Useful Diversity via Multiploidy

Emma Collingwood

Dept of Artificial Intelligence University of Edinburgh Edinburgh, UK emmac@aisb.ed.ac.uk David Corne

Dept of Computer Science University of Reading Reading, UK D.W.Corne@reading.ac.uk

Peter Ross

Dept of Artificial Intelligence University of Edinburgh Edinburgh, UK peter@aisb.ed.ac.uk

Abstract— A multiploid genetic algorithm (GA) incorporates several candidates for each gene within a single genotype, and uses some form of dominance mechanism (most simply, an encoded choice) to decide which choice of each gene is active in the phenotype. We explore a simple multiploid model. Investigation with two simplified test problems is reported, respectively suggesting certain strengths and weaknesses of employing multiploidy. In particular, multiploidy appears useful in cases where attractive suboptima are profoundly Hamming distant from the true optimum, thus requiring a GA to recover substantial lost material in order to recover from suboptima. This is distinct from cases where a GA's difficulty in solving a problem is, for example, more concerned with appropriately combining genetic material than finding it.

I. INTRODUCTION

Most implementations of Genetic Algorithms (GAs) are based on populations composed of single chromosome (haploid) genotypes. However, in nature we find that many organisms have *poly-ploid* genotypes, which consist of multiple sets of chromosomes with some mechanism for determining which gene is expressed, i.e. is *dominant* at each locus. This mechanism seems to confer a number of advantages on a system, mainly by enhancing population diversity; currently unused genes remain in a multiploid genotype, unexpressed, but shielded from extinction until they may later become useful.

A similar mechanism promises to benefit a genetic algorithm (GA). In a standard GA, the artificial evolution process gradually decreases population diversity, often leading to premature convergence. The multiploid genotype may provide a method of maintaining enough useful diversity in the population to often thwart this effect. As ever, better overall results would have to come at the expense of extra computational time and/or space usage. For example, a multiploid population may contain a large amount of ostensibly redundant information which has to be maintained by the algorithm. Also, any diversity-enhancing mechanism (for example, increasing the population size in an ordinary GA, or increasing the mutation rate) slows down convergence speed. In order to be useful, an applied multiploid GA should counteract these costs with better quality overall results. By investigating two simple test problems in this paper, we find results which suggest how we may distinguish applications in which the cost is indeed worthwhile from others where multiploidy will provide no useful benefit.

mask:	0	0	0	1	1	1	2	2	2
chromosome[0]:	\mathbf{a}	\mathbf{a}	\mathbf{a}	а	a	а	а	а	а
chromosome[1]:	\mathbf{b}	b							
chromosome[2]:	с	с	с	с	с	с	\mathbf{c}	\mathbf{c}	С
phenotype :	a	a	a	\mathbf{b}	\mathbf{b}	\mathbf{b}	с	с	с

Fig. 1. Multiploid Type 1

mask:	$0\ 1\ 2$
chromosome[0]:	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
chromosome[1]:	b b b b b b b b b
chromosome[2]:	c c c c c c c c c c
phenotype :	aaabbbccc

Fig. 2. Multiploid Type 2

A. Related Work

Early research into GAs tested a number of methods of introducing the notion of diploidy (double chromosomes) into the genotype; see [3] for a summary. But such tended to concentrate on dominance mechanisms rather than comparison with haploid GA performance, for example [1]. More recently, Yoshida & Adachi [5] look at a diploid GA in the context of solving non-stationary problems.

Work on comparing such models with standard GAs has been done by Dasgupta & MacGregor [2], who use a structured GA in which a genotype is a hierarchical structure; high level genes activate or deactivate low level genes, and so on recursively, to a limited depth. Genes that are not active remain in the structure and are carried through to subsequent generations. Dasgupta & MacGregor have generally found this method successful on a variety of problems, though do not report any particular weaknesses of the model. The multiploid model tested here is conceptually simpler than Dasgupta & MacGregor's 'structured GA', which seems justified by the lack of evidence for any particular need for multiple recursive levels of gene-activation. For example, reports on the use of the 'structured GA' tend not to extend beyond two levels.

II. The Multiploid Model

A multiploid genotype, shown in figure 1, contains p

chromosomes, each of length L, and a mask which specifies which of the p chromosomes has the dominant gene at a particular position in the chromosome. This information is decoded to yield the phenotype as follows:

An allele value of a at locus i in the mask denotes that the *i*th gene in the chromosome with index a becomes the *i*th gene of the phenotype. The mask length can be shorter than the length of the chromosomes, as for example in figure 2. In this case, if the mask length is m and the chromosome length L, then a gene at locus i in the mask with a value of a indicates that the *i*th set of L/m consecutive genes in the *a*th chromosome are dominant.

III. THE ALGORITHM

A population of multiploids, each of ploidy p is created by randomly generating each mask using alleles in the range 0 - p - 1, and generating each of the p chromosomes in each structure using alleles from a suitable range.

Multiploids are selected for reproduction using a conventional selection technique; in this study, rank based selection was used [4]. In a typical GA run, some form of traditional crossover is applied to x% of the p chromosomes in the multiploid and (optionally) to the mask to produce a child. The x% of chromosomes to cross are chosen at random — crossover is always performed by crossing chrom[i] of parent 1 with chrom[i] of parent 2 to produce chrom[i] of the child. If chrom[i] is not chosen for crossover, then chrom[i] of the child is produced by randomly selecting chrom[i] from one of the parents. Similarly, if the mask is not crossed, then one of the parent masks is chosen at random to become the child mask. The resulting child multiploid contains the same number of chromosomes as its parents.

Mutation operators can then be applied optionally to the mask and to any or all of the chromosomes in the new child. Note that operators are *not* applied to the phenotype — this is always constructed using the information from the mask and chromosomes. Convergence of the population is defined here (arbitrarily) as the point at which all phenotypes are identical - at this point however the population is still genotypically diverse.

IV. EXPERIMENTS - OVERVIEW

Experiments were performed on two test problems, each of which presents difficulties for a standard GA for different reasons. The first of these, coined the Indecisive(k)problem, comprises a landscape with multiple suboptima and one true optimum, all maximally Hamming distant from each other. The true optimum is only slightly better than the suboptima, and this difference is very difficult for schema processing to latch on to early on in a GA run. A standard GA or hillclimber can be expected to converge on any one of the suboptima with a chance equal to its converging on the true optimum. The difficulty for the GA lies in the attractiveness of the suboptima, drawing search away from the true optimum, which in this case amounts to steadily removing all but traces (at most) of the genetic material required to recover progress towards the true op-

Ploidy	% Success Rate in
	finding optimum sol'n
1 (haploid)	60%
2	96%
3	100%
4	98%
5	98%
6	98%
7	100%
8	82%
9	100%
10	98%

Table 1. Indecisive(1) Problem

timum.

The second problem was the well-known 'Max' (or 'unitation') problem, in which the fitness of a binary string is simply the number of '1's in the string. This a very simple problem which is known to present a challenge to a GA; the difficulty lies in converting the final few '0's to '1's towards the end of a run. Using gene-wise mutation in particular, in which mutation involves a small chance of individually altering any gene in the string, useful mutations at this stage are likely to be accompanied by destructive mutations elsewhere in the string. So, rather than a problem of lost genetic material, the challenge presented by the Max problem concerns judicious use of genetic operators.

V. The Indecisive(k) Problem

In this problem, alleles can take values from 0 to k. Fitness depends on the 'allele-counts'. For example, the allele count of allele j, $0 \le j \le k$ is the number of j's in the string. Allele counts are obtained for each allele, subtracting 1 from each such count for alleles j in the range $0 \le j < k$. The highest count then becomes the fitness. An 'all ks' chromosome is thus the true optimum, while the k strings 'all 0's', 'all 1's', and so on up to k - 1, are attractive suboptima, all maximally Hamming distant from each other and from the true optimum.

A haploid GA would be expected to find the optimum solution around 1/(k + 1) of the time. We expected the multiploid GA to improve on this. The intuition behind this is that in a multiploid GA, it seems likely that the alleles appropriate to the true optimum will not be so quickly and irretrievably lost. Even if phenotypic convergence to a suboptimum starts to occur, ample genetic material appropriate to the true optimum should stay around which, in conjunction with serendipitous results of genetic operations, still have a chance of re-emergence.

Two series of experiments were performed: in the first, a haploid GA using a population of 50 chromosomes was compared to the multiploid GA with a population of 50 multiploid units; the ploidy of these units was varied from 2 to 10. In the second, the population size for each of the multiploid tests was reduced in proportion to ploidy, so that in each experiment the populations contained an equivalent amount of raw genetic material. Thus a haploid GA with population 100 was compared to a diploid GA with population 50, and so on. In each experiment, the chromosome length was 20, and the experiment was

Ploidy	Population	Success Rate of
	size	finding Optimum Sol'n
1	100	66%
2	50	96%
3	33	94%
4	25	78%
5	20	62%
10	10	24%

Table 2. Indecisive(1): Population size reduced in relation to ploidy

repeated 50 times.

In each experiments, a steady state reproductive strategy was used, with rank based selection, as in Whitley's GEN-ITOR [4]. The operators applied are two-point crossover, and gene-wise mutation, with a mutation rate of 0.02. In the results reported below, crossover and mutation are applied to all of the chromosomes of the parent genotypes as well as the masks.

A. Results and Discussion

The results for Indecisive(1), given in table 1, clearly show that all values of ploidy are able to greatly outperform the haploid GA. For Indecisive(2), shown in table 2, and for ploidy values greater than 2, the multiploid is similarly more capable of finding the optimum than a haploid GA.

Table 3 shows that multiploid populations containing the equivalent amount of genetic material as a haploid population still achieve greater success rates. With very high ploidy, and hence a very small population of multiploids, the actual number of optimal solutions is low merely as a result of the very low population size. However, at the point of convergence, the best solution in *each* of the 50 trials was always dominated by '1' alleles.

The inference is that the multiploid GA is able to recover from early genetic drift, where 'good' genes become lost in the initial selection process. Good genes manage to remain in the population, shielded from any harmful over selection of 'bad' genes, and are available for reintroduction into the phenotype at a later stage. In a haploid GA, the only way to recover from this form of drift is by a highly unlikely series of chance mutations.

Looking further into the ability of the multiploid GA to recover lost material, we note that there are at least two possible mechanisms by which this can occur: First, operations on the mask can produce a new mask whose genes point at good genes that were previously recessive; second, operations on the chromosomes can introduce new 'good' genes into the chromosome so that mask genes previously pointing at a locus that contained a poor gene now point at a good gene in the same locus.

Experiments on Indecisive(1) and Indecisive(2) in which a) the mask was fixed and the chromosomes allowed to evolve and b) the chromosomes were fixed and the mask allowed to evolve, tended to show that *both* methods were independently capable of improved results over a haploid GA on the Indecisive(1) problem. This suggests that the combination of these two routes towards gene-recovery in an unrestricted multiploid GA comprises a particularly powerful and flexible system. The genotype is able to negotiate

ſ	Ploidy	% Success Rate in	I
		finding optimum sol'n	
	1	8%	Ī
	2	2%	I
	4	56%	ļ
	6	44%	l
	8	54%	I
1	10	72%	ł

Table 3. Indecisive(2) Problem



Fig. 3. Max Problem: Crossover only

conceptually separate but related routes towards optima.

VI. THE MAX PROBLEM

The efficiency of the multiploid was first investigated by comparing the performances of a multiploid and a haploid GA on a Max problem in which the only operator applied to the chromosomes was 2-point crossover. Both GAs would be expected to converge rapidly — the average fitness of the multiploid population however ought to be higher at convergence, and hence the GA should be less likely to need to rely on mutation operators later in the evolution. The population size in each case was 50 and the length of the chromosomes and mask 300.

The experiments were then repeated, adding the mutation operator into the GA with a probability of 0.02. Evolution was halted when either a genotype in the population had reached maximum fitness, or after a maximum of 40000 evaluations, and the average solution fitness of the best solution obtained in each of 50 trials calculated.

A. Results

The results from the set of experiments where the only operator was crossover (shown in figure 3) are interesting -all values of ploidy perform significantly better than the haploid (shown in t-tests), but the actual value of the ploidy seems to make very little difference. Moreover, a haploid GA with double the population size achieves a significantly higher fitness of 227.2.

Adding in mutation reveals that the multiploid populations evolve more slowly than the haploid GA, and hence after an equivalent number of evaluations the average fitness of the populations were lower than that of the haploid. The haploid GA found the optimum answer on just over 50% of tests in an average of approximately 36000 evalu-

Ploidy	Average Sol'n	Average No.
_	Fitness	Evaluations
1	299.6	35,920
2	289.54	40,000
3	289.3	40,000
4	289.02	40,000
5	288.92	40,000
6	289.6	40,000
7	289.28	40,000
8	289.18	40,000
9	289.26	40,000
10	290.0	40,000

Table 4. Max Problem: Crossover plus Mutation

ations — the optimum was never reached by a multiploid GA in 40,000 evaluations (although the GA had never converged at this point). These results are shown in table 4. Again we see little difference between ploidy values.

We can suggest an explanation for the general lack of improvement provided by multiploidy on the Max problem as follows. As we have seen with the Indecisive problem, and also amenable to intuition, the multiploid method supports diverse search by way of saving useful genetic material from being lost. In a case where the relevant material can be recovered readily with a normal GA anyway, however, this support would seem to be superfluous, and adds little. In the Max case, the basic units which need to be found are '1 alleles at loci currently sporting '0' alleles; mutation very readily provides a source of such material. The problem towards the end of a GA run on the Max problem, however, is that useful mutations will often be accompanied by detrimental ones elsewhere in the genotype. Multiploidy offers no particular support for this difficulty.

In the Indecisive problem, however, the difficulty facing a GA converging on a suboptimal solution is to somehow recover substantial collections of genetic material appropriate to the true optimum. The multiploid model helps directly with this, while the haploid GA would have a vanishingly small chance otherwise. In Max, however, a haploid GA readily possesses the ability to incrementally acquire the missing material (hence multiploidy offers little or no benefit) but is faced with logistic difficulties finding this material at appropriate moments.

VII. CONCLUSION

Our findings can be summarised in the following tentative hypothesis: Multiploidy appears useful in precisely those cases where useful genetic material may otherwise be irretrievably lost. This is the central point underlying judicious use of a simple multiploid model for optimisation. The import of this statement lies in recognising the idea that multiploidy will not be helpful in cases where the GA does not find it difficult to find the needed material (eg: '1' alleles in the Max problem). In such a case, either a haploid GA solves the problem adequately anyway, or there are other difficulties, such as there being only a very limited probability of combining the material in the appropriate way. The multiploid seems unable to effectively help in such cases, and so the extra computational burden of the multiploid algorithmic baggage is not cost-effective. Cases where necessary genetic material may be rather more difficult for a standard GA to find, however, include situations in which various high-order schemata are needed. For example, a collection of five contiguous '1' alleles in an Indecisive(1) GA run fast converging to an 'all 0's' solution. Here, the '1's solution is virtually irretrievable by the simple GA alone, since the probability of chance mutation producing a genome which both contains such material, and is fit enough to remain in the population long enough for useful further processing, is very small. More particularly, such material is not retrievable incrementally; mutations of a single or limited number of '0' alleles to '1' alleles, in the general area of the 'all 0's' suboptimum, produces less fit children which are therefore unlikely to be selected again for reproduction.

In conclusion, multiploidy, like many other methods which attempt to enhance the basic GA, is occasionally useful and occasionally superfluous or detrimental, depending on the problem landscape at hand. Here, by examining some simple well-understood problem landscapes, we have attempted to understand when multiploidy may or not be useful in terms of certain very general features of the problem landscape. Tentative conclusions from explorations so far suggest that multiploidy is of benefit when there exist optima whose discovery requires fairly high order building blocks, but attractive suboptima exist which do not use these building blocks. In particular, an element of deception should exist which impedes a standard incremental route between suboptima and optima.

References

- J.D. Bagley. The Behaviour of Adaptive Systems Which Employ Genetic and Correlation Algorithms. PhD thesis, University of Michigan, 1967.
- [2] Dipankar Dasgupta and Douglas R. McGregor. Using structured genetic algorithms for solving deceptive problems. Technical report, University of Strathclyde Department of Computer Science, 1993.
- [3] D.E. Goldberg and R.E. Smith. Nonstationary function optimisation using genetic algorithms with dominance and diploidy. In J.J Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*. San Mateo: Morgan Kaufmann, 1987.
- [4] Darrell Whitley. The GENITOR algorithm and selection pressure. In J. D. Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, pages 161–121. San Mateo: Morgan Kaufmann, 1989.
- [5] Yukiko Yoshida and Nobue Adachi. A diploid genetic algorithm for preserving population diversity — pseudo-meiosis ga. In Manner Davidor, Schwefel, editor, Parallel Problem Solving from Nature: PPSN III, pages 36-45. Springer-Verlag, 1994.