A Comparison of Dominance Mechanisms and Simple Mutation on Non-Stationary Problems

Jonathan Lewis,* Emma Hart, Graeme Ritchie

Department of Artificial Intelligence, University of Edinburgh,
Edinburgh EH1 2QL, Scotland

Abstract. It is sometimes claimed that genetic algorithms using diploid representations will be more suitable for problems in which the environment changes from time to time, as the additional information stored in the double chromosome will ensure diversity, which in turn allows the system to respond more quickly and robustly to a change in the fitness function. We have tested various diploid algorithms, with and without mechanisms for dominance change, on non-stationary problems, and conclude that some form of dominance change is essential, as a diploid encoding is not enough in itself to allow flexible response to change. Moreover, a haploid method which randomly mutates chromosomes whose fitness has fallen sharply also performs well on these problems.

1 Introduction

Genetic algorithms (GAs) are often used to tackle problems which are *stationary*, in that the success criteria embodied in the fitness function do not change in the course of the computation. In a *non-stationary* problem, the environment may fluctuate, resulting in sharp changes in the fitness of a chromosome from one cycle to the next. It is sometimes claimed that a diploid encoding of a problem is particularly suited to non-stationary situations, as the additional information stored in the genotype provides a latent source of *diversity* in the population, even where the phenotypes may show very little diversity. This genotypic diversity, it is argued, will allow the population to respond more quickly and effectively when the fitness function changes. As well as incorporating diversity, it may be possible for a diploid to maintain some kind of long term *memory*, which enables it to quickly adapt to changing environments by remembering past solutions.

The effectiveness of a diploid GA may depend on the exact details of its dominance scheme. Moreover, where a changing environment is the central issue, it is important to consider changes which affect the dominance behaviour of chromosomes over time, as this may provide added flexibility.

We have carried out tests on a number of diploid methods, including some where dominance change can occur. We found that diploid schemes without some form of dominance change are not significantly better than haploid GAs

* Now at School of Mathematical and Computational Sciences, University of St Andrews, St Andrews KY16 9SS, Scotland.

for non-stationary problems, but certain dominance change mechanisms produce a distinct improvement. Also, certain representations are very effective at maintaining memory, whilst others are more effective at maintaining diversity. Thus, the nature of the non-stationary problem may influence the methodology chosen. Furthermore, we show that in some situations, a haploid GA with a suitable mutation mechanism is equally effective.

2 Previous work

Ng and Wong[4] describe a diploid representation with simple dominance change, as follows (for simplicity, we will confine our attention to phenotypes which are strings of 0s and 1s). There are 4 genotypic alleles: 1, 0(dominant) and i, o(recessive). The expressed gene always takes the value of the dominant allele. If there is a contention between two dominant or two recessive alleles, then one of the two alleles is arbitrarily chosen to be expressed. The dominance mapping to compute phenotype from genotype is shown in figure 1, where "0/1" indicates an equal probability of either value. The occurrence of $1i$ or $0o$ is prohibited — if this does occur, the recessive gene is promoted to be dominant in the genotype. This last stipulation is a simple form of dominance change observed in nature in which recessive genes tend to be eliminated, over time, in favour of their dominant counterparts. We will refer to this arrangement as "*basic Ng-Wong*".

	0	o	1	i
0	**0**	**0**	**0/1**	**0**
o	**0**	**0**	**1**	**0/1**
1	**0/1**	**1**	**1**	**1**
i	**0**	**0/1**	**1**	**1**

Fig. 1. Ng-Wong

	A	B	C	D
A	**0**	**0**	**0**	**1**
B	**0**	**0**	**0**	**1**
C	**0**	**0**	**1**	**1**
D	**1**	**1**	**1**	**1**

Fig. 2. Additive

Ryan [5] proposed a notion of *additive dominance*. In this scheme, the genotypic alleles are regarded as having quasi-numeric (or at least ordered) values, and these values are combined using some suitably designed form of pseudo-arithmetic, with the resulting phenotypic allele depending on the value of this "addition". One way to effect this scheme is to associate actual numbers with the genotype alleles, and then apply some threshold to the result. Ryan uses 4 genotypic values A, B, C, D, and allocates these the values 2, 3, 7 and 9 respectively, with any result greater than 10 being mapped to 1 and lower values mapped to 0. The resulting dominance map is shown in figure 2.

In both these schemes, the probability of creating a phenotypic 0 is exactly 0.5, and hence the mapping in each case is unbiased.

Other forms of dominance exist which are not explored in this paper. These include using a "dominance mask", for instance, [1, 2], or implementing a form of meiosis, as observed in natural systems, in which a haploid chromosome is produced from a chromosome pair via recombination operators, for instance [6]. See [3] for further discussion of some of the issues.

3 Dominance Change

In natural systems, dominance can change over time, as a result of the presence or absence of particular enzymes. Ng and Wong [4] define a specific condition for dominance change to occur (which we adopt in this paper for all our dominance change methods): if the fitness of a population member drops by a particular percentage Δ between successive evaluation cycles, then the dominance status of the alleles in the genotype of that member is altered. That is, the dominance *mapping* for computing the phenotype does not change, but the allele values alter their dominance characteristics.

Dominance change is achieved in the Ng-Wong diploid by inverting the dominance values of all allele-pairs, such that 11 becomes ii, 00 becomes oo, $1o$ becomes $i0$ and vice versa. It can be shown that this results in a probability $3/8$ of obtaining a 1 in the phenotype where there was originally a 0, after applying the inversion. We will refer to this method as "*Full-Ng-Wong*".

We have extended Ryan's additive GA by adding a similar dominance change mechanism, in which the genotypic alleles are promoted or demoted by a single grade. Thus demoting 'B' by 1 grade makes it an 'A' whereas promoting it makes it a 'C'. Furthermore 'A' cannot be demoted, and 'D' cannot be promoted. For each locus, we choose at random one of the two genotypic alleles and then use the following procedure:

- If the phenotypic expression at this locus is '1' then demote the chosen genotypic allele by one grade, unless it is an 'A'.
- If the phenotypic expression at this locus is '0' then promote the chosen genotypic allele by one grade, unless it is 'D'

It can be proved that this "*Extended-Additive*" method results in a $3/8$ probability of changing a phenotypic 0 to a phenotypic 1.

Finally, we introduce a comparable "recovery" mechanism for the haploid GA, in which a bit-flip mutation operator is applied to each locus of the haploid genotype with probability $3/8$, whenever a decrease of Δ in the fitness of that individual is observed between successive generations.

The Extended-Additive and Haploid-Recovery schemes have been designed with a $3/8$ probability of flipping a phenotypic 0 to a 1 after a change in dominance so as to make them exactly comparable with the Full-Ng-Wong method.

4 Experiments

Methods tested. To investigate the benefit of a dominance change mechanism, we tested the Simple Additive and Basic Ng-Wong GAs (Section 2 above) *without* dominance change, and also an ordinary haploid GA with mutation rate 0.01. The dominance change GAs tested were those described in Section 3 above: Full-Ng-Wong, Extended-Additive and Haploid-Recover.

Parameters. All GAs were run with population size 150. Rank selection was used, with uniform cross-over, steady-state reproduction, and mutation rate 0.01. During crossover of diploid genotypes, chromosome I of the first parent diploid was always crossed with chromosome I of the second parent diploid. The threshold Δ for applying dominance change (Full-Ng-Wong and Extended-Additive) or recovery mutation (for Haploid-Recover) was a drop of 20% in the fitness of a phenotype. The modified version of an individual replaced the original with probability 1.0 if the modified version was no less fit; otherwise with probability 0.5. Each experiment was repeated 50 times, and the results averaged.

Test Problems. The GAs were tested on an oscillating version of the commonly known single knapsack problem. The object is to fill a knapsack using a subset of objects from an available set of size n, such that the sum of object weights is as close as possible to the target weight t. In the oscillating version, the target oscillates between two values t_1 and t_2 every o generations. A solution is represented by a phenotype of length n, where each gene x_i has a value 0 or 1, indicating if the object is to be included in the knapsack. The fitness f of any solution $\overline{x}$ is defined by

$$f(\overline{x}) = \frac{1}{1 + |target - \sum_{i=1}^{n} w_i x_i|}$$

In the following experiments, 14 objects were used. Each object had a weight $w_i = 2^i$, where i ranged from 0 to 13. This ensures that any randomly chosen target is attainable by a unique combination of objects. Two targets were chosen at random, given the condition that at least half their bits should differ. The actual targets used were 12643 and 2837, which have a Hamming separation of 9. The target weight was changed every 1500 generations. Each period of 1500 generations is referred to as an *oscillatory period* in the remainder of the text.

5 Results

5.1 Oscillating Knapsack, Fixed Targets – Simple Diploidy

The results for the basic GAs are shown in figures 3, 4 and 5. Simple Additive and the haploid GA perform very poorly for both targets after the first target change. The Basic Ng-Wong GA makes better progress towards finding a solution for the first target value, but never manages to find a solution for the second target that has fitness greater than 0.05. Clearly, diploidy alone does not maintain sufficient diversity to allow readjustment to a new target.

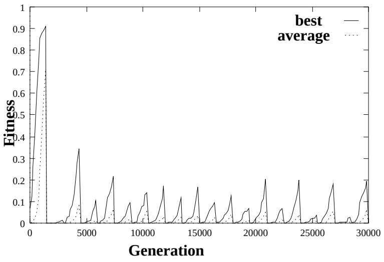

Fig. 3. Simple Haploid GA with Fixed Target Oscillation

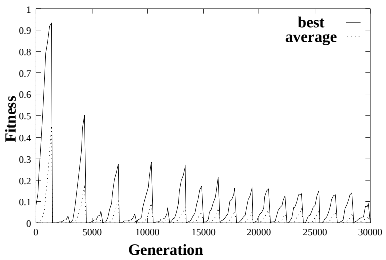

Fig. 4. Ryan's Additive GA with Fixed Target Oscillation

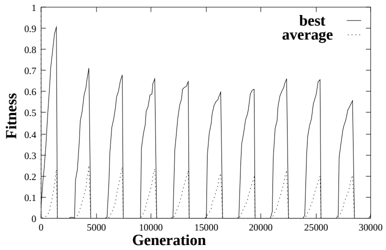

Fig. 5. Basic Ng-Wong with Fixed Target Oscillation

5.2 Oscillating Knapsack, Fixed Targets – Dominance Change

Figures 6, 7, and 8 show the averaged fitness over the 50 runs, plotted against generation for each of the 3 GAs. Each graph shows the best and average fitness of the population at each generation. Table 1 shows the number of the 50 experiments in which the optimal fitness of 1 was attained during each oscillatory period.

	Oscillation Period									
	1	*2*	3	*4*	5	*6*	7	*8*	9	*10*
Haploid-Recover	45	*44*	33	*45*	33	*44*	29	*43*	37	*47*
Extended-Additive	43	*29*	44	*42*	39	*40*	45	*37*	39	*40*
Ng-Wong	32	*21*	41	*25*	34	*27*	32	*26*	32	*27*

Table 1. Number of instances in which optimum was achieved in each period. Periods in which the target was 2837 (low) are shown in italics.

Comparison of the graphs obtained for Extended-Additive and Haploid-Recover show very similar performance. Extended-Additive finds a solution within 20% of the optimal fitness (i.e. > 0.8)in 90% of oscillation periods, compared to

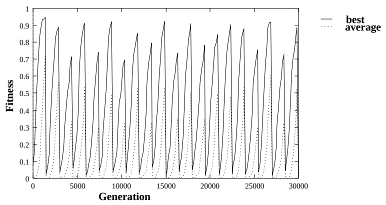

Fig. 6. Haploid-Recover with Fixed Target Oscillation

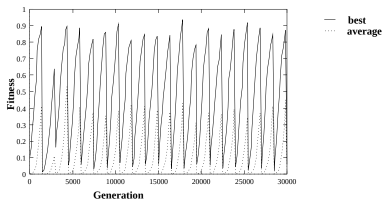

Fig. 7. Extended-Additive with Fixed Target Oscillation

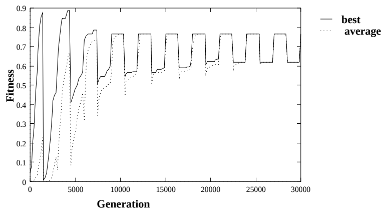

Fig. 8. Full-Ng-Wong with Fixed Target Oscillation

the haploid which finds a solution within 20% of optimum in 60% of periods. However, if we look at periods in which the solution obtained was within 10% of optimum, (i.e. > 0.9), then we find that Haploid-Recover outperforms Extended-Additive, with success rates of 35% and 15% respectively. Both methods show a rapid response to the change in environment, where the GA rapidly improves the quality of the new, poorly fit, solutions that are produced as a result of the environment change. This suggests that sufficient diversity is created in the population as a result of the dominance change of recovery mutation to allow evolution to continue efficiently.

The Full-Ng-Wong GA behaves very differently however. Firstly, we notice a incremental improvement in the best fitness obtained for the lower, 2nd target. A clear "learning curve" is observed, until after 12 complete oscillatory periods the GA is able to maintain a constant value for this target immediately after the environment changes. Secondly, the GA quickly finds a good solution for the high target, and this solution is rapidly reverted to each time the target switches. Thirdly, after 2 periods, there is no decrease in fitness for the population when the target switches from the low target to the high target. Finally, best solutions achieved for both targets are poor when compared to the haploid-recover and additive-recovery GAs — 0.62 for the low target and 0.77 for the high target.

The performance of Full-Ng-Wong can be explained by examining the dominance mechanism. If no '10' or 'io' contentions exist, then a genotype can encode two arbitrary solutions, changing from one solution to another by merely applying the dominance change mechanism. Thus, it is possible to encode a genotype that represents the perfect solution to both targets, and flip between the two by inverting the dominance, without any requirement for further evolution. Thus the gradient shown in figure 8 is due to the population simply "learning" a sequence of dominance values that enables this rapid change to take place. Notice that this mechanism allows the "remembering" of only 2 solutions in the genotype, so this mechanism will not be useful in an environment where there are more than 2 possible situations, or, more generally, where environmental change results in a completely new fitness function, or target in this case. To confirm this and to investigate the ability of the other GAs to cope with such changes, we repeated the experiments using a random-oscillating knapsack problem.

5.3 Knapsack with Randomly Changing Targets

The 14-object knapsack problem was repeated, but this time a random new target was chosen at the end of each oscillation period of 1500 generations. Target values were confined to the range 0 to 16383. Figures 9, 10 and 11 illustrate the performance of the three GAs on this problem.

The results show that Full-Ng-Wong performs poorly compared to the other two methods. Maintaining a memory of the environment is not useful in the random target case, and any GA must rely on maintaining a sufficiently diverse population to be able to adapt to the changing conditions. The results imply that the dominance change mechanism in the Full-Ng-Wong case does not reintroduce diversity into the population, whereas the use of straightforward mutation can be extremely useful.

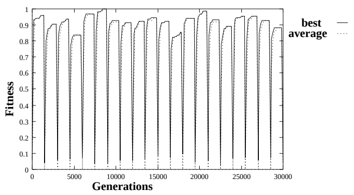

Fig. 9. Haploid-Recover with Random Target Oscillation

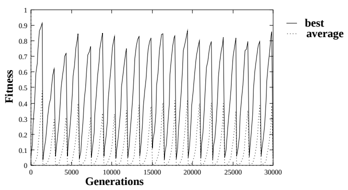

Fig. 10. Extended-Additive with Random Target Oscillation

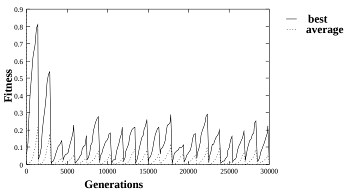

Fig. 11. Full-Ng-Wong with Random Target Oscillation

5.4 Analysis of Population Variance

In order to analyse the performance of each GA in more detail, we can look at the phenotypic variance in the population as each GA evolves, and for Full-Ng-Wong and Extended-Additive we can compare the phenotypic diversity to the genotypic diversity. Figures 12, 13 and 14 show the phenotypic gene-variance across the population at each locus in the phenotype plotted against generation for the fixed target experiments. For each type of GA, two graphs are plotted showing the variance (vertical axis) either side of the two target changes (low to high, generation 3000; high to low, generation 4500).

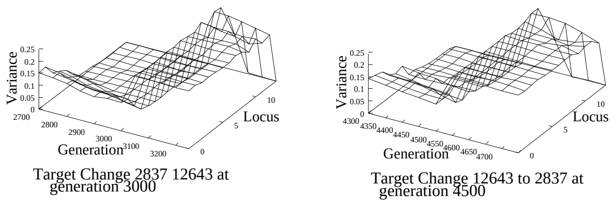

Fig. 12. Phenotypic Population Variance: Haploid-Recover

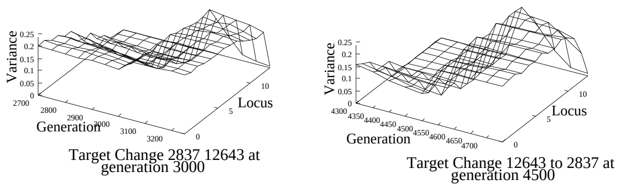

Fig. 13. Phenotypic Population Variance: Extended-Additive

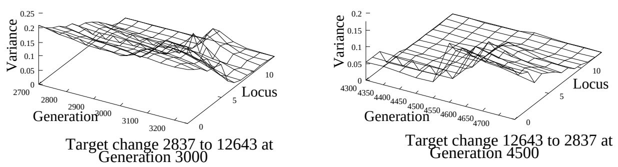

Fig. 14. Phenotypic Population Variance: Full-Ng-Wong

Figure 12 shows that Haploid-Recover has almost converged each time the target changes, but diversity is rapidly introduced due to the recovery mutation. Extended-Additive maintains slightly more phenotypic diversity in its population throughout the run than Haploid-Recover. This is unsurprising as a diploid GA would be expected to converge more slowly than a haploid. The effect of the dominance change is the same however. Full-Ng-Wong shows a slightly different picture. Just before the change from the low to high target, diversity in the population is high. However, the next time the target switches, phenotypic diversity is low across all loci and only a small increase is gained as a result of applying the dominance change mechanism. The effect becomes more pronounced as the number of target switches the population is exposed to increases.

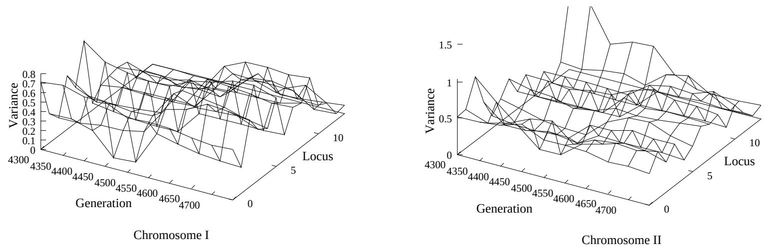

Fig. 15. Genotypic Population Variance for Extended-Additive

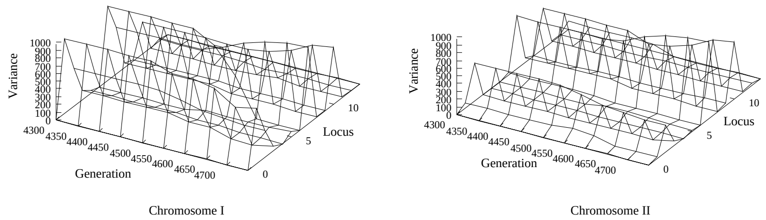

Fig. 16. Genotypic Population Variance for Full-Ng-Wong

We examine the genotypic diversity by plotting a similar graph for each of the two strings that make up the diploid genotype. Figures 15 and 16 show the *genotypic* diversity for Full-Ng-Wong and Extended-Additive either side of the 3rd target change, at generation 4500. For Extended-Additive, both parts of the genotype retain diversity, but the mapping from genotype to phenotype results in a less diverse phenotype than genotype. The genotypic diversity for Full-Ng-Wong shows a series of peaks running parallel to the generation axis, indicating some loci with very diverse genotypes and others that have completely converged. Closer examination reveals that those loci with little variance are exactly those loci in which the phenotype remains invariant from the optimal solution of target 1 to the optimal solution of target 2, hence even at generation 4500 the population is already starting to learn the two different solutions.

6 Conclusions

Using two variations of a non-stationary problem, we have shown that a simple diploid scheme does not perform well in either case. Adding some form of dominance change mechanism considerably improves matters, but the form of the change mechanism can have a significant effect. In the case of Full-Ng-Wong, the dominance change mechanism introduces a form of memory, which allows a population to "learn" two different solutions. Although this may be useful in certain situations, it cannot be used if there are more than two possible solutions, or if the environment changes do not follow a regular pattern.

For the problems considered, extending the additive dominance scheme with a change mechanism improves it considerably. It responds quickly to changes in the environment, even when the changes are random. However, there is little difference in performance between this GA and a simple haploid GA which undergoes heavy mutation when a decrease in fitness is observed between evaluations. Future experimentation with other non-stationary problems will make it possible to observe if these results can be generalised across this class of problems. If so, then the case for implementing a diploid mechanism as opposed to a simple mutation operator may be weakened, given that diploid schemes require more storage space and extra evaluations to decode genotype into phenotype.

References

1. Emma Collingwood, David Corne, and Peter Ross. Useful diversity via multiploidy. In *Proceedings of International Conference on Evolutionary Computing*, 1996.
2. David Corne, Emma Collingwood, and Peter Ross. Investigating multiploidy's niche. In *Proceedings of AISB Workshop on Evolutionary Computing*, 1996.
3. Jonathan Lewis. A comparative study of diploid and haploid binary genetic algorithms. Master's thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, 1997.
4. Khim Peow Ng and Kok Cheong Wong. A new diploid sceme and dominance change mechanism for non-stationary function optimisation. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, 1995.
5. Conor Ryan. The degree of oneness. In *Proceedings of the ECAI workshop on Genetic Algorithms*. Springer-Verlag, 1996.
6. Kukiko Yoshida and Nobue Adachi. A diploid genetic algorithm for preserving population diversity. In *Parallel Problem Solving from Nature: PPSN III*, pages 36–45. Springer Verlag, 1994.

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