Finding Multiple Solutions with an Evolutionary Algorithm

Simon Ronald
School of Computer and Information Science
The University of South Australia
The Levels, South Australia, 5095
s.ronald@unisa.edu.au

ABSTRACT

A new multiple-solution technique is presented that addresses some of the limitations of existing speciation and multiple-solution techniques. This genetic-algorithm (GA) technique packs multiple problem points within a genotype and a uses a fitness function based on inter-solution distance and individual solution fitness. The technique is demonstrated on a contrived multimodal TSP test problem and it is found effective in finding two maximally distant and near-optimal solutions. The technique can be used with a generational or steady-state GA model and does not depend on the explicit use of crossover or a binary-based encoding. Therefore the technique may be of interest in other population-based computational models other than genetic algorithms.

1. A New Multiple-Solution Technique

Genetic Algorithms (GAs) with permutation-based encodings have been applied to problems which have solutions governed by a list of gene values. Such GAs manipulate these lists, and strive to find the optimum permutation of gene values for the problem. In the past, permutation GAs have been applied to numerous classical and real-world problems with the objective of converging on an optimal or near-optimal solution. Finding a single good solution has been the research focus in order-based problems. However, in an order-based domain, multiple solutions are often of interest to the practitioner. An algorithm capable of finding a number of different, high-quality solutions gives the practitioner the choice of one solution over another. In certain classes of design problems, this becomes an important benefit to the designer as one solution may be better than another for reasons completely outside of the cost model used in the objective function of the GA.

Speciation techniques have been employed in real-function optimisation to obtain a number of different but fit solutions of potential interest [3, 5]. For example, a control system may have an K-dimensional response function of which each dimension refers to some control-system parameter. The control engineer might wish to examine a number of fit peaks and choose a particular peak on the basis of system stability rather than choosing a parameter set which has the best global fitness. It may not be the absolute fitness of the peak, but the quality of the peak itself, the steepness of the shoulders and the proximity to other peaks or troughs, which may be criteria for selection of a particular solution.

Speciation techniques have recently been used in GAs that optimise over a discrete domain. For example in [5], Pham and Yang applied sharing functions to the design of a gearbox system. The GA was responsible for evolving a gearbox design (number of shafts and the number of teeth for each gear) which met the operational requirements. These requirements included a range of input motor speeds and the required output gearbox speed. The fitness function included penalty factors for violated constraints as well as positive fitness contributions for correct gear ratios, and for smaller number of shafts in the design. Goldberg's sharing-function technique [2] was used to find a small number of alternative gearbox designs which met the output speed requirements based on one of a number of alternative input motor speeds. The GA was run, and found a variety of solutions which were structurally different. The range of solutions returned by the GA were of use to the practitioner who could then use different solutions based on the availability of an input motor rated at a particular speed.

In permutation-encoded problems, a technique yielding a number of different, but fit, solutions enables the practitioner, i.e. a production engineer or train controller, to examine different alternatives to find desirable characteristics not incorporated into the fitness function. This would allow the comparison of one solution with another. For instance, the practitioner may wish to find a number of low cost, but different, job shop schedules, or to find a number of nearly optimal, but different, crossing plans in a train timetable.

A new multiple-solution technique is presented which addresses some of the restrictions of current multiple-solution techniques. The technique will be referred to as Multiple Solution Technique (MST). MST is designed to find a small number of good quality solutions which are as structurally different as the problem landscape permits. MST is not a speciation technique, as it does not strive to partition the genotypes into distinct species where a set of genotypes S_1 targets one peak, and another set S_2 targets a different peak in the problem space. Rather, MST uses the simple-GA evolution strategy and has two novelties: 1) a number of different solutions are contained in a single genotype, and 2) a special fitness function is used to evaluate the fitness of the compound genotype. The MST fitness function has two interrelated objectives: 1) maximise inter-solution distances and, 2) maximise individual solution fitnesses. It will be demonstrated that this results in a GA capable of multi-modal search. The main limitation of MST is its lack of ability to scale up to problems in which a large number of solutions are required by the practitioner, however, typically the practitioner is interested in a few near-optimal and structurally varied solution.

2. Multi-Modal Optimisation

2.1. Obtaining a Precise Number of Solutions

If the practitioner is interested in only s solutions, then deterministic crowding [4] and sharing function [2] techniques can prove to be troublesome. The number of solutions found by sharing functions depends on the number, location, and fitnesses of the maxima in the landscape, along with the population size, and the setting of the distance threshold parameter θ_{share}. The technique of deterministic crowding has no explicit control mechanism that would select and evolve towards precisely the number of solutions that are required. The sequential-niche technique [1] addresses the objective of obtaining the desired number of solutions, however, the choice of the r threshold parameter can either hide desired maxima or create false maxima in the problem space [6].

2.2. Aims of a New Multiple-Solution Technique

Ideally, a GA-based multiple-solution technique would devote all of its search effort only to the top s solutions desired by the practitioner. The sequential niche technique is a winner in this regard [1] however, there is no guarantee that the solutions found with this technique will represent a structurally diverse range [6]. The specific aims of a new multiple-solution technique relate to a technique that :

- Is not critically dependant of a niche-radius measure.
- The number of solutions required can be specified by a single parameter s.
- Devotes all of it's search effort only to the number of solutions required.
- Co-adapts the solutions found so that the solutions evolved are as distant from each other as the landscape allows.
- Gives as much search effort to locally optimum peaks (that are identified as required solutions), as is given to globally optimum peaks – even if the local peaks are a tiny fraction of the fitness of the global peak. For example, if the fitness landscape contains only two peaks, p_1 and p_2, and if two solutions are required, and if $f(p_1) \gg f(p_2)$ then the search effort should be evenly divided between p_1 and p_2.
- Can be effectively implemented on a parallel processor.
- Has a small time complexity relative to s, the number of solutions required.
- Scales up well with large values of s.

Existing multiple-solution techniques use an explicit control mechanism to induce the GA to find different, and fit solutions. It has been argued that sometimes this control mechanism gives unwanted behaviour when searching for multiple peaks in the problem space [6]. A case in point is the sequential-niche technique where the squashing function that eliminates previously located peaks can cause distortion peaks in the landscape [1]. In fact, these distortion peaks may be identified as solution peaks in their own right. The idea behind creating a new multiple-solution technique was to minimise the control mechanism and to transfer as much of the process of multi-modal search over to the GA itself. The proposed technique uses an expanded encoding resulting in an increased search space.

MST uses an encoding that encompasses all s desired solutions into a single genotype and the evolution proceeds to find exactly s solutions. The entire search effort is devoted to finding only those s solutions. Thus the technique, like deterministic crowding and sharing functions finds the solutions in a parallel fashion, and not sequentially like the sequential niche technique. An interesting side effect of such a scheme is that the population does not contain a mix of genotypes representative of different niches. This technique ensures that problem points that are attempting to target a particular solution ~~will almost always be~~ mated together. This is in contrast to deterministic crowding and sharing functions which can generate a large number of still-born (population worst) genotypes in the latter stages of evolution when parents in different niches mate together[1].

Each genotype is partitioned into s chunks, or subchromes with each subchrome encoding a single problem point. Mating occurs strictly within these subchrome boundaries such that the ith subchrome in genotype x always crosses over with the ith subchrome in genotype y.

2.3. The Fitness Function

The following notation is introduced: The genotype x contains s subchromes. The notation $x(j,k)$ refers to the gene value (allele), where j is the subchrome identifier, and k is the locus identifier relative to the start of subchrome j. The notation $x(j,*)$ refers to all alleles in the jth subchrome, and fully specifies the jth problem point encoded in genotype x. Function $f_\partial(x(j,*))$ denotes the fitness of the jth subchrome of individual x. The function $\overline{d}(,)$ as used in $\overline{d}(x(i,*),x(j,*))$ refers to the normalised distance between the ith and jth subchromes in x and lies in the range [0,1]. The distance d refers to a meaningful measure of distance, whether it be Euclidean if a parameterised real-domain is used, or otherwise. The d_f term is a constant in the range [0,1].

The fitness of a chromosome x is calculated according to the following formula

$$f(x) = \frac{1}{s}\left[\sum_{i=1}^{s} f_\partial(x(i,*)) \times \left(d_f + (1-d_f)\prod_{j=1}^{s,j\neq i} \overline{d}(x(i,*),x(j,*))\right)\right]$$

It can be seen from the form of the fitness equation (Equation 1) that each subchrome makes a contribution to the total fitness of the genotype x. However, the fitness of a subchrome $f_\partial(x(i,*))$ is effectively multiplied by the product of the distances between the ith subchrome and all of the other subchromes in the genotype. Therefore a maximum fitness value of 1 is awarded, as long as 1) all subchromes encode different points in the problem space and 2) each point encoded in every subchrome in x has a maximum fitness value of 1 (ie. there are s global optima) and 3) the distances between each problem point encoded in each subchrome of x is 1 (maximally distant).

The constant d_f controls the tradeoff between evolution towards the goal of fitness for each subchrome, and evolution towards the goal of achieving diversity between each subchrome, and is a value chosen in the range $[0,1]$.

2.4. Characteristics of the Multiple Solution Technique

The fitness function in Equation 1 is designed such that a genotype will be under pressure to evolve such that the distances between each of the encoded problem points within a genotype are large, and also that each problem point encoded has a large fitness. There is a dynamic tradeoff between distance and fitness within a genotype. The tradeoff can be controlled with an appropriate choice of the d_f term.

A second characteristic of this technique is that multiple solutions co-evolve in the population. This means the GA is able to find s solutions according to the distance tradeoff specified by the d_f. A low value of d_f (say 0.1) means that the search is to be directed towards fit peaks in the problem space with less regard for the distance between them. A high value of d_f (say 0.9), means that peaks are to be located (if possible) with very large distances between them. The co-adaptation that occurs makes it possible for these tradeoffs to be done intelligently according to the nature of the fitness landscape, and according to the value of d_f. It is argued that the choice of this parameter is in fact, less critical, than the choice of the niche-radius parameter in the sharing function [2] and sequential-niche techniques [1]. If d_f is kept at an intermediate value, i.e. 0.7, then this should allow a reasonable tradeoff between fitness and inter-solution distance for most problem types.

An important aspect of the new multiple-solution technique is that mating occurs between matching subchromes in the two parents. Towards the end of an evolution run, the kth subchrome in the genotypes contained in a population, target a peak in the problem space, and adaptation occurs towards that peak. Since crossover occurs wholly within sub-chromosome boundaries this peak is never crossed over with the other peaks encoded in the other subchrome locations. Evolution occurs within a subchrome boundary and the computation model resembles a simple GA, apart from the fitness function. This prevents a large number of still-born individuals which would otherwise be created in a niche-GA[2] and represents wasted computational effort.

The third characteristic of this technique is that the entire search effort is devoted to finding only s solutions. This is because a genotype only contains s subchromes with each subchrome encoding a single solution point. No other speciation or population-partitioning technique is used. This means that the entire search effort will be focussed on the s solutions.

[1] There have been approaches to fix this problem, these include niche clustering and mating restriction, each requiring an additional level of control.

[2] In a niche-GA two parents p_1 and p_2 that target different solutions often mate to beget a child lying in the low-fitness valley between p_1 and p_2 in the fitness landscape.

The multiple solution technique lends itself to parallel implementation. Once two parents p_1 and p_2 are chosen, then s crossover/mutation operations could be implemented in parallel to produce the children c_1 (and c_2). Then the fitnesses of each of the s problem points encoded in each child could be evaluated as one parallel operation. The $\frac{1}{2}s(s-1)$ distance function calculations could also be performed in parallel. In a generational GA model these three steps could be done in parallel with up to $\frac{N}{2} - 2$ of the other parents selected for mating (or mutation) in the population.

Distance function calculations in order-based encodings are more costly than those required for parameter-based encodings [6]. The multiple solution fitness function in Equation 1 requires $\frac{1}{2}s(s-1)$ distance calculations per genotype. If s is small, then this compares well with the sharing function technique where N (the population size) distance function calculations are required per genotype [2].

2.4.1. Summary of the Multiple Solution Technique

The limitations of the MST technique is that the total encoding space is exponential with respect to s. If Φ is the size of the encoding space for one subchrome, then a genotype encoding s subchromes forms an encoding space of size Φ^s. It is argued that the practitioner is often interested in a few fit, but structurally varied solutions so a choice of one solution over another can be made on an ad hoc basis. Therefore the MST technique is advocated for such situations where s is small.

This multiple solution technique meets all but the last of the objectives set down in Section 2.2. This multiple-solution technique will now be referred to more specifically as Multi-Chromosomal Cramping (MCC). The name was chosen as the fitness function is designed to award lower fitness values if the solutions encoded in the subchromes are cramped close together, according to a suitable distance measure, in the problem space.

3. A Sample Problem - The Travelling Salesperson Problem

3.1. Finding a Suitable Order-Based Distance Function

Many evolutionary-algorithm techniques require a measure of distance that takes two genotypes as input and returns a real-valued measure of distance as output. These techniques include mating restriction, incest prevention, speciation, and multiple solution methods. Distance functions exist in the literature for genotypes which use binary encodings [3]. Various phenotypic distance functions have also been presented [3, 5]. There is, however, a gap in the literature regarding distance functions for order-based encodings (permutations).

For a binary-encoded GA the simplest form of distance function is the Hamming distance [3]. The Hamming distance is a calculation of the number of differing bits in matching bit positions between two genotypes. However, this distance measure is less meaningful when the genotype is structured into parameters, where each parameter is decoded through a complex decoding scheme into the phenotype. In such a case, a phenotypic distance measure is more accurate. It was shown in [3], that for parameter encoded problems with a Euclidean-like problem space, a convenient measure is the Euclidean distance between matching parameters in the phenotype. This technique was designed for n dimensional fixed-parameter-encoded problems, where the parameter value difference $p_{x,k} - p_{y,k}$ refers to some meaningful measure of difference in both genotypes x and y and relates to the same problem-specific property encoded in the kth parameter. However, such a distance measure is not relevant for order-based genotypes. Order-based genotypes are characterised by adjacent gene values (edge based) or by the absolute location of a gene value within a genotype (positional). Other characteristics, such as the preservation of subsequences are discussed in [6].

The TSP problem landscape relates to problems where only near-neighbour gene adjacency is important. A cyclic distance function is defined for the TSP. This distance function considers the last gene value to be adjacent to the first gene value.

3.2. Cyclic Edge Distance

The edge distance measure considers the commonality in neighbouring genes between two individuals. TSP problems, for instance exhibit this property because the distance between two adjacent towns has a direct contribution to the cost of the TSP solution.

For the cyclic genotype case, the maximum number of common edges is l, the number of genes in the genotype. A measure of cyclic-edge distance is defined as $d^{ec}_{i,j}$ between genotypes x_i and x_j as

$$d^{ec}_{i,j} = l - e_{i,j} \tag{1}$$

Where $e_{i,j}$ is the number of common edges between genotypes x_i and x_j.

The edge distance measure has a maximum value of l when the two genotypes have no common edges ($d^{ec}_{max} = l$).

Whenever a new distance function is proposed it is important that it conforms with the formal rules required in a distance metric. This ensures that the distance function behaves correctly in a minimal sense. A formal definition of a distance func-

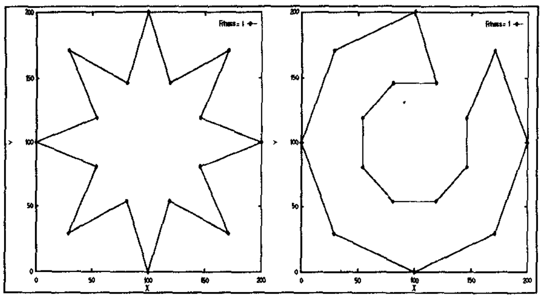

Fig. 1: Two anticipated global maxima of the T_1 problem *(left)* and *(right)*.

tion involves the compliance of five metric axioms. Each axiom must hold for the distance function $d(,)$ to be a valid distance metric. The metric axioms include non-negativity, symmetry, triangle inequality, and the identical case requiring zero-distance. A final metric axiom stipulates that if $i \neq j$, then $d_{i,j} > 0$. A proof for each of these metric axiom properties for this cyclic-edge distance can be found in [7].

4. Designing a Multimodal Order-Based Test Function

This section shows the MCC applied to a multimodal TSP test problem. The T_1 test problem was devised by selecting towns on the Euclidean plane in such a way that two very different global optima were expected. Integer town coordinates were designated on a 200 x 200 grid. This test problem is described by the coordinates

$$\begin{aligned} T_1 \;=\; & \{(0, 100), (29, 171), (29, 29), (54, 119), (54, 81), \\ & (81, 146), (81, 54), (100, 0), (100, 200), \\ & (119, 146), (119, 54), (146, 119), (146, 81), \\ & (171, 171), (171, 29), (200, 100)\} \end{aligned}$$

This particular test problem was chosen anticipating that it would contain two main structurally different and optimal circuits, as can be seen in Figure 1. The two circuits were designed to have only two common edges from a total of 16, as is shown by the superimposed circuits in Figure 2. Both circuits in Figure 1 are global optima. However a branch-and-bound algorithm revealed that the problem in fact contained 938 different *global* optima, with 64 pairs having no edges in common. These properties make T_1 a good multi-modal test problem for an algorithm that attempts to find a number of fit, but structurally different solutions.

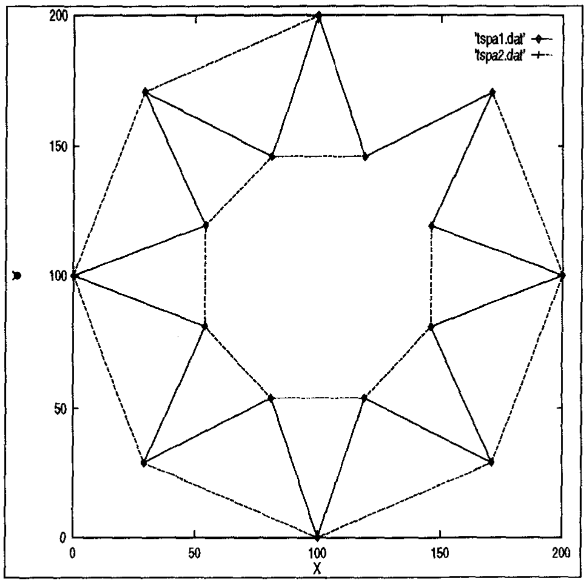

Fig. 2: The two anticipated globally optimum circuits superimposed to show the two common edges.

4.1. Experiment: Without Using the Multiple Solution Method

As an indication of the performance of a regular-GA on the T_1 problem 50 independent runs were performed. A population size of 200 was used with Whitley's Genetic Edge Recombination Operator employed for crossover. Mutation was not employed. Using a regular GA, the top solutions in each of the 50 end-of-run populations were recorded. The average best-member final-population fitness value was 1, indicating that each final population member was a global optimum solution. In the run with the most varied final-run population composition the two most distant genotypes had a cyclic edge distance of 0.25, and therefore shared 12 of the 16 edges between them. On inspection of each best final-population solution, it was found that each solution was identical, or a variant of the star-shaped solution in Figure 1 *(left)*[3]. This suggests that a specific multiple-solution technique must overcome the dominant maximum and find the other peaks of interest in the T_1 problem landscape.

4.2. Experiment: Using Multi-Chromosomal Cramping

The MCC method was applied using the GA parameters in Section 4.1. Figure 3 shows the best scoring pair out of 50 runs for the MCC technique. In this solution, no edges were shared between the two solutions. One of the solutions was a global

[3] A justification why this solution forms a dominant maximum is given in [6]

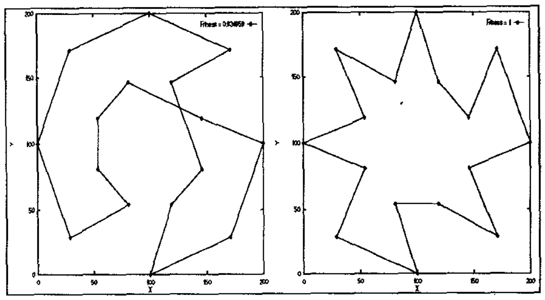

Fig. 3: Two highest-scoring final-population individuals at generation 20,000 when multi-chromosome cramping employed on the T_1 problem. No common edges.

optima, the other solution had a fitness of approximately 0.935. These two solutions are very close to the best possible fitness that can be achieved with the MCC technique, and represents an intelligent tradeoff between fitness and inter-solution distance. The other MCC GA runs evolved global solution pairs with at most three shared edges.

Other results that compare the MCC technique to deterministic crowding and sharing functions over ten order-based test functions (including five fully deceptive functions) can be found in [6].

5. Conclusions

A new technique called multi-chromosomal cramping has been proposed where a genotype encodes all of the desired s solutions. The technique has a number of advantages over other techniques. MCC focuses all of the search effort precisely on the s solutions, it is not sensitive to a niche-radius measure. The solutions are co-adapted with an intelligent tradeoff between fitness and distance that separates the solutions. MCC can be easily implemented in a parallel fashion, and is efficient for small values of s. The drawbacks of the technique are that the problem space increases exponentially with respect to s, and the distance function calculations increase as the square of s. These two considerations limit the use of MCC to small s values. However, these limitations do not rule out the use of the MCC technique as the practitioner is often interested in finding a few good solutions, to which the MCC technique is well suited.

Results were demonstrated on an order-based problem. A multimodal TSP test problem was constructed and the MCC algorithm found two solutions that were maximally distant and near optimal in the problem space. Other comparative results [6] indicate that MCC has complimentary strengths compared with other muti-solution techniques and is worthy of consideration in multimodal optimisation situations.

References

[1] D. Beasley, D. Bull, and R. Martin, "A sequential niche technique for multimodal function optimization," *Evolutionary Computation*, vol. 1(2), pp. 101–125, 1993.

[2] D. Goldberg, "Genetic algorithms with sharing for multimodal function optimization," *Genetic Algorithms and their Applications. Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41–49, 1987. Boston, MA.

[3] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[4] S. Mahfoud, "Crowding and preselection revisited," IlliGAL Report No. 92004, Department of Computer Science. University of Illinois at Urbana-Champaign, 1304 West Springfield Avenue, Urbana, IL 61801, Apr 1992.

[5] D. Pham and Y. Yang, "Optimization of multimodal discrete functions using genetic algorithms," *Journal of Automobile Engineering*, vol. 207, no. 1, pp. 53–59, 1993.

[6] S. Ronald, *Genetic algorithms and permutation-encoded problems. Diversity Preservation and a Study of Multi-Modality.* PhD thesis, University South Australia. The Department of Computer and Information Science, Full Draft (and/or abstract) available by SEED Web Server, http://arch.adelaide.edu.au/. Hard copy by request - E-mail s.ronald@unisa.edu.au, 1995.

[7] S. Ronald, "Genetic algorithms and scheduling problems," in *The Practical Handbook of Genetic Algorithms. Volume 1*, (L. Chambers, ed.), (Boca Raton, Florida), pp. 367–430, CRC Press, 1995.