeffrey Horn, David E Goldberg, and Kalyanmoy Deb. Long path problems. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel Problem Solving from Nature III*, pages 149–158, Jerusalem, Israel, October 9–14 1994. Springer Verlag.

[60] Terry Jones and Stephanie Forrest. Genetic algorithms and heuristic search. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 95)*, 1995.

[61] Leila Kallel and Bart Naudts. Candidate longpaths for the simple genetic algorithm. In Wolfgang Banzhaf and Colin Reeves, editors, *Foundations of Genetic Algorithms 5*, pages 27–43. Morgan Kaufmann, San Mateo, California, 1997.

[62] Hillol Kargupta. Gene expression: The missing link in evolutionary computation. In D Quagliarella, J Periaux, C Poloni, and G Winter, editors, *Genetic Algorithms and Evolutionary Strategy in Engineering and Computer Science*, pages 59–84. John Wiley & Sons, Inc., New York, 1998.

[63] Hillol Kargupta and David E Goldberg. SEARCH, blackbox optimization, and sample complexity. In Richard K Belew and Michael D Vose, editors, *Foundations of Genetic Algorithms 4*, pages 291–324. Morgan Kaufmann, San Mateo, California, 1997.

[64] A J Keane and S M Brown. The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques. In Ian Parmee, editor, *Proceedings of the 2nd International Conference on Adaptive Computing in Engineering Design and Control*, pages 107–113, Plymouth, UK, March 1996. The University of Plymouth.

[65] Jong-Hwan Kim and Hyun Myung. Evolutionary programming techniques for constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 1(2):129–140, 1997.

[66] Johannes Krottmaier. *Optimizing Engineering Designs*. McGraw-Hill, London, 1993.

[67] Rajeev Kumar and Peter Rockett. Assessing the convergence of rank-based multiobjective genetic algorithms. In *Proceedings of the 2nd International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 97)*, pages 19–23, Glasgow, UK, 1997. Institution of Electrical Engineers.

[68] David Lane, Franco Malerba, Robert Maxfield, and Luigi Orsenigo. Choice and action. Working Paper 95-01-004, The Santa Fe Institute, November 1994.

[69] D B Lenat and E A Feigenbaum. On the thresholds of knowledge. *Artificial Intelligence*, 47:185–250, 1991.

[70] Gunar E Liepins and Michael D Vose. Deceptiveness and genetic algorithm dynamics. In Gregory J E Rawlins, editor, *Foundations of Genetic Algorithms*, pages 36–50. Morgan Kaufmann, San Mateo, California, 1991.

[71] T J Loredo. From Laplace to supernova SN 1987A: Baysian inference in astrophysics. In P F Fougere, editor, *Maximum Entropy and Baysian Methods*, pages 81–142. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.

[72] Sushil J Louis and Gregory J E Rawlins. Pareto optimality, GA-easiness and deception. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 118–123, University of Illinois at Urbana-Champaign, 1993, July 17-21. Morgan Kaufmann.

[73] Bernard Manderick, Mark de Weger, and Piet Spiessens. The genetic algorithm and the structure of the fitness landscape. In Richard K Belew and Lashon B Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 143–150, San Mateo, CA, 1991. Morgan Kaufmann.

[74] Z Michalewicz, K Deb, M Schmidt, and T H Stidsen. Evolutionary algorithms for engineering applications. In K Miettinen, P Neittaamäki, M Mäkelä, and J Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, pages 73–94. John Wiley & Sons, Inc., New York, 1999.

[75] Zbigniew Michalewicz. Introduction (constraint handling techniques). In Thomas Bäck, David B Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, pages C5.1:1–3. Oxford University Press, Oxford, 1997.

[76] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer Verlag, New York, 1999.

[77] Zbigniew Michalewicz. The significance of the evaluation function in evolutionary algorithms. In Lawrence Davis, Kenneth De Jong, Michael D Vose, and L Darrell Whitley, editors, *Evolutionary Algorithms*, pages 151–166. Springer Verlag, New York, 1999.

[78] Melanie Mitchell. *An Introduction to Genetic Algorithms.* The MIT Press, Cambridge, Massachusetts, 1996.

[79] Harry Nelson and Robert Jurmain. *Introduction to Physical Anthropology.* West Publishing Company, St Paul, Minnesota, 1988.

[80] Karoly Pal. Selection schemes with spatial isolation for genetic optimization. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel*

Problem Solving from Nature III, pages 170–179, Jerusalem, Israel, October 9–14 1994. Springer Verlag.

[81] Jan Paredis. Co-evolutionary constraint satisfaction. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel Problem Solving from Nature III*, pages 46–55, Jerusalem, Israel, October 9–14 1994. Springer Verlag.

[82] I C Parmee. The maintenance of search diversity for effective design space decomposition using cluster-oriented genetic algorithms (COGAs) and multiagent strategies (GAANT). In Ian Parmee, editor, *Proceedings of the 2nd International Conference on Adaptive Computing in Engineering Design and Control*, pages 128–138, Plymouth, UK, March 1996. The University of Plymouth.

[83] I C Parmee. Designing the evolutionary design station. In U Lindemann, H Birkhofer, H Meerkamm, and S Vajna, editors, *Proceedings of the 12th International Conference on Engineering Design*, volume 2, pages 1031–1034, Munich, Germany, August 1999. Technische Universität München.

[84] I C Parmee, M Johnson, and S Burt. Techniques to aid global search in engineering design. In *Proceedings of IEAAIE '94*, 1994.

[85] Ray Paton. Introduction to computing with biological metaphors. In Ray Paton, editor, *Computing with Biological Metaphors*, pages 1–8. Chapman and Hall, London, UK, 1994.

[86] Charles C Peck and Atam P Dhawan. Genetic algorithms as global randomsearch methods: An alternative perspective. *Evolutionary Computation*, 3(1):39–80, 1995.

[87] Martin Pelikan, David E Goldberg, and Erick Cantú-Paz. BOA: The Bayesian optimization algorithm. In Wolfgang Banzhaf, Jason Daida, Agoston E Eiben,

Max H Garzon, Vasant Honavar, Mark Jakiela, and Robert E Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume I, pages 525–532, Orlando, 1999. Morgan Kaufmann.

[88] David Powell and M Skolnick. Using genetic algorithms in engineering design optimization with non-linear constraints. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 424–431, University of Illinois at Urbana-Champaign, 1993, July 17-21. Morgan Kaufmann.

[89] Xiaofeng Qi and Francesco Palmieri. The diversification role of crossover in the genetic algorithm. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 133–137. Morgan Kaufmann, University of Illinois at Urbana-Champaign, 1993, July 17-21.

[90] Pearce R. Constraint resolution in genetic algorithms. In A M S Zalzala and P J Fleming, editors, *Genetic Algorithms in Engineering Systems*, pages 79–98. Institution of Electrical Engineers, London, 1997.

[91] Soraya Rana and L Darrell Whitley. Search, binary representations and counting optima. In Lawrence Davis, Kenneth De Jong, Michael D Vose, and L Darrell Whitley, editors, *Evolutionary Algorithms*, pages 177–189. Springer Verlag, New York, 1999.

[92] Singiresu S Rao. *Engineering Optimization: Theory and Practice (Third Edition)*. John Wiley & Sons, Inc., New York, 1996.

[93] Khaled Rasheed and Haym Hirsh. Learning to be selective in genetic-algorithm-based design optimization. *Artificial Intelligence for Engineering Design Analysis and Manufacturing*, 13:157–169, 1999.

[94] Caroline Ravisé, Michele Sebag, and Marc Schoenauer. Induction-based control of genetic algorithms. In J M Alliot, E Lutton, E Ronald, M Schoenauer, and D Snyers, editors, *Artificial Evolution*, pages 100–119, Brest, France, September 1995. Springer Verlag.

[95] I Rechenberg. Evolutionary experimentation. In David B Fogel, editor, *Evolutionary Computation. The Fossil Record*, pages 297–300. The Institute of Electrical and Electronic Engineers Press, New York, 1998.

[96] Colin R Reeves. Using genetic algorithms with small populations. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 92–99, University of Illinois at Urbana-Champaign, 1993, July 17-21. Morgan Kaufmann.

[97] Colin R Reeves and Christine C Wright. Genetic algorithms and the design of experiments. In Lawrence Davis, Kenneth De Jong, Michael D Vose, and L Darrell Whitley, editors, *Evolutionary Algorithms*, pages 207–226. Springer Verlag, New York, 1999.

[98] Jon T Richardson, Mark R Palmer, Gunar Liepins, and Mike Hilliard. Some guidelines for genetic algorithms with penalty functions. In J David Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 191–197, George Mason University, 1989. Morgan Kaufmann.

[99] Simon Ronald. Robust encodings in genetic algorithms. In D Dasgupta and Z Michalewicz, editors, *Evolutionary Algorithms in Engineering Design*, pages 29–44. Springer Verlag, New York, 1997.

[100] Rajkumar Roy and Ian Parmee. Adaptive restricted tournament selection for the identification of multiple sub-optima in a multi-modal function. In T C Foga-

rty, editor, *Evolutionary Computing*, volume 1143 of *Lecture Notes in Computer Science*, pages 236–256. Springer Verlag, New York, 1996.

[101] Günter Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5(1):96–101, January 1994.

[102] Ralf Salomon. Performance degradation of genetic algorithms under coordinate rotation. In Lawrence J Fogel, Peter J Angeline, and Thomas Bäck, editors, *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 155–161, Cambridge, Massachusetts, 1996. The MIT Press.

[103] B Sareni and L Krähenbühl. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106, 1998.

[104] J David Schaffer, Richard A Caruana, Larry J Eshelman, and Rajarshi Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In J David Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 51–60, George Mason University, 1989. Morgan Kaufmann.

[105] Marc Schoenauer and Spyros Xanthakis. Constrained GA optimization. In Richard K Belew and Lashon B Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, San Mateo, CA, 1991. Morgan Kaufmann.

[106] L I Sedov and A G Volkovets (translation). *Similarity and Dimensional Methods in Mechanics*. CRC Press, Florida, USA, 1993.

[107] Herbert A Simon. *The Sciences of the Artificial, 3rd Edition*. The MIT Press, Cambridge, Massachusetts, 1996.

[108] Alice E Smith and David M Tate. Genetic optimization using a penalty function. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 499–503, University of Illinois at Urbana-Champaign, 1993, July 17-21. Morgan Kaufmann.

[109] Michael Syrjakow and Helena Szczerbicka. Efficient parameter optimization of direct global and local search methods. In Lawrence Davis, Kenneth De Jong, Michael D Vose, and L Darrell Whitley, editors, *Evolutionary Algorithms*, pages 227–249. Springer Verlag, New York, 1999.

[110] Dirk Thierens and David Goldberg. Convergence models of genetic algorithms selection schemes. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel Problem Solving from Nature III*, pages 119–129, Jerusalem, Israel, October 9–14 1994. Springer Verlag.

[111] Alasdair Turner, David Corne, Graeme Ritchie, and Peter Ross. Obtaining multiple distinct solutions with genetic algorithm niching methods. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature IV*, pages 451–460, Berlin, Germany, September 22–26 1996. Springer Verlag.

[112] Andrew Tuson and Peter Ross. Co-evolution of operator settings in genetic algorithms. In Terence C Fogarty, editor, *Evolutionary Computing*, pages 286–296, New York, April 1996. Springer Verlag.

[113] G Venturini. GA consistently deceptive functions are not challenging problems. In *Proceedings of the 1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 95)*, pages 357–364, Sheffield, UK, 1995. Institution of Electrical Engineers.

[114] Gilles Venturini. Towards a genetic theory of easy and hard functions. In J M Alliot, E Lutton, E Ronald, M Schoenauer, and D Snyers, editors, *Artificial Evolution*, pages 54–68, Brest, France, September 1995. Springer Verlag.

[115] Michael D Vose. Logarithmic convergence of random heuristic search. *Evolutionary Computation*, 4(4):395–404, 1997.

[116] Gang Wang, Terrence Dexter, William F Punch III, and Erik D Goodman. Simultaneous multi-level evolution. Technical Report GARAGe96-03-01, Michican State University, Genetic Algorithms Research and Applications Group (GARAGe), 1996.

[117] Gang Wang, Erik D Goodman, and Willian F Punch III. On the optimization of a class of blackbox optimization algorithms. In *Proceedings of the IEEE International Conference on Tools for Artificial Intelligence*, November 1997.

[118] Karsten Weiker and Nicole Weiker. Locality vs randomness—dependence of operator quality on the search state. In Wolfgang Banzhaf and Colin Reeves, editors, *Foundations of Genetic Algorithms 5*, pages 147–163. Morgan Kaufmann, San Mateo, California, 1997.

[119] David H Wolpert and William G Macready. No free lunch theorems for search. Working Paper 95-02-010, The Santa Fe Institute, February 1995.