To appear in "Evolutionary Computing: AISB Workshop", Ed: T.C. Fogarty, Springer-Verlag LNCS ?, 1996.

Inoculation to Initialise Evolutionary Search

Patrick D. Surry, Nicholas J. Radcliffe
{pds,njr}@quadstone.co.uk

Quadstone Ltd, 16 Chester Street, Edinburgh, EH3 7RA, UK
Department of Mathematics, University of Edinburgh, King's Buildings, EH9 3JZ, UK

Abstract. An important factor in the successful application of evolutionary techniques to real-world problems is the incorporation of domain knowledge. One form such knowledge often takes is the possession of one or more high-quality solutions. Non-random initialisation, or inoculation, of the population in an evolutionary algorithm provides a way to incorporate such knowledge. A body of folklore about the methods and results of such initialisation techniques exists, but is largely unwritten and unquantified. This paper discusses the need for hybridisation, through whatever means, and concentrates on the potential offered by seeding the initial population with extant good solutions. Such ideas also have implications for algorithmic restarts after convergence. Experiments conducted using a number of real industrial and commercial problems confirm some of the accepted folklore, and highlight several interesting new results. In particular, it is found that both average solution quality and run-times improve when reasonable inoculation strategies are used, but that the quality of the best solution found over a number of runs often deteriorates as the initial populations become less random.

1 Introduction

Most techniques for evolutionary search are regarded as weak methods, in that they fail to incorporate domain knowledge explicitly. However, it is recognised (albeit not so widely as might be desirable) that success or failure may depend crucially on the quality of the information implicitly captured by the representation and operators which are employed (e.g. Radcliffe, 1992). The importance of such prior knowledge of the target function is already widely accepted in the global optimisation community at large (Zhigljavsky, 1991). Nevertheless, much of the research in evolutionary computation takes a problem-independent view. In fact, some authors go so far as to speculate about 'black-box' optimisation (Goldberg, 1989; Kargupta, 1995), despite the fact that any hopes for strong results here are provably fruitless (Radcliffe, 1992; Wolpert & Macready, 1995; Radcliffe & Surry, 1995). These authors, amongst others, have argued strongly that useful results (i.e. which distinguish a given method from enumerative search) are only possible when parameterised by characteristics of a problem domain of interest. Ironically, as well as work on black-box optimisation, there have been numerous studies of individual problem *instances*, but disappointingly few on well-specified problem *classes*.

One way in which genetic search often can be strengthened within a given problem domain is through *hybridisation* with other search methods. Davis (1991) is a particular advocate of this, as are many of the practitioners actively using evolutionary methods in real-world applications. Due to their inherent generality and simplicity, evolutionary algorithms provide an extremely convenient context in which to embed other techniques, and numerous approaches have been considered.

The addition of specialised move operators incorporating heuristics or exploiting domain knowledge in other ways is often effective (Michalewicz, 1992; Davis & Orvosh, 1993). If a true local-optimisation algorithm exists, these ideas can be extended by applying full local optimisation to each child solution produced before evaluation, yielding a *memetic algorithm* (Moscato & Norman, 1992; Radcliffe & Surry, 1994b). Embedding a search technique in the genotype-phenotype mapping is another possibility (Belew *et al.*, 1990; Valenzuela & Jones, 1994). In this case, the evolutionary algorithm may be regarded as searching not directly over the space of solutions, but rather over the space of inputs to the search method with which it is being hybridised.

Although such hybrid approaches allow evolutionary techniques to benefit from existing domain knowledge or heuristics in a dynamic way as the search progresses, additional information may be available in forms which do not readily fit within this framework. In particular, it is often the case that one or more good candidate solutions are known (i.e. solutions which are statistically unlikely to be found in reasonable times by random search). These may be the result of applying a local optimisation algorithm (perhaps one too expensive to embed in an evolutionary search), specialist knowledge or experience (which may be difficult to capture algorithmically), current best-practice, or may even be the result of a previous run. Alternatively, some knowledge of the fitness function (such as partial separability) may be available. A reasonable question is thus: in what way (if any) can knowledge be usefully exploited to find better solutions in an evolutionary context?

In section 2 we review previous work on the problem of initialisation, highlighting the different approaches taken and the further lines of attack that they suggest. In section 3 we use these ideas to explore the theoretical implications of the question posed above, leading to discussion of several practical initialisation strategies. This is followed by an examination of the issues of population diversity and convergence in section 4. Several real industrial and commercial applications are used to test the proposed methods in section 5 and the results are discussed in section 6, which also contains proposals for further work.

2 Previous Approaches to Initialisation

Little emphasis is given in the evolutionary computation literature to the problem of initialisation; the near-universal approach is to choose an initial population uniformly at random from the search space (Holland, 1975; Goldberg, 1989). Particularly in real-world problem domains where hybrid approaches are often adopted for the search itself, it seems likely that more effective strategies must exist. It is in fact the authors' experience that there is a widespread body of unwritten and unquantified folklore about hybridisation via non-random initialisation. One often hears that non-random initialisation "works" and "has been used successfully for application X" if care is taken to "avoid premature convergence". Specifics, however, are often difficult to pin down.

Grefenstette (1987) captures the main issues involved in non-random initialisation when he states that:

> . . . the initial population . . . might be chosen heuristically rather than randomly, with the goal of introducing some helpful building blocks into the gene pool. This should be done carefully since [a genetic algorithm] may quickly converge to a local optimum if the initial population contains a few structures that are far superior to the rest of the population.

He carried out empirical studies of the travelling sales-rep problem using three different initialisation strategies. In the first method he attempted to maximise allelic entropy in the initial population (super-randomness, if you will), while in the other two the initial population contained some fraction of tours generated by the commonly-used nearest-neighbour heuristic and a simple variant thereof. He reported no significant benefit to using heuristic initialisation.

Related work on the travelling sales-rep problem includes results from simulated annealing, such as that of Sourlas (1986). He used conventional simulated annealing to find the (small) set of edges which appeared regularly in tours at some fixed low temperature. He was then able to search for optima much more efficiently using an annealing algorithm which only considered moves involving these *acceptable* edges. He also reported empirical studies which indicated that only short edges appear with high probability in this acceptable set (i.e. each city is very likely to be visited immediately before and after one of its nearest neighbours). This suggests the possibility of initialising based on some non-uniform probability distribution over the set of all edges (which can be viewed as alleles in this problem; Radcliffe & Surry, 1994a). This could obviously be generalised to any domain in which a useful prior distribution over alleles is available, perhaps even developed from previous runs.

The phenomenon of premature convergence (usually taken to mean the situation in which a population has become nearly homogeneous without including a global optimum) is important in evolutionary search, and has been widely studied. Numerous methods for avoiding it, or for restarting search following its occurrence, have been proposed. If one views a single good solution (from whatever source) as the focus of a “converged population”, the relevance of such restart methods to the initialisation problem is apparent. Most work in this area has considered only binary representations (sometimes encoding real-valued parameters) whereas in this paper we consider the non-binary representations typically employed in real-world applications (Davis, 1991). This introduces important technical differences in the character of mass-mutation.

Shaefer (1987) introduced many novel ideas of relevance to initialisation in his ‘ARGOT’ strategy, which utilised a dynamically adapting representation. Both the resolution and range with which real parameters were represented by binary strings were adapted over time, based on population measures of gene-wise convergence and diversity, helping to avoid premature convergence. Shaefer & Smith (1990) extended ARGOT to handle combinatorial problems such as the travelling sales-rep. In a similar approach, Schraudolph & Belew (1990) considered a dynamic parameter-encoding scheme in which the interval represented by a given gray-coded binary gene shrinks over time, triggered based on measures of population convergence. Both pairs of researchers reported good results on a test suite of problems.

In his CHC genetic algorithm, Eshelman (1990) employed *mass mutation* to restart the search when the population had converged. (In normal operation CHC used no mutation.) During a restart, copies of the best individual found to date were mutated at a high rate (e.g. 35%) to produce the new population. Typically the best individual was also added unchanged. Whitley *et al.* (1991) instead advocated the *delta-coding* technique for restarts. In this method, chromosomes encoded relative distances from previously discovered good solutions, rather than absolute parameter values. Their algorithm was repeatedly run until convergence and then restarted with the new chromosomes representing displacements from the best previous solution.

It has been reported that even in the absence of known good solutions it can be computationally efficient to allocate more than the minimum resources to initialisation before the search itself takes place. For example, Bramlette (1991) used what he called *extended random initialisation* in which each population member was selected as the best of n randomly chosen individuals. He considered several standard test functions and used a meta-GA to evolve this tournament size (among other parameters), and reported good results with n up to 20, for which about 14% of function evaluations were used in initialisation. The *messy genetic algorithm* of Goldberg *et al.* (1989) used an analogous *primordial phase* to find 'useful' building-blocks.

In real-world applications, the importance of non-random initialisation has often been cited. Fogarty (1989) used a genetic algorithm to optimise valve settings for an industrial furnace application. He experimented with time-varying mutation rates, and found that an effective initialisation strategy involved starting with a completely homogeneous population (cloned from a conservative solution) and using an initially high but exponentially decreasing mutation rate in the ensuing evolutionary process. He also showed that such a mutation schedule was not particularly effective when starting with a randomised population. There are clear links with the mass-mutation approach but the two methods are not equivalent.

De la Maza (1989) studied the problem of generating production rules to make inferences about a database (here concerning horse-racing). Finding that run-times were excessive when starting from a random population, he instead began with a population consisting of the best single-variable rules produced by a simple heuristic.

Powell *et al.* (1989) combined a genetic algorithm with an expert system for engineering design optimisation (e.g. turbine design). Here, the evaluation function was a complex simulation code with many degrees of freedom. To avoid the perceived inefficiency of evaluating many random solutions, they seeded more than half of the initial population with a combination of previous good solutions and prior design steps made by the expert system. They reported that this reduced by a factor of five the number of runs required to yield a given performance improvement.

Ramsey & Grefenstette (1993) evolved condition-action rules in the face of a changing environment in their SAMUEL system. They employed a restart strategy they termed *case-based initialization*. At the start of each new epoch (environment change) a large part of the population was replaced with a combination of best strategies from similar epochs, robust general strategies, and some random strategies. They claimed that this mix helped to preserve diversity and to avoid premature convergence.

In his work on Tierra, Ray (1994) considered the evolution of computer programs which illustrated many parallels with biological systems, and seeded the initial "primordial soup" with either hand-crafted or previously evolved programs. He introduced the term *inoculation* for the process of non-random initialisation. As defined by the Collins English dictionary,

> **inoculate** *vb.* **1.** to introduce (the causative agent of a disease) into the body in order to induce immunity **2.** to introduce (microorganisms, esp. bacteria) into (a culture medium) **3.** to cause to be influenced or imbued, as with ideas

he used it in its second form. However, it is interesting to note that the third form is particularly apt for the process by which we try to exploit the unknown good characteristics captured by some known high-quality solution(s).

3 Initialisation and Inoculation Strategies

We now return to the question of how to exploit domain knowledge through initialisation. We start by establishing the conditions under which this may be feasible, and go on to suggest various mechanisms by which it might then be accomplished.

3.1 Theoretical Considerations

When presented with one or more good solutions to a search problem, but with no additional information, we can argue from fundamental limitations on search that they are useless in the quest for better solutions—that is, the search will perform equally well without them. The so-called *no free lunch theorem* (Wolpert & Macready, 1995) and related results (Radcliffe, 1992; Radcliffe & Surry, 1995) captures the straightforward idea that in the absence of prior information (i.e. domain knowledge) no search method can outperform enumeration. The essential intuition is that even after sampling an unknown function at some subset of its domain, there is no principled way in which to predict its value at some as-yet unsampled point without making assumptions about the function. Clearly, additional knowledge of the function's structure is required before any particular set of assumptions could be justified. When we consider a set of good solutions and their corresponding function values in the same light, as if they formed some "past search history", we see that they must therefore be useless to further search unless we can make some assumptions about the problem structure.

We must therefore posit some additional (domain-specific) knowledge in order to admit the possibility that known good solutions might be useful (and indeed to have any purpose in continuing this discussion). This is not unreasonable as any real problem is likely to contain a great deal of structure. Of course, this does not mean to say that said structure is necessarily easy to exploit!

As discussed previously, evolutionary methods typically make use of domain knowledge only implicitly through the representation and operators that they employ. It thus seems that this is a reasonable mechanism by which to inject the required additional information. That is, we will assume that a context of representation and operators (in some senses duals of one another; Battle & Vose, 1991) is given, and that they are in some loose sense "well-suited" to the problem at hand.

We may then refocus our attention on the problem of how to make use of known good solutions, given also a suitable representation and operators which will be employed in the search. We claim that such high-quality individuals will be useful in finding better solutions only if they share properties with respect to the operators and representation at hand. In the language of evolutionary search, this may mean that the hamming distance between the corresponding genomes is relatively small, or that the average fitness of some schemata (or their generalisation, formae; Radcliffe, 1991) to which they belong is relatively high. However, because the mechanisms by which evolutionary search "works" are as yet poorly understood, these are by no means the only possibilities.

It should also be noted that the oft-stated goal of finding a global optimum of a given function is poorly conceived unless such an optimum shares characteristics with other high-quality solutions that permit it to be found after visiting only a small fraction of the search space. It is not *in general* the case that a representation which is well-suited to a search problem in the sense discussed above will have the same qualities for the particular points where global optima happen to be located.

The arguments above suggest that any approach to the problem of initialisation should take place within the context defined by the particular representation and operators to be used in the ensuing search. Further, it leads one to believe that search using a given representation and operators will likely be more effective if non-random initialisation (using given high-quality solutions) is performed using the *same* combination of representation and operators. (This could be explicitly tested in the context of any problem for which we can construct multiple natural representations.) We proceed to explore some of the possibilities for such initialisation strategies.

3.2 Practical Inoculation Strategies

When no heuristic knowledge is available, and no assumptions are made concerning representation or operator quality, we proceed by simply selecting the population uniformly at random from the search space. There is also the option of biasing the sampling to increase the uniformity of coverage of the search space (Holland, 1975; Grefenstette, 1987).

The simplest strategy for incorporating domain knowledge in the initialisation process is to *inoculate* a random population by adding a good solution and then to let the algorithm take its course. The heuristic will be manipulated by the hypothesised good operators in the well-suited representation, and if it does capture useful information, may help the search to proceed more efficiently. In practice, details of the particular algorithm used will also have dramatic effects on efficacy as discussed in section 4.

Given any mutation operator, *mass mutation* can be used to inject variants of a good solution into an initial population. In this procedure, a solution is repeatedly cloned and mutated at a high rate to produce each member of the population. This population may also be inoculated with an unmodified copy of the original solution. As the rate of mutation is increased, the resulting population becomes less clustered about the original solution and more like a random one. Note however that with any mutation operator that does not chose uniformly between alleles (such as creep mutation) even mass mutation with a rate of 100% does not result in a truly random population.

One could also envisage using a recombination operator in the initialisation process but this will be pursued in a future paper.

A further alternative, not requiring a particular good-quality solution, presents itself when allele values are known to capture fitness information in some way (for instance, there may be some degree of linear separability in the fitness function). In this case it is possible to initialise the population with a non-uniform distribution over allele frequencies. This is explored further in the discussion of the travelling sales-rep problem in section 5.4.

It is clear that any method for incorporating previously known good-quality solutions into an evolutionary algorithm can also be used for *restarting* the algorithm once it has converged in some sense (see section 4). In such a case, the best solution or solutions from the final population can be used to seed a new initial population. However, it is important to distinguish between the two cases. For while we have to posit a good representation and operators in order for an externally provided solution to be potentially useful, in the case of a restart we know that the solutions we are initialising with have been discovered exactly because we used the particular representation and operators that we did. This may appear to be a subtle distinction, but seems to be borne out in experiments—in the case of the TSP (section 5.4) we find that we can effectively inoculate with a heuristic tour but that an efficient restart strategy is elusive. Further research is clearly required here.

4 Premature Convergence and Population Diversity

A number of practical algorithmic details must be considered when utilising any initialisation procedure. Most important perhaps is the issue of avoiding premature convergence and preserving population diversity, as it is widely believed that inoculation strategies exacerbate these problems. We discuss various ways in which convergence (or conversely, diversity) can be measured, and then mention some methods by which it can be combatted.

Population diversity can be defined in a variety of ways on each of a number of levels. We can consider measuring genotypic diversity, phenotypic diversity, or the diversity of objective-function (fitness) values.

Fitness diversity measures are typically simplest, as they involve only the calculation of parameter estimates to characterise the spread in the distribution of a single real variable (such as variance or higher moments).

Phenotypic diversity measures are normally application dependent, but involve measuring the heterogeneity in expressed characteristics after morphogenesis.

Genotypic measures, at least in the case of genetic or pseudo-genetic representations (Radcliffe & Surry, 1994a), involve measuring the spread of a set of points in a multi-dimensional vector space (in which the axes correspond to genes and allowable co-ordinates correspond to alleles). Any pair-wise distance metric, such as hamming distance (for cardinal genes) or euclidean distance (for ordinal genes) can be used to collapse the multi-dimensional space into a single-variable distribution which can be analysed using conventional statistical techniques. Alternatively, entropic methods seek to calculate the first-order information content of a typical population member (e.g. Grefenstette, 1987). They are applicable only in the case when alleles are drawn from a finite set. An further approach would be to employ cluster analysis techniques, although the authors are unaware of any work in this area.

Combined definitions are also possible—for instance Whitley *et al.* (1991) measure diversity by calculating the hamming distance between the pair of genomes (bit strings in this case) with the best and worst fitnesses.

Numerous techniques have also been developed for countering convergence and thus encouraging diversity. These include exploitation of population structure and speciation (surveyed in Radcliffe & Surry, 1994c), co-evolutionary models (e.g. Hillis, 1991; Husbands & Mill, 1991), adaptive mutation (Whitley & Hanson, 1989), incest-prevention (Eshelman, 1990), crowding (DeJong, 1975) and sharing (Goldberg, 1989), as well as others. Due to the diverse nature of the problems studied here (section 5) we used only simple measures based on the selection and replacement regime. Our algorithms enforced uniqueness within the population and used tournament selection to induce a relatively low selection pressure. These measures help to avoid any difficulties created by the existence of a *super-individual* (a solution substantially better than the rest of the population), which is likely to be the case after inoculation. We also employed elitism in order to prevent the stochastic loss of the best solution (particularly important when only a single inoculant is initially present).

As noted below, we actually find that no unusual loss of diversity is associated with runs using non-random initialisation, but that we do nevertheless often observe convergence to lower-quality solutions than with random initialisation.

5 Experimental Results

Quadstone Ltd, a decision support company, has worked on a variety of industrial and commercial optimisation projects. Several of these applications provide the basis for experiments on various of the previously discussed initialisation strategies. This brings the twin advantages of having non-trivial, real-world applications rather than artificially constructed problems, and also of building on the extensive prior effort on algorithmic tuning and sensitivity studies.

Four problems are presented, two involving high-dimensional real-valued search domains, and two of a combinatorial nature. They include both constrained and unconstrained problems. In each case, one or more high-quality solutions is available from external sources. Several initialisation strategies are investigated, including inoculation of a random population with a single instance of a heuristic, mass mutation from clones of a heuristic (but with the heuristic itself not present in the initial population), and random start. Some problem-specific methods are also used.

5.1 Gas-network pipe sizing

Quadstone has worked with British Gas plc on a pipe-sizing problem for a gas-network (Boyd *et al.*, 1994). In this application, a network topology is specified along with a pattern of nodal gas demands and supplies. The requirement is to select diameters for each

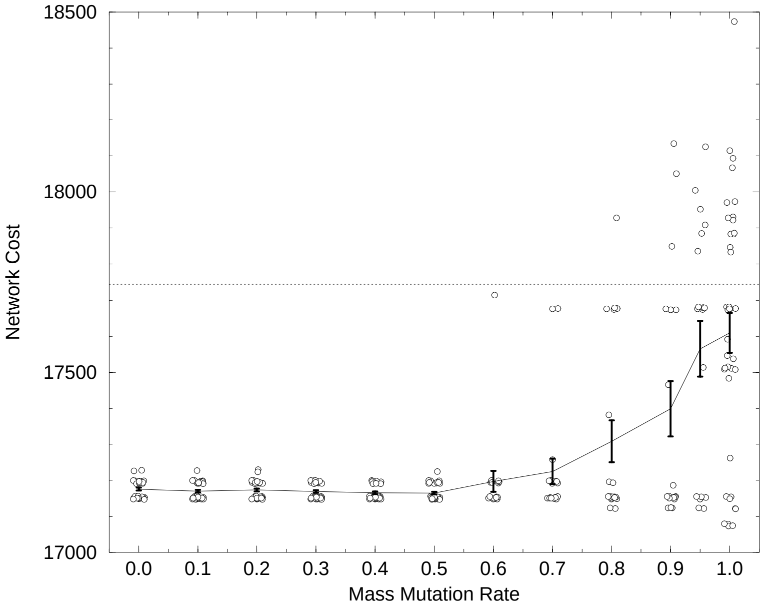

Fig. 1. Each circle in the figure represents the best result found in a single run from an initial population formed by mass mutation of a known good solution (produced by a proprietary heuristic; cost shown as dotted line). A rate of 1.0 corresponds to a completely random population, and a rate of 0.0 is used to indicate a random population containing a single unchanged copy of the inoculant. A small amount of random noise has been added to both the coordinate values in order that multiple identical points can be distinguished. The curve indicates the mean best performance with standard error bars. Although mean performance improves as the population starts more tightly clustered around the heuristic solution (lower mass-mutation rates), the spread also decreases so that the best solution found over a number of runs actually deteriorates. Indeed, the very best networks are found *only* when starting with a random population. See also table 1 and figure 2.

pipe segment to achieve lowest cost (where smaller pipes are cheaper). The problem difficulty arises due to two constraints—that nodal pressures must exceed a minimum design value, and that each pipe must have at least one upstream pipe of the same or greater diameter. Even a relatively small network (25 pipes) with a small range of allowable diameters (6 for each pipe) leads to a respectably-sized search space ($6^{25} \simeq 3 \times 10^{19}$) in which valid networks are sparse (random sampling indicates that only of the order of 1 in 10^7 networks satisfies both constraints). A network designed by British Gas using a proprietary heuristic had previously been installed, and this provided the high-quality solution used to seed the populations in our experiments.

A previously tuned genetic algorithm based on the COMOGA method (Surry *et al.*,

Network Cost	Hamming distance from:	
	Heuristic	Best Known
17743.8	(0)	12
17677.3	9	19
17227.4	4	12
17195.3	**5**	**13**
17151.5	**7**	**15**
17125.1	15	5
17075.3	12	(0)

Table 1. The hamming distances from both the heuristic solution (top) used to seed the populations and from the best known solution (bottom) are tabulated. (Each solution comprises 25 integer-valued genes, each with six alleles.) The two entries in bold represent the strong attractors shown in figure 1, which are closer to the heuristic than the two better solutions. However, it is likely that the topology of the search-space induced by the constraints also plays an important role.

1995) was used, with populations of 100 networks and terminating after at most 20,000 evaluations or 5,000 with no improvement. Due to the ordinal nature of the integer-valued genes, both creep and random mutation are used, but mass mutation was carried out using random allele replacement. Each of the 600 runs carried out during the experiment converged to a viable network notwithstanding the low density of feasible solutions in the search space. Results are detailed in figures 1 and 2, along with table 1. In general we find that as we incorporate more heuristic knowledge, the average quality of final solutions improves and run-times decrease. However, the variability of final solutions also decreases, and in fact the very best solutions are only discovered when starting with a random population.

5.2 Oil-field production scheduling

Quadstone has worked with British Petroleum to maximise the economic return from a group of interdependent oil and gas fields (Harding *et al.*, 1995). This involves setting the target production rates for each field in each year. A trade-off arises because earlier extraction offers earlier revenue, but incurs higher costs (since larger facilities are required to produce and handle the flow). There are also more subtle effects, such as the phasing of the start of production for the different fields, and the choice of when to abandon each field. The resulting optimisation problem involves searching for high-quality target production schedules, represented as real-valued matrices with over 200 entries, while satisfying a number of constraints. An existing solution produced by BP using expert knowledge, simulated annealing and sequential quadratic programming was made available to us. In Quadstone's previous work, a hybrid evolutionary technique with specialised operators was employed. A number of representations were considered, using both memetic and genetic approaches, and a large number of parameter sensitivity studies were carried out, resulting in a high-quality, tuned reproductive plan.

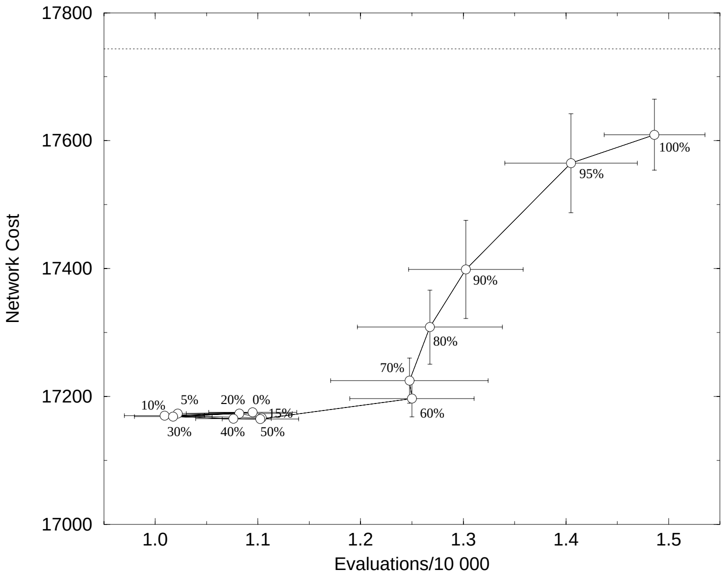

Fig. 2. This figure summarises the results shown in figure 1 (note different scale), but includes time to convergence (defined here as 5000 evaluations without improvement). Each point plots the mean best fitness versus mean number of evaluations at convergence for a variety of mass-mutation rates (100% yielding a fully random population, and 0% indicating inoculation with a single unmodified copy of the heuristic). Increasing homogeneity around the heuristic solution (dashed line) tends to increase *average* solution quality and to reduce expected run time, but this hides the fact that best *overall* solution quality over a number of runs actually deteriorates.

We used an algorithm with no local optimisation, population sizes of about 500 solutions and termination after approximately 250,000 evaluations. The results of initialisation experiments using the solution provided and the previously best known solution (found using a memetic algorithm) gave the results shown in figure 3. We again find that mean performance improves for intermediate mass-mutation rates but that variability correspondingly deteriorates.

5.3 Credit scoring

Institutions typically attempt to determine the credit-worthiness of applicants prior to issuing credit facilities. One method for doing this is to use a scorecard to rate an applicant's suitability. The applicant is rated in a number of categories, and these ratings are

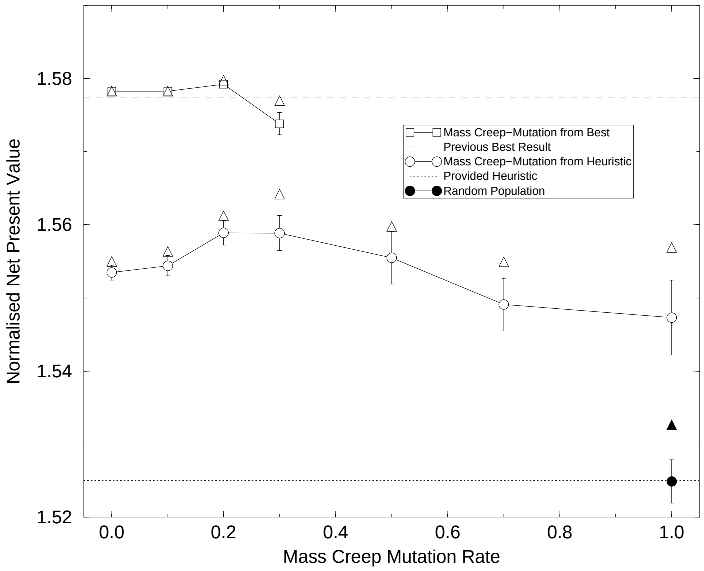

Fig. 3. The figure shows the results of initialisation experiments on an oil-field production-scheduling problem, involving more than 200 real variables. Two sets of experiments are shown, one using a supplied heuristic solution (dotted line) and one using the best known solution (found using a prior run; dashed line). Mean best performance after approximately 250,000 evaluations is plotted with standard errors, and the overall best is shown as a triangle. Mass creep-mutation is used so that even using a rate of 1.0 is not equivalent to starting with a random population (shown as filled symbols). Note the mid-range peaks and decreasing variance with decreasing initial population diversity (lower mass-mutation rates).

combined using a set of weights to yield an overall score for the applicant, upon which a decision can be based.

Quadstone has used a variety of evolutionary algorithms to produce enhanced credit-scoring models (Bruce & Radcliffe, 1995). For a linear (additive) scorecard, this involves finding a set of real-valued weights which maximises the predictive power of the scorecard over a database of customers whose credit-worthiness is known. The measure considered here is the Gini score, which indicates how well a proposed scorecard can separate good from bad risks, varying from -100% (perfectly reversed) to 100% (perfect), with random weights leading to scores of about zero.

In the work previously carried out, an existing scorecard was made available. This was used as a heuristic value with which to inoculate the initial population of 100 scorecards. Runs were carried out for 10,000 evaluations (with each requiring the processing and sorting of a database of about 10,000 records); however, greedy techniques are still

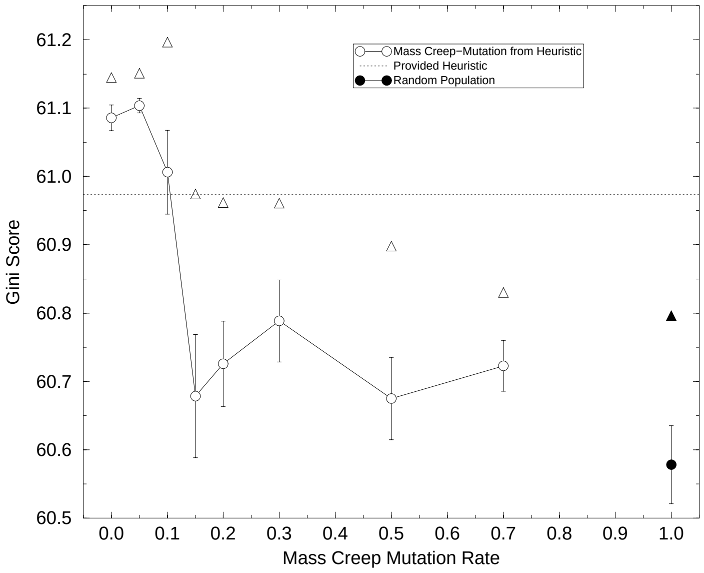

Fig. 4. The results of inoculating a population of scorecards with an existing heuristic solution (fitness of dotted line) are shown. The mean performance over several runs of the best scorecard found after 10,000 evaluations is plotted, along with with standard error bars and overall best (triangles). Since a creep mutation operator is used in the mass-mutation process, a rate of 1.0 is not equivalent to a random population (shown filled). This figure shows significantly more noise than the other results (figures 1, 3 and 5). This is due in part to averaging over fewer runs which are computationally expensive, but nevertheless similar trends are exhibited: mean performance generally decreases with more random initial populations, but spread increases.

competitive until significantly more evaluations are permitted. The results, which serve for comparative purposes, are summarised in figure 4. Although somewhat noisy, similar trends to the other experiments are apparent.

5.4 Travelling sales-rep

The travelling sales-rep problem is a well-known model of combinatorial optimisation problems, and has been widely studied in the literature on evolutionary computation. All of the usual representations and operators for tours used in evolutionary search make the problem appear to be constrained due to non-orthogonality (Radcliffe & Surry, 1994a)—the function being minimised (tour length) is a simple sum over allelic contributions with the difficulty arising in not being able to simply choose the n shortest edges.

A large number of heuristic techniques are known which are effective in finding

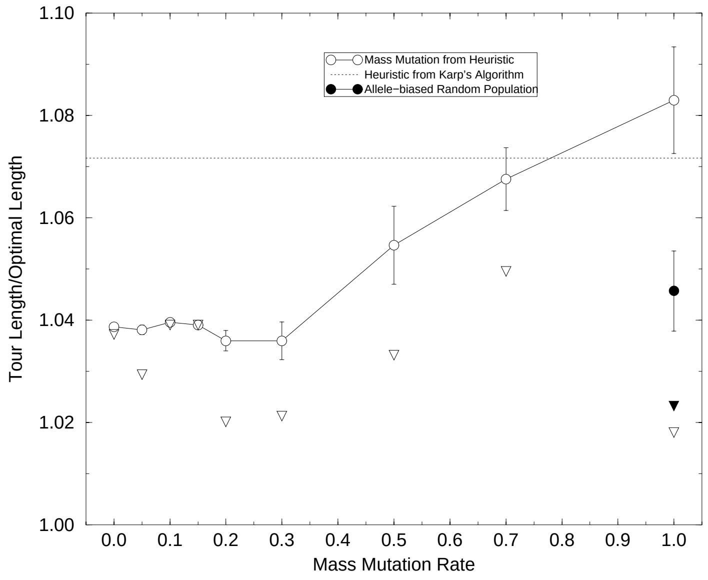

Fig. 5. The figure summarises the results of initialisation experiments on a small (100 city) travelling sales-rep problem using a simple genetic algorithm with no local search. The tour resulting from (deterministically) Karp-stitching all of the cities together was used as the seed tour (dotted line). The best value found after 100,000 tour evaluations is averaged and plotted along with standard error bars and the overall best (shortest) tour found over a series of runs. The filled points represent a population initialised with tours formed preferentially with short edges (not making use the heuristic). Note that the best tour found at rate 0.05 can be considered an outlier, as all other runs in that set converged to the same value (near the plotted mean).

near-optimal tours. For the purposes of the work here, we used Karp's stitching algorithm which recursively merges sub-tours to form a tour of all cities (a deterministic tour results from stitching the n "sub-tours" generated by considering each city as a single self-loop). The resulting tour was used in the inoculation experiments shown in figure 5. A simple genetic algorithm was used with populations of 100 tours and terminating after 50,000 evaluations. (Note that much larger problems can be solved much more efficiently by incorporating local search, but such was not the purpose of the experiments.)

An alternative initialisation procedure was also used, in which tours are not generated randomly, but instead favour short edges. Work is ongoing in this area, but preliminary results are extremely encouraging, particularly when the idea is extended to yield a fast partial local optimisation scheme.

Work on restart strategies for the TSP is also underway, and suggests that inoculation with a heuristic tour may be fundamentally different from restarting using a previously evolved tour. Various mass mutation techniques seem to either have either no or detrimental effects on the convergence properties of the algorithm.

6 Discussion

Through experiments on inoculation and mass mutation in a wide range of real-world problem domains, the existing folklore regarding initialisation has been explored. We find a tendency for mean performance (in terms of average solution quality and number of evaluations until convergence) to increase as initial populations become more tightly clustered around an existing high-quality solution, peaking at some intermediate value (mass-mutation rates of between 10% to 50%) and then trailing off slightly. However, this is coupled with a corresponding decrease in variance and often a deterioration in the quality of the best solution found over a number of runs. A particularly striking example of this is found in a pipe-sizing problem, in which the very best networks were only found when starting from a random population (see figure 1). There is also some indication that when multiple heuristics are available, best results are obtained when lower mutation rates are used with correspondingly higher-quality solutions.

Preliminary results also indicate that after the first few generations, there is little quantitative difference in diversity characteristics between algorithms using random initial populations and those using inoculated ones. For instance in both the TSP and pipe-sizing problems (sections 5.1 and 5.4), we observe little difference in measured first-order entropy (for each problem, the diversity of both inoculated and randomly initialised populations falls to 2–4% of a random population after a few generations). We do however see different convergence patterns, suggesting that diversity is not everything.

Planned future work includes investigation of the distinction between inoculation with externally provided solutions and restart strategies, recombinative forms of initialisation (as opposed to techniques based on mutation), the importance of representation and the study of fitness distributions near inoculants (using analysis of formae variance and/or fitness-distance correlation measures).

Acknowledgments

The authors would like to thank Alasdair Bruce and Dr. Timothy Harding for their assistance and encouragements in carrying out the reported experiments on credit scoring and oil-field production scheduling respectively.

References

David L. Battle and Michael D. Vose, 1991. Isomorphisms of genetic algorithms. In Gregory J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 242–251. Morgan Kaufmann (San Mateo).

Richard K. Belew, John McInerny, and Nicol N. Schraudolph, 1990. Evolving networks: Using the genetic algorithm with connectionist learning. Technical Report CS90–174, UCSD (La Jolla).

Ian D. Boyd, Patrick D. Surry, and Nicholas J. Radcliffe, 1994. Constrained gas network pipe sizing with genetic algorithms. Technical Report EPCC–TR94–11, Edinburgh Parallel Computing Centre.

Mark F. Bramlette, 1991. Initialization, mutation and selection methods in genetic algorithms for function optimization. In *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

R. Alasdair Bruce and Nicholas J. Radcliffe, 1995. Genetic scorecard optimisation. Quadstone Ltd. (unpublished technical report).

Lawrence Davis and David Orvosh, 1993. Shall we repair? Genetic algorithms, combinatorial optimization, and feasibility constraints. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

Lawrence Davis, 1991. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold (New York).

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

Michael de la Maza, 1989. A SEAGUL visits the race track. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

Larry J. Eshelman, 1990. The CHC adaptive search algorithm: How to have safe search when engaging in non-traditional recombination. In *Foundations of Genetic Algorithms*. Morgan Kaufmann (San Mateo).

Terence C. Fogarty, 1989. Varying the probability of mutation in the genetic algorithm. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

David E. Goldberg, Bradley Korb, and Kalyanmoy Deb, 1989. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3:493–530.

David E. Goldberg, 1989. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley (Reading, Mass).

John J. Grefenstette, 1987. Incorporating problem specific knowledge into genetic algorithms. In Lawrence Davis, editor, *Genetic Algorithms and Simulated Annealing*. Pitman (London).

T. J. Harding, P. R. King, and N. J. Radcliffe, 1995. Hydrocarbon production scheduling with genetic algorithms. submitted to J. Petr. Tech.

W. Daniel Hillis, 1991. Co-evolving parasites improve simulated evolution as an optimization procedure. In Stephanie Forrest, editor, *Emergent Computation*. MIT Press (Cambridge, MA).

John H. Holland, 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press (Ann Arbor).

Philip Husbands and Frank Mill, 1991. Simulated co-evolution as the mechanism for emergent planning and scheduling. In *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

Hillol Kargupta, 1995. *SEARCH, polynomial complexity, and the fast messy genetic algorithm*. PhD thesis, Univerity of Illinois at Urbana-Champaign. IlliGAL Report No 95008.

Zbigniew Michalewicz, 1992. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag (Berlin).

Pablo Moscato and Michael G. Norman, 1992. A “memetic” approach for the travelling salesman problem — implementation of a computational ecology for combinatorial optimisation on message-passing systems. In *Proceedings of the International Conference on Parallel Computing and Transputer Applications*. IOS Press (Amsterdam).

David J. Powell, Siu Shing Tong, and Michael M. Skolnick, 1989. EnGENEous domain independent, machine learning for design optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

Nicholas J. Radcliffe and Patrick D. Surry, 1994a. Fitness variance of formae and performance prediction. In L. Darrell Whitley and Michael D. Vose, editors, *Foundations of Genetic Algorithms III*, pages 51–72. Morgan Kaufmann (San Mateo, CA).

Nicholas J. Radcliffe and Patrick D. Surry, 1994b. Formal memetic algorithms. In Terence C. Fogarty, editor, *Evolutionary Computing: AISB Workshop*, pages 1–16. Springer-Verlag, Lecture Notes in Computer Science 865.

Nicholas J. Radcliffe and Patrick D. Surry, 1994c. The Reproductive Plan Language RPL2: Motivation, architecture and applications. In J. Stender, E. Hillebrand, and J. Kingdon, editors, *Genetic Algorithms in Optimisation, Simulation and Modelling*. IOS Press (Amsterdam).

Nicholas J. Radcliffe and Patrick D. Surry, 1995. Fundamental limitations on search algorithms: Evolutionary computing in perspective. In Jan van Leeuwen, editor, *Computer Science Today: Recent Trends and Developments, Lecture Notes in Computer Science, Volume 1000*, pages 275–291. Springer-Verlag (New York).

Nicholas J. Radcliffe, 1991. Equivalence class analysis of genetic algorithms. *Complex Systems*, 5(2):183–205.

Nicholas J. Radcliffe, 1992. Non-linear genetic representations. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature 2*, pages 259–268. Elsevier Science Publishers/North Holland (Amsterdam).

Connie Loggia Ramsey and John J. Grefenstette, 1993. Case-based initialization of genetic algorithms. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

Thomas S. Ray, 1994. An evolutionary approach to synthetic biology: Zen and the art of creating life. *Artificial Life*, 1(1/2):195–226.

N. Schraudolph and R. Belew, 1990. Dynamic parameter encoding for genetic algorithms. Technical Report CS90–175, University of California, San Diego.

Craig G. Shaefer and Stephen J. Smith, 1990. The ARGOT strategy II: Combinatorial optimizations. Technical Report RL90-1, Thinking Machines Inc.

Craig G. Shaefer, 1987. The ARGOT strategy: Adaptive Representation Genetic Optimizer technique. In *Proceedings of the Second International Conference on Genetic Algorithms*. Lawrence Erlbaum Associates (Hillsdale).

N. Sourlas, 1986. Statistical mechanics and the travelling salesman problem. *Europhysics Letters*, 2(12):919–923.

Patrick D. Surry, Nicholas J. Radcliffe, and Ian D. Boyd, 1995. A multi-objective approach to constrained optimisation of gas supply networks: The COMOGA method. In Terence C. Fogarty, editor, *Evolutionary Computing: AISB Workshop*, pages 166–180. Springer-Verlag, Lecture Notes in Computer Science 993.

Christine L. Valenzuela and Antonia J. Jones, 1994. Evolutionary divide and conquer (I): A novel genetic approach to the TSP. *Evolutionary Computation*, 1(4):313–334.

Darrell Whitley and Thomas Hanson, 1989. Optimizing neural networks using faster, more accurate genetic search. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 391–396. Morgan Kaufmann (San Mateo).

D. Whitley, K. Mathias, and P. Fitzhorn, 1991. Delta coding: An iterative search strategy for genetic algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

David H. Wolpert and William G. Macready, 1995. No free lunch theorems for search. Technical Report SFI–TR–95–02–010, Santa Fe Institute.

Anatoly A. Zhigljavsky, 1991. *Theory of Global Random Search*. Kluwer Academic Publishers.

This article was processed using the LaTeX macro package with LLNCS style