PERGAMON

Engineering Applications of Artificial Intelligence 12 (1999) 639–654

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

A comparison of various optimization algorithms on a multilevel problem

M.A. El-Beltagy*, A.J. Keane

Department of Mechanical Engineering, University of Southampton, Highfield, Southampton SO17 1BJ, UK

Received 1 August 1997; accepted 1 June 1999

Abstract

In many problems in science and engineering, there are often a number of computational models that can be used to simulate the problem at hand. Models of physical systems can differ according to computational cost, accuracy and precision. This paper presents the concept of multilevel optimization, where different models of the problem are used in combination. This initial study compares several strategies for combining fast evaluations of limited accuracy with a few accurate calculations. It also attempts to show how different optimizers work under these different combination strategies. A specially designed test function is used to carry out these comparisons. Of the proposed strategies and optimisers, a sequential mixing strategy applied to a genetic algorithm with clustering gives the best results. This paper highlights the need to develop specialized optimization algorithms for this kind of problem. © 1999 Published by Elsevier Science Ltd. All rights reserved.

Keywords: Optimization; Multilevel problems; Genetic algorithms

1. Introduction

In many optimization problems there may exist a number of different ways in which a particular problem is modelled. Some methods may be quite elaborate in their representation, while others involve a simplification of the problem, with the former being more accurate but at the same time more computationally expensive than the latter. It is therefore important to understand how a significant number of less accurate evaluations may be integrated with fewer accurate ones, to arrive at an optimum design.

The multiplicity of computational models for a given object of simulation may arise from at least three main causes. It could be due to different mathematical formulations being used to construct the model, such as Euler and Navier–Stokes approximations in computational fluid dynamics (CFD). It could also be due to different discretization limits within one formulation, such as mesh densities in finite elements analysis (FEA). Finally, it may come from the availability of approximate empirical models such as neural networks or response surfaces.

The term 'multilevel optimization' (MLO) is used here to denote the process of optimizing such a multiplicity of models, where each level is essentially one of these models.

In contrast to the static optimization problem

$$f(\boldsymbol{x}) \to \text{opt}, \quad (\boldsymbol{x} \in M),$$

the multilevel optimization problem may be stated as

$$f_1(\boldsymbol{x}) \to \text{opt}, \quad (\boldsymbol{x} \in M),$$

where $f_1(\boldsymbol{x})$ is the most accurate function and there exist many $f_k(x)$ models where $k = 1, \ldots, L$. The levels are such that $f_i(x)$ is more accurate and computationally expensive than $f_j(x)$ for $i < j$. During the optimization, it is usually the case that the computationally expensive function levels cannot be used often. There

* Corresponding author. Tel.: +44-1703-592369; fax: +44-1703-593392.

E-mail addresses: M.A.El-beltagy@soton.ac.uk (M.A. El-Beltagy), Andy.Keane@soton.ac.uk (A.J. Keane).

are many possible ways in which such approximate and accurate representations can be integrated. In this paper, three strategies are attempted: sequential multilevel optimization, gradually mixed multilevel optimization, and totally mixed multilevel optimization. These integration methods are explained in subsequent sections. The main aim has been to see how different optimization methods work, using these strategies, paying particular attention to genetic algorithms (GAs).

In Holland's (1975) introduction to genetic algorithms as a means to design and implement robust adaptive systems, he emphasized that these systems should be able to handle uncertainty and change in the environments in which they operate, and should be able to self-adapt over time. Even so, most work with GAs has been carried out using time-invariant environments rather than the dynamic ones described in Holland's original works (Lund, 1994). Moreover, the details of how the GA goes about searching a given landscape are not well understood. Mitchel et al. (1991) state:

> ...there is no firm theoretical grounding for what is perhaps the most prevalent 'folk theorem' about GAs — that they will outperform hillclimbers and other common search and optimization techniques on a wide spectrum of difficult problems, because crossover allows the powerful combination of partial solutions.

The position is even more obscure when the environment may suddenly change, as is the case in multi-level optimization. Nonetheless, it is hoped that by using suitable strategies, the notion of a 'combination of partial solutions' can transcend the boundaries of different levels.

This paper is arranged as follows. The next section briefly overviews previous work related to multilevel optimization using traditional as well as evolutionary approaches. Section 3 describes the multilevel optimization test function proposed here. Sections 4–6 describe the Sequential, Gradually Mixed, and Totally Mixed Multilevel optimization strategies tested. Section 7 details the optimization methods used in this study. Section 8 gives an overview of niching as used in GA and its intuitive advantages for this type of problem. Section 9 details the experimental results obtained, and Section 10 highlights the main findings. The paper closes with a brief conclusion and a discussion of future work.

2. Previous work related to multilevel optimization

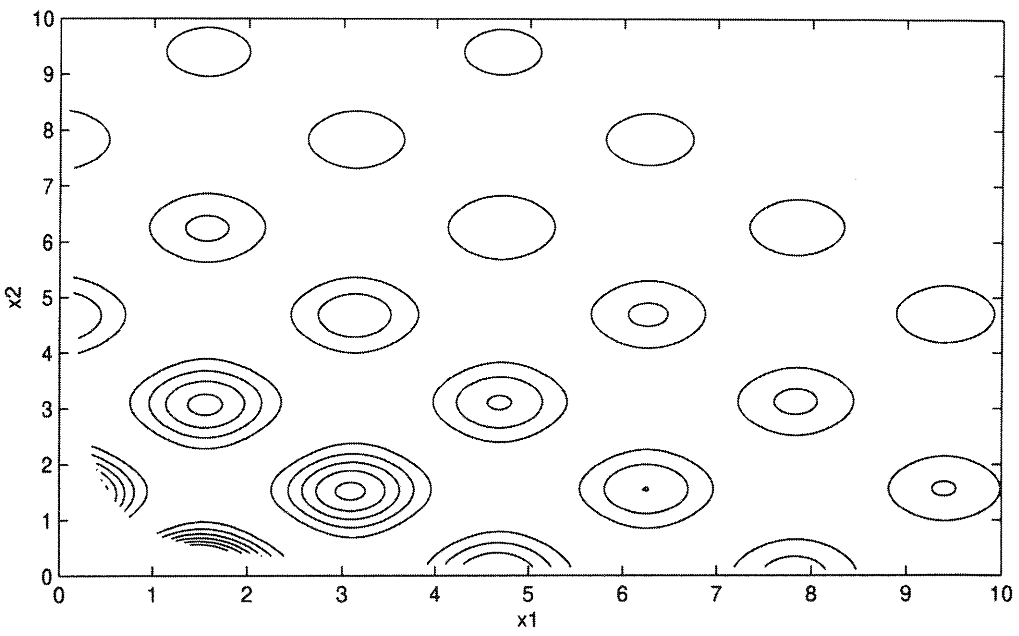
Multilevel optimization can be regarded as an instance of searching in a dynamic environment. In such environments, the objective function value for a given $\boldsymbol{x}$ does not remain constant with time. Although the nature of the changes that occur when one model is substituted for another are not strictly stochastic, this process is akin to dealing with a noisy function.

A number of workers have studied the optimization of noisy functions over the years. In these studies, the objective function value is presumed to be governed by a certain stochastic distribution for any given problem parameter vector $\boldsymbol{x}$. One classical approach that has been attempted for dealing with such functions is that based on the simplex method of Nelder and Mead (NM) (Nelder and Meade, 1965). This is a strategy for *unconstrained* optimization using local exploration. It was shown that in the optimization of noisy functions that this was unable to cope with very simple problems with high dimensions (Elster and Neumaier, 1995). However, more elaborate methods have been proposed by Elster and Neumaier (1993), using quadratic models and a restriction of the evaluation points to successively refined grids. This method works well for low-dimensional, bound-constrained problems. Extensive tests have shown the algorithm to be of comparable performance with the quasi-Newton method in the noiseless case, and much more robust than NM, in the noisy case.

Another approach for dealing with noisy data is based on multi-point approximations, which have been used for the optimization of noisy finite element problems. This approach works by successively fitting a surface in the problem space to approximate function evaluations (Van Keulen et al., 1995; Wang and Grandhi, 1995). It is akin to fitting a curve through a series of experimental data that are subject to random errors, so as to observe the underlying trend in the readings. In most of this work, traditional gradient-based optimizers were then used on the approximate surface, which was generally smoother than the underlying function. A thorough survey of related approximation concepts used in Multi-Disciplinary Optimization (MDO) can be found in Sobieszczanski-Sobieski and Haftka's (1996) survey.

Dunham et al. (1963) appear to be the first to have addressed the problem of multilevel optimization within an evolutionary optimization context. They worked with a two-level problem. In their study, they used an approximate model most of the time, using the accurate/computationally expensive model only at the final stages of refinement.

Using GAs, Grefenstette and Fitzpatrick (1985) tried to answer the question: 'Given a fixed amount of computation, is it better to devote substantial effort to seeking highly accurate evaluations or to obtain quick, rough evaluations and run the GA for many more generations?' Using statistical sampling theory, the



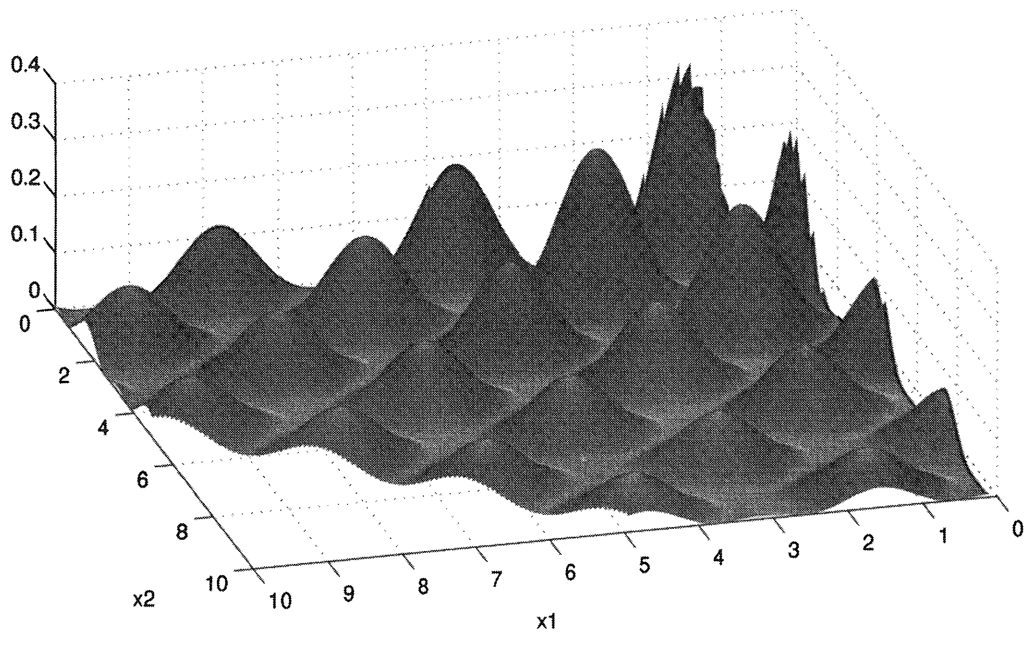
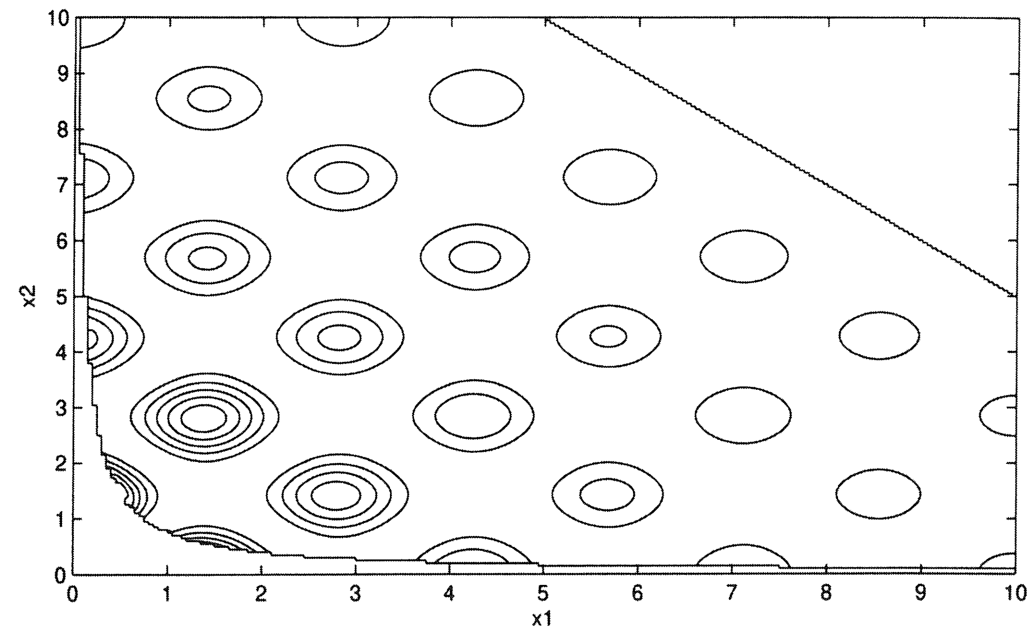


Fig. 1. A contour map and a 3D plot of bump for $n = 2$.

authors showed that the optimization process would proceed more quickly if less time was spent on individual accurate evaluations, and instead the number of generations performed was increased.

A method based on model selection was proposed for hillclimbing search by Ellman et al. (1993). This method works by generating error estimates for different models (different levels of accuracy) of a yacht hull. Based on the estimates, it decides which model (level) to choose from.

Miller and Goldberg (1995a,b) constructed a model on how selection schemes in genetic algorithms respond to the varying effect of noise in the onemax domain. Aizawa and Wah (1993) presented an algorithm for adjusting the configuration parameters of genetic algorithms that operate in noisy environments.



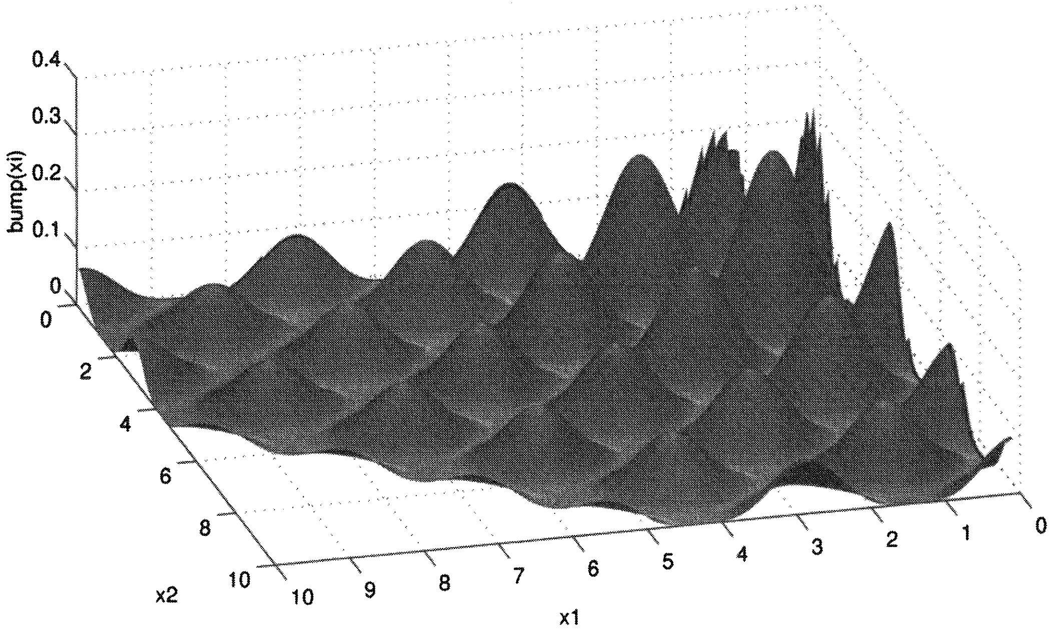


Fig. 2. A contour map and a 3D plot of the modified 2D bump ($n = 2, \alpha = 1.1, \beta = 0$).

Perhaps the most promising GA work in this area is based on the concept of an Injection Island architecture (iiGA) (Eby et al., 1998; Parmee and Vekeria, 1997). The iiGA works by using a series of subpopulations representing different levels of representations, and evolving them upon separate islands. Exchange of chromosomes between the low- and high-accuracy islands occurs at set numbers of evaluations. The best individuals from the less accurate island migrate, and replace the worst individuals in the more accurate island.

Angeline (1997) presented the case for tracking extrema in dynamic environments using evolutionary programming. Bäck (1998) presented a similar study using evolutionary strategies. Most recently, Nissen and Propach (1998) executed a comparative study on the robustness of population- and point-based search heuristics in the presence of noise.

None of the work cited has compared a wide range of different optimization techniques within the context of multilevel optimization.

3. A multilevel optimization test function

In order to study how the different optimisers work in a multilevel environment, it is first necessary to construct a suitable test function. This test function should have parameters that allow the variation of the local optima positions and properties from one level to the next to be controlled. While no claim to completeness is made, the test function presented here is easy to evaluate but hard to optimize, while having many local optima and arbitrary dimensions. It is based on the 'bump' function which has been fairly widely used in the genetic algorithms literature for comparative studies.

3.1. The bump problem

The 'bump' problem, introduced by Keane (1994), is very hard for most optimizers to deal with. It is quite smooth but contains many peaks, all of similar heights. Moreover, like many problems in engineering, its optimal value is defined by the presence of a constraint boundary.

The problem is defined as

$$\text{maximize} \frac{\text{abs}\left(\sum_{i=1}^{n}\cos^4(x_i) - 2\prod_{i=1}^{n}\cos^2(x_i)\right)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \tag{1}$$

for

$$0 < x_i < 10 \quad i = 1, \ldots, n \tag{2}$$

subject to

$$\prod_{i=1}^{n} x_i > 0.75 \quad \text{and} \quad \sum_{i=1}^{n} x_i < 15n/2 \tag{3}$$

starting from

$$x_i = 5, \quad i = 1, \ldots, n.$$

Fig. 1 shows a contour map and a 3D plot of bump for $n = 2$. An interesting feature of this function is that the surface is nearly but not quite symmetrical in $x_1 = x_2$, so that the peaks always occur in pairs, but with one always bigger than its sibling. The global optimum is defined by the product constraint. When the problem is generalized for n greater than two, it becomes even more demanding with families of similar peaks occurring within a highly complex constraint surface. These properties of the bump problem have made it suitable for the study of GA performance and optimising GA control parameters (Keane, 1995a; Michalewicz, 1996), as well as the control parameters of other evolutionary optimization methods (Keane, 1995b).

3.2. The extended bump

The bump function has been extended for multilevel optimization by the introduction of two distortion parameters. Different degrees of distortions are used to simulate different levels of accuracy. Eq. (1) was generalized to have a frequency shift parameter α and a spatial shift parameter β:

$$\text{maximize} \frac{\text{abs}\left(\sum_{i=1}^{n}\cos^4(\alpha(x_i+\beta)) - 2\prod_{i=1}^{n}\cos^2(\alpha(x_i+\beta))\right)}{\sqrt{\sum_{i=1}^{n} i(x_i+\beta)^2}}. \tag{4}$$

The α frequency shift parameter distorts the bump, spreading out the peaks ($\alpha < 1$) or making them closer ($\alpha > 1$). The spatial shift parameter β just shifts the peaks of bump in x_i. Hence, the undistorted bump becomes one in which $\alpha = 1$ and $\beta = 0$. Fig. 2 shows a distorted bump function for $\alpha = 1.1$ and $\beta = 0$; the number of peaks is clearly greater than in Fig. 1.

4. Sequential multilevel optimization

Perhaps the most obvious strategy to adopt when dealing with multilevel optimization is to use the increasingly accurate function in a simple sequential manner. In this approach, the optimization is started using the least accurate level of representation. Then, after a certain set number of function evaluations, the optimization on this level is stopped, and the results are used as starting points for the next, more accurate level. (In the case of population-based methods, the final population is used and re-evaluated using the more accurate function, becoming the initial population for the next level.) This process is carried on sequentially, and the number of function evaluations is decreased from one level to the next until the most accurate level is reached, where fewest function evaluations are carried out. The number of function evaluations carried out at each level would ordinarily be chosen to roughly equalize the computational effort expended at each level. For the bump problem con-

Table 1
Number of evaluations at each level and corresponding distortion parameters

Optimization level	1	2	3
Number of evaluations	12,500	2500	500
α	1.5	1.1	1
β	0.5	0.1	0

sidered here, the first, least accurate level corresponds to maximum distortion (high values of α and/or β). The optimization then proceeds through an intermediate level ($\alpha \longrightarrow 1$ and $\beta \longrightarrow 0$), after which the final accurate representation is reached ($\alpha = 1$ and $\beta = 0$).

As has also already been mentioned, the number of function evaluations per level decreases from the least accurate level to the most accurate one. This reduction mimics the situation where more refined models become more computationally expensive, and hence, only a limited number of evaluations would be afforded. Details of the number of generations used here for each level are shown in Table 1. Assuming that an equal amount of effort were expended at each level, this supposes that the true function evaluation is 25 times more expensive than that for $\alpha = 1.5$ and/or $\beta = 0.5$, and five times more expensive than for $\alpha = 1.1$ and/or $\beta = 0.1$.

5. Gradually mixed multilevel optimization

In the next strategy, the optimization procedure is carried out with the three levels stochastically mixed throughout the optimization process. The probability of using a particular level varies with the number of function evaluations as shown in Fig. 3.

Using this scheme, the first 10,200 evaluations are carried out using the first level. During the next 4600 evaluations, the first and second levels are mixed gradually. This is followed by 400 evaluations, where the second and third levels are mixed until finally, the

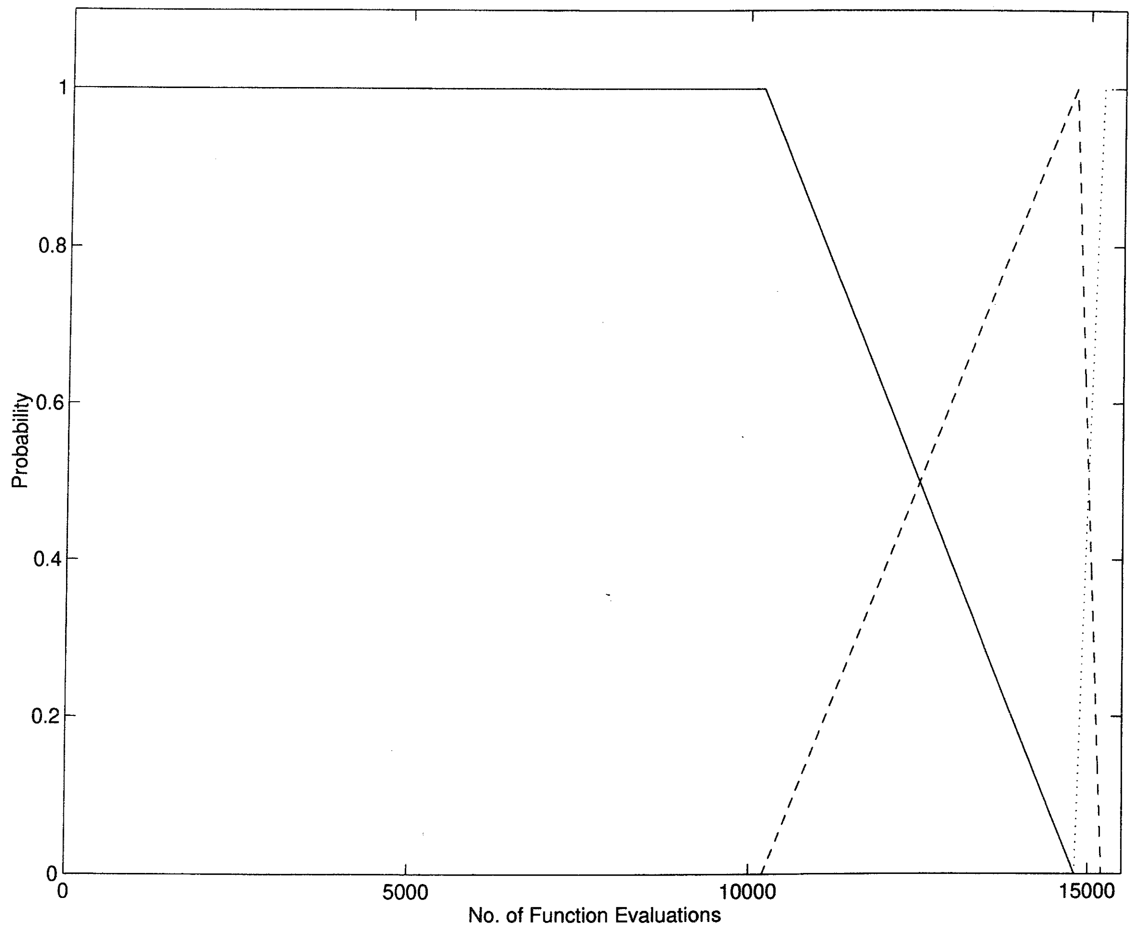


Fig. 3. Probability of selecting a level after a given number of function evaluations. The first level is solid, the second dashed, and the third dotted.

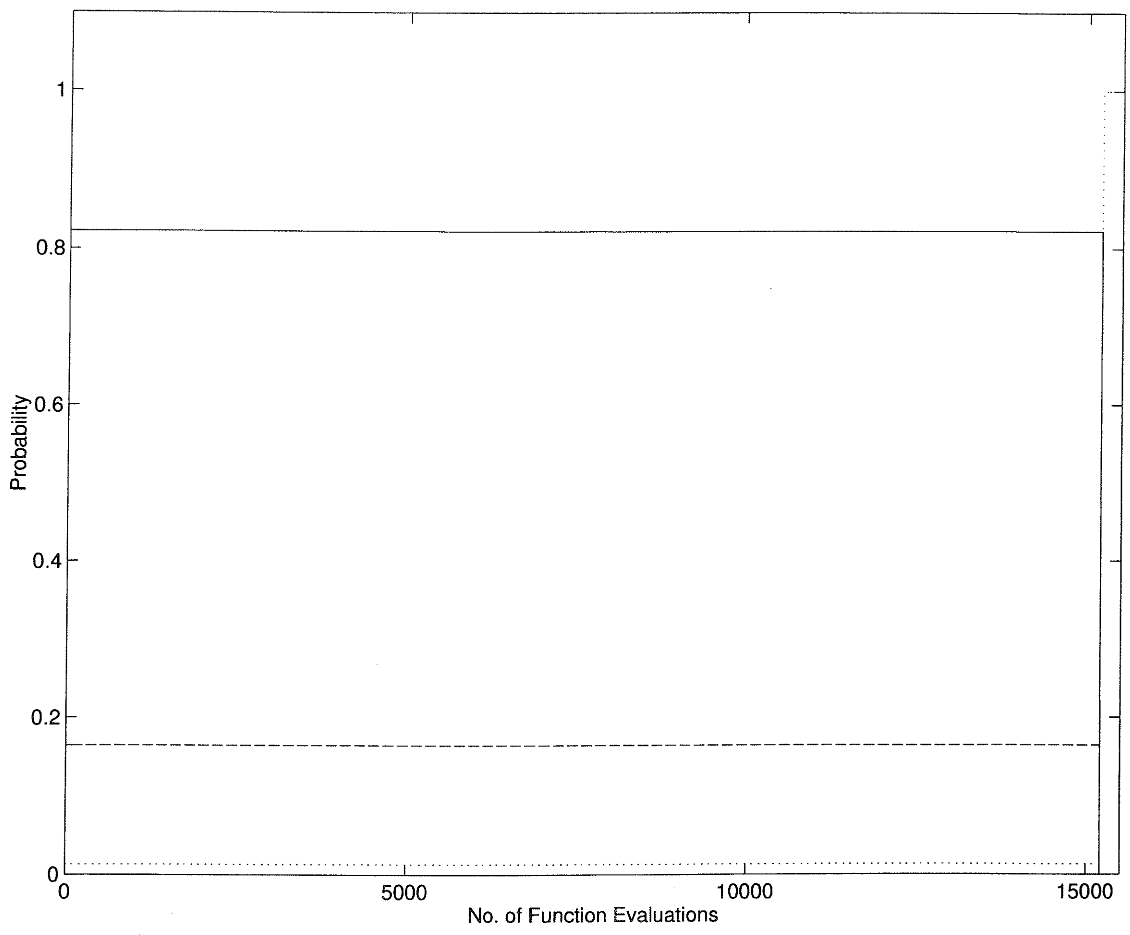


Fig. 4. Probability of selecting a level after a given number function evaluations. The first level is solid, the second dashed, and the third dotted.

last 300 evaluations are carried out solely at the third and most accurate level. The scheme is constructed such that the average overall computational cost is equal to that in the sequential mixing scheme.

6. Totally mixed multilevel optimization

The last strategy tested here is based on a totally mixed multilevel approach, where the probability of using a particular level is constant, throughout most of the optimization process. However, to ensure stability in the results towards the very end of the optimization, only the most accurate level is used. In this case, the first level has a probability of 82.2%, the second has a probability of 16.5%, and the third and most accurate level has a probability of 1.3%. In the last 300 evaluations, the third level probability becomes 100%, see Fig. 4. This scheme is set up such that the computational cost on average is the same as in the previous two schemes.

7. The optimization methods

A primary aim of this study was to consider the utility of a wide range of search methods for multilevel problems. Consequently, a rather large number of optimization methods have been used in these tests. Almost all the methods used are available in a single design-exploration system developed by Keane (1995c), called OPTIONS. Among the different methods in OPTIONS, some are from standard libraries (e.g., Schwefel, 1995; Siddall, 1982), while others have been specially developed for the suite, based on ideas culled from the literature. OPTIONS currently contains the following methods:

- A genetic algorithm based on clustering and sharing (GACS) (Yin and Germay, 1993);
- Adaptive random search (Adrans) (Siddall, 1982);
- The Davidon–Fletcher–Powell strategy (David) (Siddall, 1982);
- Fletcher's 1972 method (Fletch) (Siddall, 1982);
- Powell direct search method (PDS) (Siddall, 1982);

- Hooke and Jeeves direct search as implemented by Siddall (Seek) (Siddall, 1982);
- The simplex strategy of Nelder and Meade (1965) as implemented by Siddall (Simplx) (Siddall, 1982);
- The method of successive linear approximation (Approx) (Siddall, 1982);
- Random exploration with shrinkage (Random) (Siddall, 1982);
- NAg routine E04UCF, a sequential quadratic programming method (NAg) (NAg E04UCF, 1990);
- A bit climbing algorithm (BClimb) (Davis, 1991);
- A dynamic hill-climbing algorithm (DHClimb) (Yuret and de la Maza, 1993);
- A population-based incremental learning algorithm (PBIL) (Baluja, 1994);
- The Powell routine as implemented in the Numerical Recipes cookbook (Num Rcp) (Press et al., 1986);
- Repeated application of a one-dimensional Fibonacci search (FIBO) (Schwefel, 1995);
- Repeated application of a one-dimensional Golden section search (Golden) (Schwefel, 1995);
- Repeated application of a one-dimensional Lagrangian interpolation search (LAGR) (Schwefel, 1995);
- Hooke and Jeeves direct search as implemented by Schwefel (HandJ) (Schwefel, 1995);
- Rosenbrock's rotating co-ordinated search (ROSE) (Rosenbrock, 1960);
- The strategy of Davis, Swan and Campey, with Gram–Schmidt orthogonalization (DSCG) (Schwefel, 1995);
- The strategy of Davis, Swan, and Campey with Palmer orthogonalization (DSCP) (Schwefel, 1995);
- Powell's strategy of conjugate directions (Powell) (Schwefel, 1995);
- The Davidon–Fletcher–Powell strategy (DFPS) (Schwefel, 1995);
- The simplex strategy of Nelder and Meade (1965) as implemented by Schwefel (Simplex) (Schwefel, 1995);
- The complex strategy of Box (1965) (Complex) (Schwefel, 1995);
- Schwefel's two-membered evolution strategy (2MES) (Schwefel, 1995);
- Schwefel's multi-membered evolution strategy (MMES) (Schwefel, 1995);
- Simulated annealing (SA) (Kirkpatrick et al., 1983);
- Evolutionary programming (EP) (Fogel, 1993);
- An evolution strategy based on the earlier work of Bäck et al. (1991) (ES).

In addition, two more GAs were used outside OPTIONS:

- A simple haploid GA (Goldberg, 1989) (SGA) ;
- A niching GA using phenotypic sharing (Deb and Goldberg, 1989; Deb, 1989) (NGA).

8. The advantages of niching and clustering

In natural terms, a niche is viewed as an organism's environment, while a species is a collection of organisms with similar features. The subdivision of an environment based on an organism's role reduces interspecies competition for environmental resources, and this reduction in competition helps stable sub-populations to form around different niches in the environment (Deb and Goldberg, 1989). This division into several sub-populations/species may be useful when there is sudden environmental change, as one of the species may be very successful in the new environment, while others may perish. Of course, in GA/optimization terms, the organism is analogous to an individual function evaluation, while the environment is analogous to the function being optimized.

Now, for many optimization problems there exist multiple peaks within the parameter search space: the bump function is an example of such a problem. A simple GA cannot readily maintain stable populations at the different optima of such functions. On the other hand, a GA with niching can force the population to be distributed over many peaks in the parameter space.

The use of niching therefore seems to be intuitively advantageous when the optimization is carried out over several levels in a sequential manner. In the more approximate/distorted/cheaper function representation, the GA with niching distributes the population over 'promising' areas of the (distorted) search space. When this population is placed in the more accurate/expensive function/environment, there is then a higher chance of some of the groupings of the population being nearer to the truly good areas of the space, than if only one such area is passed on to the next level of representation.

To induce niche-like behaviour in genetic search, sharing functions are normally used. Sharing functions degrade an individual's fitness proportional to the number of other members in its neighbourhood. The amount of sharing contributed by each individual to its neighbour depends on the proximity of the two, so that the closer the individuals are, the more degradation there is.

The shared fitness of an individual i is given by

$$\text{Shared fitness} = \frac{\text{True fitness}}{\sum_j s(d_{ij})}. \tag{5}$$

Table 2
Sequential multilevel optimization of a 2D bump

	Alpha (1)	Beta (2)	Alpha and beta (3)	Average	Average no. of steps
NGA	0.3055	0.3008	0.2890	0.2984	15500
GACS	0.2767	0.2739	0.3082	0.2863	15500
SA	0.2907	0.2852	0.2736	0.2832	15500
SGA	0.2697	0.2819	0.2862	0.2793	15500
DHClimb	0.2980	0.2674	0.2722	0.2792	15526
MM_ES	0.2805	0.2775	0.2726	0.2768	5166
EP	0.2839	0.2615	0.2745	0.2733	15500
Bclimb	0.2626	0.2432	0.2679	0.2579	15500
PBIL	0.1756	0.1916	0.3464	0.2379	15500
2MES	0.2051	0.2175	0.2322	0.2183	325
ES	0.2206	0.2005	0.2179	0.2130	15500
FIBO	0.3229	0.1337	0.1551	0.2039	311
Golden	0.3228	0.1337	0.1551	0.2038	315
Complex	0.1031	0.2022	0.2145	0.1733	656
HandJ	0.0868	0.2629	0.1034	0.1510	392
DSCG	0.0708	0.1094	0.2629	0.1477	131
DSCP	0.0708	0.1094	0.2629	0.1477	133
yLAGR	0.0708	0.2629	0.1034	0.1457	174
Adrans	0.0014	0.0951	0.3352	0.1439	15959
Random	0.1425	0.1425	0.1425	0.1425	922

The sharing function $s(d_{ij})$ may be defined as

$$s(d_{ij}) = \begin{cases} 1 - (d_{ij}/\sigma_{\text{share}}) & \text{if } d_{ij} < \sigma_{\text{share}} \\ 0 & \text{otherwise} \end{cases}. \qquad (6)$$

The distance metric d_{ij} indicates the proximity of two individuals. Using phenotypic sharing (Deb and Goldberg, 1989) in a problem with p-dimensional space and q assumed peaks, the equivalent normalized distance measure d_{ij} between the ith and jth individuals is such as (Deb, 1989)

$$d_{ij} = \sqrt{\sum_{k=1}^{p}\left(\frac{x_{k,i} - x_{k,j}}{x_{k,\max} - x_{k,\min}}\right)^2} \qquad (7)$$

where

$x_{k,i}$ = kth parameter of individual i

$x_{k,j}$ = kth parameter of individual j

$x_{k,\max}$ = maximum allowable value for kth parameter

$x_{k,\min}$ = minimum allowable value for kth parameter.

The limiting distance between individuals to partake in sharing is given as $\sigma_{\text{share}} = 0.5q^{(-1/p)}$.

This method has been applied with good results in several applications (e.g., KrishnaKumar et al., 1994). A difficulty of the method is that for good results, prior knowledge is required of the number of peaks in the solution space. In most applications, this information is not readily available.

Another niche-forming method based on MacQueen's adaptive KMEAN clustering algorithms can be effective at revealing unknown multimodal function structures, and is able to maintain subpopulation diversity (Yin and Germay, 1993). This method established analogies between clusters and niches in the following way: the GA population is divided, using the KMEAN algorithm, into clusters that have similar properties. The members of each cluster are then penalised according to the number of members each cluster has, and how far it lies from the cluster centre. Using this approach, it is also possible to restrict the crossover process that forms the heart of the GA, so that large successful clusters mix mainly with themselves. This speeds up convergence, since radical new ideas are prevented from contaminating such sub-pools (Keane, 1995b). Following this approach, the number of clusters is not fixed a priori, but is determined by the algorithm itself during processing. It is, however, quite computationally expensive for large population sizes (>200).

9. Experiments and results

Experiments have been performed for both the 2D and 20D bump functions. The optimization was carried out for (1) varying α; (2) varying β; (3) varying both α and β, all using the values and limits on function evaluations for the levels shown in Table 1. For the stochastic techniques BClimb, PBIL, SGA, NGA, GACS, SA, EP, ES, DHClimb, 2MES and MMES, the results were averaged over 30 optimization runs.

Table 3
Sequential multilevel optimization of a 20D bump

	Alpha (1)	Beta (2)	Alpha and beta (3)	Average	Average no. of steps
GACS	0.5227	0.5951	0.5306	0.5495	15500
DHClimb	0.5379	0.4968	0.5468	0.5272	15933
PBIL	0.4417	0.5601	0.4416	0.4811	15500
EP	0.4222	0.4820	0.4209	0.4417	15500
SA	0.4004	0.4081	0.3880	0.3988	15500
Bclimb	0.4502	0.3343	0.3759	0.3868	15425
2MES	0.2850	0.2549	0.2784	0.2728	3238
MM_ES	0.2936	0.2132	0.2683	0.2584	11913
NGA	0.2346	0.2460	0.2481	0.2429	15500
Adrans	0.2542	0.3123	0.1610	0.2425	15870
SGA	0.2441	0.2329	0.2354	0.2375	15500
ES	0.1721	0.1866	0.1660	0.1749	15500
Simplex	0.0193	0.0468	0.0860	0.0507	15513
Powel	0.0154	0.0154	0.0154	0.0154	18
DSCG	0.0078	0.0078	0.0078	0.0078	15514
DSCP	0.0070	0.0070	0.0070	0.0070	15504
Complex	0.0032	0.0029	0.0028	0.0029	3
PDS	0.0019	0.0019	0.0019	0.0019	18
Jo	0.0019	0.0019	0.0019	0.0019	66
Simplx	0.0018	0.0018	0.0018	0.0018	66
SEEK	0.0018	0.0018	0.0018	0.0018	1269

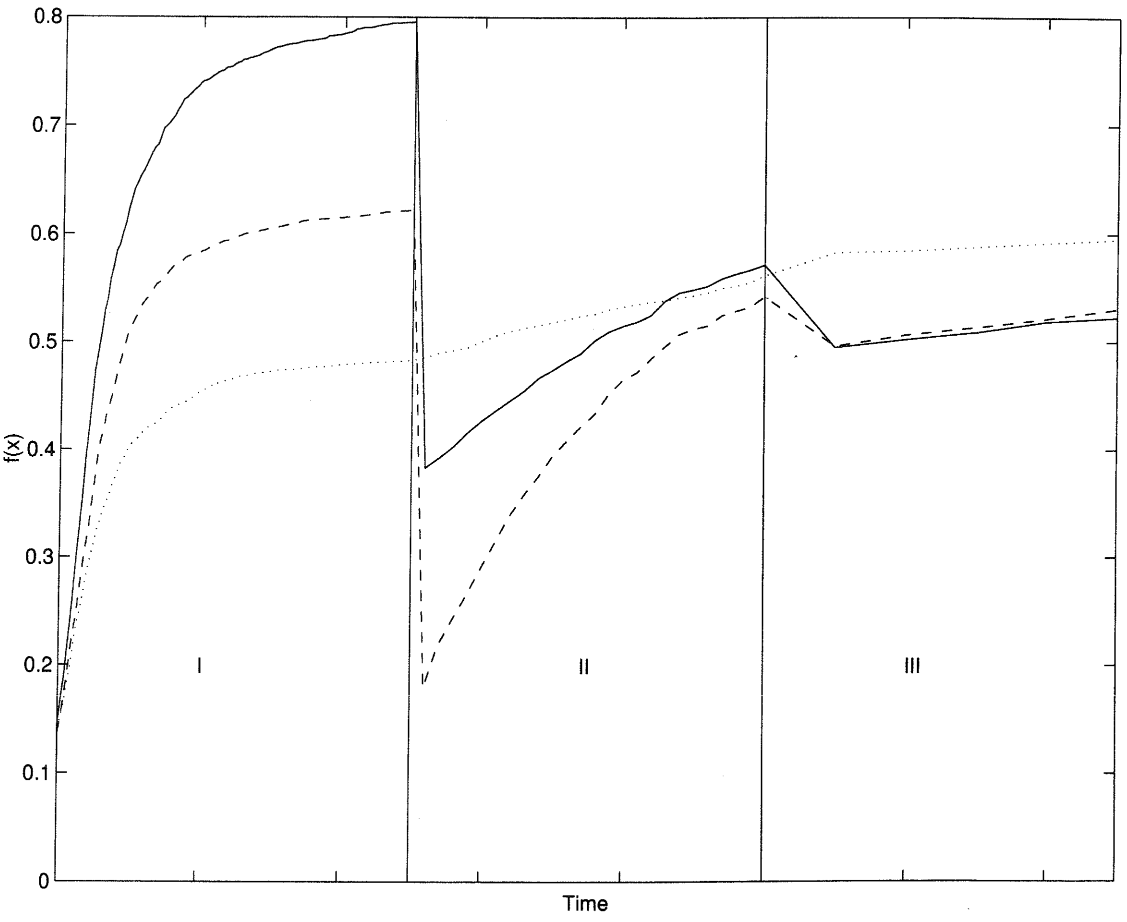


Fig. 5. Best of generation versus time using GACS for a 20D bump averaged over 30 runs. The solid line is for varying α, the dotted is for β, and the dashed is for varying αandβ.

Table 4
Gradually mixed multilevel optimization of a 2D bump

	Alpha	Beta	Alpha and Beta	Average	Average no. of trials
GACS	0.2807	0.2525	0.3251	0.2861	15,500
SA	0.2572	0.2892	0.2032	0.2498	15,500
PBIL	0.1492	0.1301	0.3413	0.2069	15,500
DHClimb	0.0018	0.1211	0.3296	0.1508	15,526
EP	0.0015	0.0668	0.3359	0.1348	15,500
ES	0.1304	0.1191	0.1332	0.1276	15,500
BClimb	0.0076	0.1259	0.2019	0.1118	15,500

In some methods the number of evaluations could not be exactly controlled. If a method converged before the specified number of evaluations for a certain level was reached, the optimization procedure was restarted on the next level with the converged values. Hence, the average number of steps is also shown in the tables below.

For the stochastic methods, the initial populations were randomly selected, but contained the point $x_i = 5$ for $i = 1,\dots,n$. For all the other methods, the optimization process was carried out starting from $x_i = 5$ for $i = 1,\dots,n$.

9.1. Sequential multilevel optimization results

For the case of the 2D bump, the results in descending order of performance are as shown in Table 2

It is clear from Table 2 that the GAs having a niching mechanism, i.e., NGA and GACS, showed superior performance. NGA's performance may be explained in part by the fact that a good estimate of the number of peaks in the search space was readily available. This can be easily obtained from the plot of the 2D bump. This is, of course, not possible in general. Note that most of the stochastic methods, on average, converged to similar peaks, regardless of the distortion sequence they have gone through. Also, the stochastic methods clearly outperformed traditional hill-climbing methods.

Table 3 shows the results for the 20-dimensional function, and it is clear that the NGA's relative performance is diminished. This is largely due to the estimate for the number of peaks here, not being accurate. In general, it is not obvious for any given problem how the number of peaks can be estimated a priori. In this test, the dynamic hill-climbing (DHClimb) algorithm's performance is comparable to that of the GACS, and it performs better in two distortion sequences. The performance of the stochastic methods contrasts even more sharply with the other methods, and even ADRANS is now markedly improved as compared to the 2D bump. The traditional hill-climbing methods clearly fall short on the more difficult 20D bump.

Fig. 5 illustrates the search behaviour using GACS, averaged over 30 runs. It is clear that the objective function values can drop significantly at the point where the switch from one level to the next occurs, but the method then recovers quite quickly afterwards.

Based on the results obtained in Tables 2 and 3, further comparisons are restricted to the top seven stochastic methods (GACS, SA, PBIL, DHClimb, EP, ES, and BClimb).

9.2. Gradually mixed results

For the mixed strategies, elitism was disabled in all methods to prevent an individual with a high objective function in the distorted representation from dominating and influencing the optimization in the latter stages, where the more accurate representations are used. During these stages, such an individual might have a reduced objective function value. This contrasts with the sequential case, where the optimization is car-

Table 5
Gradually mixed multilevel optimization of a 20D bump

	Alpha	Beta	Alpha and beta	Average	Average no. of trials
GACS	0.4788	0.5853	0.4765	0.5135	15,500
SA	0.3995	0.4509	0.4023	0.4176	15,500
PBIL	0.3367	0.5580	0.2732	0.3893	15,500
DHClimb	0.2402	0.4313	0.1731	0.2815	15,777
EP	0.2883	0.4292	0.0833	0.2669	15,500
BClimb	0.2553	0.2972	0.0808	0.2111	15,425
ES	0.1374	0.1377	0.1342	0.1364	15,500

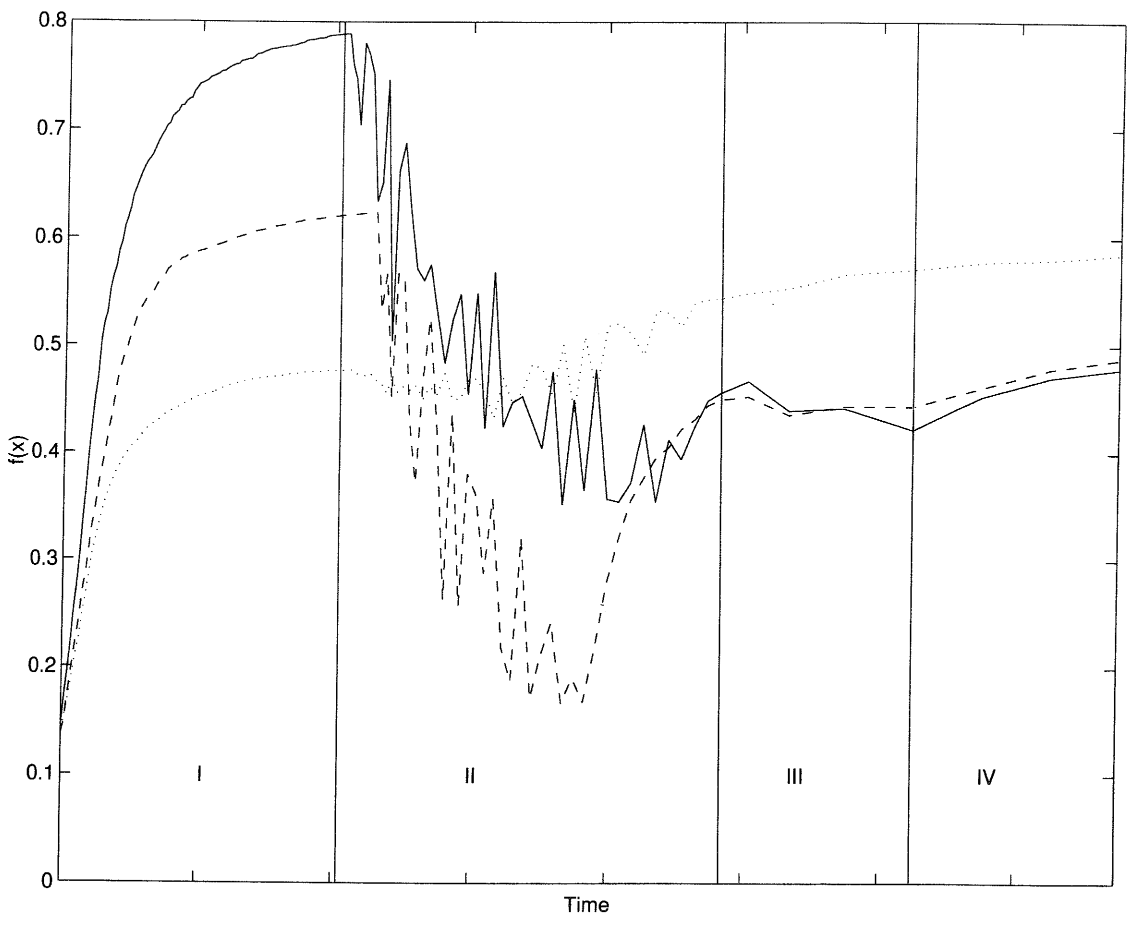


Fig. 6. Best of generation versus time using GACS for a 20D bump, averaged over 30 runs. The solid line is for varying α, the dotted is for β, and the dashed is for varying α and β.

ried over three distinct stages and hence, elitism can be used at each stage.

As in the previous section, three distortion sequences were carried out. Results were again averaged over 30 runs, see Tables 4 and 5.

Table 4 shows that, on average, GACS outperforms all the other methods on the 2D problem. Note that in the 2D case, most methods do particularly well in the distortion sequence where α and β are being varied simultaneously, but they perform poorly on the other distortion sequences.

It is also clear from the tables that many of the methods tried performed relatively better on the 20D problem than for the 2D case. This would appear, because the changes in function value between levels are then less dramatic, due to the increased complexity of the function.

Fig. 6 illustrates the search behaviour using GACS averaged over 30 runs for sequential mixing. The search is divided into four phases. In phase I only the first level is used. In phase II the first and second levels are gradually mixed; it is characterised by a 'jumpy'

Table 6
Totally mixed multilevel optimization of a 2D bump

	Alpha	Beta	Alpha and beta	Average	Average no. of trials
GACS	0.1996	0.2070	0.3026	0.2364	15,500
PBIL	0.1178	0.1261	0.3376	0.1938	15,500
DHClimb	0.0016	0.1190	0.3284	0.1496	15,535
EP	0.0017	0.1018	0.3342	0.1459	15,500
ES	0.1335	0.1444	0.1187	0.1322	15,500
BClimb	0.0659	0.1337	0.1899	0.1298	15,500
SA	0.0017	0.0802	0.2891	0.1237	15,500

Table 7
Totally mixed multilevel optimization of a 20D bump

	Alpha	Beta	Alpha and beta	Average	Averaeg no. of trials
PBIL	0.3838	0.5711	0.4822	0.4791	15,500
GACS	0.3526	0.5440	0.2953	0.3973	15,500
BClimb	0.3332	0.2854	0.3273	0.3153	15,981
EP	0.3055	0.4365	0.1079	0.2833	15,500
DHClimb	0.2738	0.3796	0.1788	0.2774	15,425
SA	0.2634	0.3771	0.1569	0.2658	15,500
ES	0.1390	0.1365	0.1376	0.1377	15,500

behaviour (this is evident in the graphs even though the results are averaged over 30 runs). During phase III, the second and third levels are mixed, and finally in phase IV, only the last level is used.

9.3. Totally mixed results

As in the previous section, elitism was again disabled, and three distortion sequences were carried out. The results were averaged over thirty runs, and are as shown in Tables 6 and 7.

Again, GACS performed better than the other methods on average for the 2D case. And again, all the other methods performed poorly on the α and β only distortion sequences.

As in the gradual mixed case, the relative performance of the other methods improves for the 20D case. Here, PBIL has overtaken GACS, although the latter is no longer performing as well as in the previous two strategies.

Fig. 7 illustrates the search behaviour using GACS averaged over 30 runs for total mixing. In phase I of

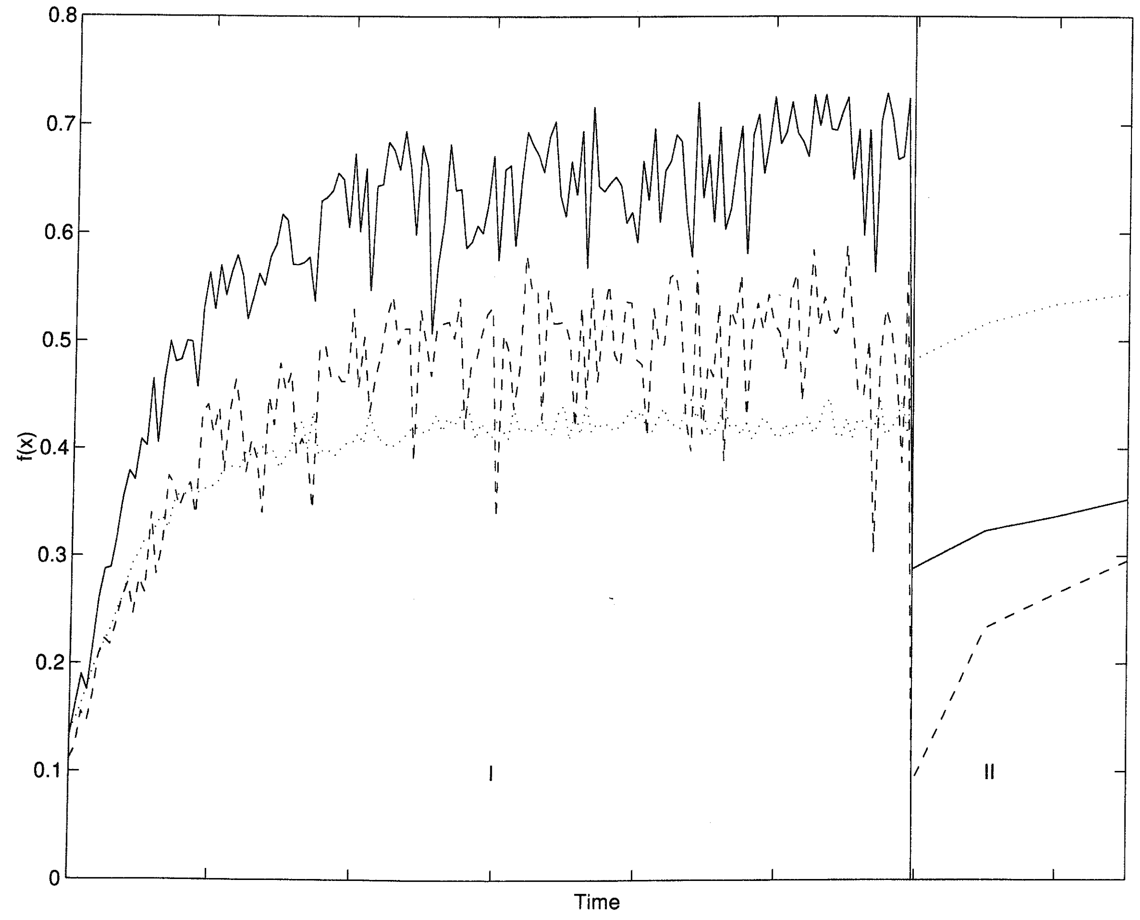


Fig. 7. Best of generation versus time using GACS for a 20D bump averaged over 30 runs. The solid line is for varying α, the dotted is for α, and the dashed is for varying α and β.

Table 8
Results for a single-level optimization using the accurate bump

	2D Average	20D Average	2D BST	20D BST
GACS	0.3118	0.4727	0.3638	0.5974
DHClimb	0.2812	0.3968	0.3576	0.5888
PBIL	0.2931	0.2243	0.3584	0.2962
EP	0.3111	0.3106	0.3630	0.3969
SA	0.3007	0.3935	0.3638	0.4821
BClimb	0.2623	0.4508	0.3626	0.5908
ES	0.2734	0.1833	0.3576	0.2110

the search, all three levels are mixed. The many fluctuations seen here are due to the mixing (they are too random to be smoothed out, even after averaging over 30 runs). There is then a sharp dip after the optimization moves into phase II, where only the most accurate representation is used and earlier, misleading results are discarded.

10. Discussion

To gain a basis of comparison for these various results, it is useful to consider the case where the entire computational effort is dedicated to using the most accurate level (i.e., 1500 such evaluations, assuming the ratios between the number of evaluations given in Table 1). Using this approach, the performance of the stochastic optimizers is as in Table 8. Shown are the averaged (AVG) and best ever obtained results (BST) over 30 runs for each optimizer.

The comparative improvement of each method using any of the three proposed strategies may then be calculated, see Table 9. Here the results are normalised by dividing the average performance of each method in Tables 2–7 by the values in Table 8. The results may also be normalised by dividing by the best ever results obtained for the accurate function evaluation (0.365^1 and 0.8035^2, respectively, for the 2D and 20D cases), see Table 10.

It clear from the tables that:

- Sequential Mixing is the best strategy when averaged across all methods.
- None of the three proposed strategies provided any improvement for the 2D case.
- For the 20D case, only GACS, PBIL, and EP were in general improved by adopting the mixing strategies proposed.
- The overall best approach for the 20D case was to use GACS with the sequential strategy. This gives an average final objective function on the three tests (α only, β only, α plus β) of 0.5495, see Table 3. However, this final value is only 16% better than a straight-forward use of GACS on the accurate function for 1500 steps.
- Although PBIL and EP showed improvements in the mixed methods (with a factor of 2 for PBIL and 1.0645 for EP), their objective function values were generally lower than GACS (the exception being PBIL on the 20D totally mixed case); also, their performance was rather erratic between runs.

These observations lead to the conclusion that to work well in these mixed-method environments, an optimizer must be specifically designed to cope with this kind of domain. In the case of GA, this may perhaps point to the need for a diploid scheme as a means for coping with this kind of environment, or perhaps a modified injection island scheme.

11. Conclusion and future work

A distorted bump function has been presented here as being representative of a multimodal, objective function that may be used as a tool for understanding how different optimizers behave in multilevel environments. Multilevel optimization is characterized by the cost of evaluating the function being directly related to its accuracy. Hence, precise evaluations must be used sparingly for an efficient search. Different multilevel

Table 9
Comparative improvements using the three mixing strategies

	Sequential mixing		Gradual mixing		Total mixing		Average	
	2D	20D	2D	20 D	2 D	20 D	2 D	20 D
GACS	0.9182	1.1624	0.9176	1.0864	0.7582	0.8405	0.8647	1.0298
DHClim b	0.9930	1.3286	0.5364	0.7095	0.5322	0.6991	0.6872	0.9124
PBIL	0.8116	2.1450	0.7058	1.7357	0.6613	2.1358	0.7262	2.0055
EP	0.8785	1.4221	0.4331	0.8594	0.4690	0.9121	0.5935	1.0645
SA	0.9417	1.0136	0.8308	1.0611	0.4113	0.6755	0.7279	0.9167
Bclimb	0.9831	0.8580	0.4262	0.4682	0.4950	0.6994	0.6348	0.6752
ES	0.7791	0.9541	0.4666	0.7443	0.4836	0.7513	0.5764	0.8165
Average	0.9007	1.2691	0.6167	0.9521	0.5444	0.9591	0.6872	1.0601

Table 10
Absolute performance using the three mixing strategies

	Sequential mixing		Gradual mixing		Total mixing		Average	
	2D	20D	2D	20D	2D	20D	2D	20D
GACS	0.7844	0.6838	0.7838	0.6391	0.6477	0.4944	0.7386	0.6058
DHClimb	0.7650	0.6561	0.4132	0.3504	0.4100	0.3452	0.5294	0.4505
PBIL	0.6517	0.5987	0.5668	0.4845	0.5311	0.5962	0.5832	0.5598
EP	0.7487	0.5497	0.3692	0.3322	0.3997	0.3526	0.5059	0.4115
SA	0.7758	0.4963	0.6845	0.5196	0.3389	0.3308	0.5997	0.4489
Bclimb	0.7065	0.4813	0.3063	0.2627	0.3557	0.3924	0.4562	0.3788
ES	0.5836	0.2176	0.3495	0.1698	0.3622	0.1714	0.4318	0.1863
Average	0.7165	0.5262	0.4962	0.3940	0.4350	0.3833	0.5492	0.4345

strategies and optimization methods have been used to study this process. The use of a GA based on clustering and sharing (GACS), which distributes the population over many peaks in a changing fitness landscape, has been shown to give improved results using most of the different strategies. Though a number of methods came close to the GACS on some tests, none was as robust under all the different distortion sequences and different multilevel integration strategies. Overall, a sequential strategy using cheap but inaccurate solutions first, followed by a lesser number of intermediate solutions before finally using a few calls to the fully accurate but most expensive function, proved to be the most effective approach. It was, however, only 16% more efficient than a simple use of only the most accurate function over an equivalent, but limited, number of trials.

Future work will be directed towards more efficient mixing strategies, as well as more specialized optimization techniques, specifically designed to be able to handle this kind of problem.

Acknowledgements

This work was supported under EPSRC grant no. GR/L04733

References

Aizawa, A.N., Wah, B.W., 1993. Dynamic control of genetic algorithms in a noisy environment. In: Forrest, S. (Ed.), Proceedings of the 5th International Conference on Genetic Algorithms. Morgan Kaufmann, Palo Alto, pp. 48–56.

Angeline, P.J., 1997. Tracking extrema in dynamic environments. In: Angeline, P.J., Reynolds, R.J., McDonnell, J.R., Eberhart, R. (Eds.), Evolutionary Programming VI: Proceedings of the Sixth Annual Conference on Evolutionary Programming. Springer-Verlag, Berlin, pp. 335–345.

Bäck, T., 1998. On the behaviour of evolutionary algorithms in dynamic enviroments. In: Proceedings of the Fifth IEEE Conference on Evolutionary Computation. IEEE, Piscataway, NJ, 446–451.

Bäck, T., Hoffmeister, F., Schwefel, H.P., 1991. A survey of evolution strategies. In: Belew, R.K., Booker, L.B. (Eds.), Proceedings of The Fourth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, CA, pp. 174–196.

Baluja, S., 1994. Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Report CMU-CS-94-163. Carnegie Mellon University.

Box, M.J., 1965. A new method for the constrained optimization and a comparison with other methods. Computer Journal 8, 42–52.

Davis, L., 1991. Bit-Climbing, representational bias, and the test suite design. In: Belew, R.K., Booker, L.B. (Eds.), Proceedings of The Fourth International Conference on Genetic Algorithms. Morgan Kaufman, San Mateo, CA, pp. 18–23.

Deb, K., 1989. Genetic algoritms in multimodal function optimization. M.S. Thesis, College of Engineering, University of Alabama, Tuscaloosa AL.

Deb, K., Goldberg, D.E., 1989. An Investigation of niche and species formation in genetic function optimization. In: Forest, S. (Ed.), Proceedings of The Fifth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, CA, pp. 42–50.

Dunham, B., Fridshal, D., Fridshal, R., North, J.H., 1963. Design by natural selection. Synthese 15, 254–259.

Eby, D., Averill, R.C., Punch, W.F., Goodman, E.D., 1998. Evaluation of injection island GA performance on flywheel design optimization. In: Parmee, I. (Ed.), Proceedings of Third Conference on Adaptive Computing in Design and Manufacturing. Springer Verlag, London, pp. 121–136.

Ellman, T., Keane, J., Schwabacher, M., 1993. Intelligent model selection for hillclimbing search in computer-aided design. In: Proceedings of the National Conference on Artificial Intelligence, 594–599.

Elster, C., Neumaier, A., 1993. A trust region method for the optimization of noisy functions. Computing 58, 31–46.

Elster, C., Neumaier, A., 1995. A grid algorithm for bound constrained optimization of noisy functions. IMA J. Numer Anal 15, 585–608.

Fogel, D.B., 1993. Applying evolutionary programming to selected traveling salesman problems. Cybernetics and Systems 24, 27–36.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA.

Grefenstette, J.J., Fitzpatrick, J.M., 1985. Genetic search with ap-

[2] This is from the result reported in (Michalewicz, 1996 p, 156).

[2] This is from the result reported in (Michalewicz, 1996 p, 156).

proximate function evaluations. In: Grefenstette, J.J. (Ed.), Proceeding of the First International Conference on Genetic Algorithms and Their Application. Lawarence Erlbaun, Hillsdale, NJ, pp. 112–120.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor.

Keane, A.J., 1994. Experiences with optimizers in structural design. In: Parmee, I. (Ed.), Proceedings of the Conference on Adaptive Computing in Engineering Design and Control (ACEDC'94). PEDC, University of Plymouth, UK, Plymouth, pp. 14–27.

Keane, A.J., 1995a. A brief comparison of some evolutionary optimization Methods. In: Proceedings of the Conference on Applied Decision Technologies (Modern Heuristic Search Methods), 125–137.

Keane, A.J., 1995b. Genetic algorithm optimization of multi-peak problems: studies in convergence and robustness. Artificial Intelligence in Engineering 9 (2), 75–83.

Keane, A.J., 1995c. The OPTIONS design exploration system: reference manual and user guide.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680.

KrishnaKumar, K., Swaminathan, R., Montgomery, L., 1994. Multiple optimal solutions for structural control using genetic algorithms with niching. Journal of Guidance and Control 17 (6), 1374–1377.

Lund, H.H., 1994. Adaptive approaches towards better GA performance in dynamic fitness Landscapes. Report DAIMI PB-487. Aarhus University.

Michalewicz, Z., 1996. Genetic Algorithms + Data Structures = Evolution Programs, 3rd ed. Springer, New York.

Miller, B.L. and Goldberg, D.E., 1995a. Genetic algorithms, selection schemes, and the varying effects of noise. Report 95009. University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

Miller, B.L. and Goldberg, D.E., 1995b. Genetic algorithms, tournament selection, and the varying effects of noise. Report 95007,University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

Mitchel, M., Forrest, S., Holland, J.H., 1991. The royal road for genetic algorithms: fitness landscapes and GA performance. In: Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life. MIT Press, Cambridge, MA.

NAg E04UCF, 1990. Nag mark 14 reference manual, Numerical Algorithms Group Ltds, Oxford.

Nelder, J.A., Meade, R., 1965. A simplex method for function minimization. Computer Journal 7, 308–313.

Nissen, V., Propach, J., 1998. On the robustness of population-based versus point-based optimization in the presence of noise. IEEE Transactions on Evolutionary Computation 2, 107.

Parmee, I.C., Vekeria, H.D., 1997. Co-operative evolutionary strategies for single component design. In: Bäck, T. (Ed.), Proceedings of The Seventh International Conference on Genetic Algorithms. Morgan Kaufmann, San Fransico, CA, pp. 529–536.

Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., 1986. Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, Cambridge.

Rosenbrock, H.H., 1960. An automatic method for finding the greatest or least value function. Computer Journal 3, 175–184.

Schwefel, H.P., 1995. Evolution and Optimum Seeking. Wiley, New York.

Siddall, J.N., 1982. Optimal Engineering Design: Principles and Applications. Marcel Dekker, New York.

Sobieszczanski-Sobieski, J., Haftka, R.T., 1996. Multidisciplinary aerospace design optimization: survey of recent developments. In: 34th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA Paper No. 96-0711,.

Van Keulen, F., Toropov, V.V., Polynkine, A.A., 1995. Shape optimization strategies using the multi-point approximation method and adaptive mesh refinement. In: Proceedings of the First World Congress on Structural and Multidisciplinary Optimization. Pergamon Press, Oxford, 67–74.

Wang, L., Grandhi, v., 1995. Recent multi-point approximations in structural optimization. In: Proceedings of the First World Congress on Structural and Multidisciplinary Optimization. Pergamon Press, Oxford, 75–82.

Yin, X., Germay, N., 1993. A fast genetic algorithm with sharing scheme using cluster methods in multimodal funtion optimization. In: Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms. Springer-Verlag, Innsbruck, 450–457.

Yuret, D., de la Maza, M., 1993. Dynamic hill climbing: overcoming the limitations of optimization techniques. In: Proceedings of the 2nd Turkish Symposium of AI and ANN, 254–260.

M.A. El-Beltagy received his BSc. in Mechanical Engineering from the American University in Cairo in 1994. He received an M.Sc. in Mechatronics from Lancaster University in 1996. He is currently doing his PhD. at the University of Southampton on Multidisciplinary Optimisation. His work is supported by BAe plc and the EPSRC.

A.J. Keane is currently director of Southampton University's Computational Engineering and Design Centre. He has worked in the fields of structural dynamics, computational engineering and design exploration. To date he has published over 50 papers and edited two books in these fields. His work has been supported by the EPSRC, the DTI, the DERA, BAe plc, Glaxo Wellcome, Rolls Royce, and Cable and Wireless plc, amongst others.