Incremental Commitment in Genetic Algorithms

Richard A. Watson **Jordan B. Pollack**

Dynamical and Evolutionary Machine Organization
Brandeis University, Waltham, Massachusetts, USA
{richardw, pollack}@cs.brandeis.edu

Abstract

Successful recombination in the simple GA requires that interdependent genes be close to each other on the genome. Several methods have been proposed to reorder genes on the genome when the given ordering is unfavorable. The Messy GA (MGA) is one such 'moving-locus' scheme. However, gene reordering is only part of the Messy picture. The MGA uses another mechanism that is influential in enabling successful recombination. Specifically, the use of partial specification (or variable length genomes) allows the individuals themselves, rather than the ordering of genes within an individual, to represent which genes 'go together' during recombination. This paper examines this critical feature of the MGA and illustrates the impact that partial specification has on recombination. We formulate an Incremental Commitment GA that uses partially specified representations and recombination inspired by the MGA but separates these features from the moving-locus aspects and many of the other features of the existing algorithm.

1 Introduction

The Messy GA (MGA) [Goldberg et al 1989] has numerous features that distinguish it from the simple GA [Holland 1975]. Related research has built upon this feature set and developed it still further [e.g. Goldberg et al 1993, Harik 1997, Kargupta 1997]. In this paper we attempt to go back to basics - to simplify the feature set of the original MGA to see if we can better understand its essential elements. In particular, we are interested in how the MGA represents schemata to enable successful recombination. We use the original Messy GA [Goldberg 1989] as our point of reference throughout.

In the simple GA (henceforth, GA) a genome may contain some subset of interdependent genes that confer above average fitness - a fit schema. Hopefully, these will be adjacent to each other so that they will stay together during crossover but, more likely, they will be scattered along the genome as shown by the bold genes in the example given in Figure 1. The Messy GA uses a 'moving-locus' representation for genomes. Each gene is represented by a locus/allele pair. The bold genes from the GA representation are transferred to an example MGA representation in the second line of Figure 1. Moving the genes together in this way allows the MGA to transfer the schema intact during recombination.

0**1**010**0**1**1**	- GA
((2,1), (6,0), (8,1))	- MGA, moving-locus
-1---0-1	- fixed-locus, partial-commitment

Figure 1 representation of genomes / schema they contain for GA, MGA, and alternate method. See text.

However, we argue that the feature of the MGA that most fundamentally distinguishes its operation from that of the GA is *partial commitment*. That is, individuals in the MGA commit to specifying alleles on only a subset of the entire gene set. As our MGA example shows, not only have the genes of the schema been moved together, but some genes have been omitted - in this example, we have omitted those that are not part of the schema. The third line of Figure 1 introduces an alternate representation to illustrate the distinction between the moving-locus aspect and the partial commitment aspect of the MGA. Here the schema has been represented in a fixed-locus tertiary genome - each gene may be 0, 1, or unspecified ("-").

The following section reviews recombination in the simple and Messy GAs with emphasis on distinguishing the moving-locus aspect of the MGA from the effects of partial commitment. We re-cast MGA recombination as an operator that uses this partially-specified but fixed-locus representation. The discussion illustrates that partial-commitment has a profound effect on the operation of recombination and, regardless of moving-loci, can enable successful recombination.

In Section 3 we discuss the other features of the MGA that distinguish it from the GA and attempt to reduce them so we can focus on the effects of partial commitment. In particular, the MGA has a two-phase operation where the first phase uses genomes of limited commitment and the second phase allows full commitment. This two stage mechanism is a special case of what we will call an *Incremental Commitment* GA (ICGA). We implement an ICGA that utilizes partial commitment and recombination operations inspired by the MGA and applies them consistently throughout the operation of the algorithm - replacing the two-phase process with an integrated and incremental approach. The resultant algorithm is not without its own share of complications but it performs very well on a problem that is not solvable by the MGA. More importantly, in the process, we clarify some important concepts in the operation of the MGA and GA recombination in general.

Experimental results, Section 4, utilize a hierarchical building-block problem, shuffled H-IFF [Watson et al 1998], that exhibits strong interdependency between genes and random genetic linkage. We demonstrate that the ICGA successfully discovers and recombines building-blocks through all levels of the problem. Finally, we discuss the limitations of the ICGA and relate the concepts illustrated here to the abilities of alternate algorithms.

2 Recombination and partial commitment

Genetic Linkage. The performance of the simple GA is dependent on the correspondence between *genetic linkage* and *epistatic linkage*. Genetic linkage refers to the proximity of genes on the genome and their corresponding tendency to travel together during crossover. Epistatic linkage refers to the interdependency of gene expression (without regard for gene position) and this defines important schemata in a problem.[1] Genetic linkage is said to be *tight* when genes are close to each other on the genome and tight genetic linkage between the genes of a schema is a requirement for the growth (increased representation in the population) of that schema in the simple GA. As the defining length of a schema increases, the probability that crossover will transfer the entire schema to the offspring decreases [Holland 1975].

It is not reasonable to assume that the given ordering (and certainly not a random ordering) of parameters for a problem will always reflect the interdependency of those variables. One way to rectify the problem of poor linkage is to re-order the genes so that interdependent genes are close together - so that genetic linkage and epistatic linkage correspond. Accordingly, several variants of the GA have proposed moving-locus schemes, e.g. GA with inversion operator [Franz 1972], Messy GAs [Goldberg 1989], Linkage Learning GA [Harik 1997].

Despite the validity of these approaches, it is important to remember that the need to rearrange genes is a result of using a recombination operator that is sensitive to gene positions. Let us reflect on the broader view of recombination. Recombination is supposed to take some good genes from one individual and some good genes from a second individual and make an offspring with the good parts put together. The problem is determining, for each donor individual, which part is a good part to take. Crossover uses the heuristic of adjacency on the genome to determine which genes will stay together and so it is dependent on tight linkage. The genetic operators in the Messy GA (and its variants) also use bit adjacency for representing epistatic linkage - though, importantly, they do not use the original ordering of genes.[2] But this is not the only mechanism at work in the MGA. Because the MGA uses partially specified genomes - individuals code for only a subset of the possible genes - the individuals themselves, rather than the position of genes within the individuals, can represent subsets of epistatically linked genes. Disregarding the moving-locus aspects of the algorithm, underspecification or partial-commitment makes the nature of recombination in the Messy GA radically different from the nature of recombination in the simple GA. In fact, we shall see that the moving-locus aspects of the algorithm are not essential to successful recombination even in problems of random linkage.

Resolving Conflicts. Let us look at recombination from a different perspective: specifically, the appropriate resolution of *conflicts*. In the simple GA each gene of one parent will either agree or be in conflict with the corresponding gene of the other parent. Where the parents agree recombination is not problematic. See Figure 2. Incidentally, this observation has led some researchers to characterize recombination as (nothing more than) a 'similarity preserving' operator, and further to conclude that uniform crossover is an appropriate mechanism for recombination. This point of view is valid in the simple GA but our work here exemplifies that this view does not always hold, and that the more profound role of recombination described by the Building-Block Hypothesis [Holland 1975] is valuable.

```
A: 01010011
B: 10000110
-----------
C: ??0?0?1?
```

Figure 2. Successful recombination must determine how to resolve the conflicts ("?") in allele values supplied by two parents (A and B) to create the offspring, C.

Taking the conflict resolution perspective, the task of recombination is to resolve allele disagreement in an appropriate manner. Crossover uses genetic linkage as a heuristic for resolving conflicts in a self-consistent manner - informally, 'if we take this bit we should also take the bits next to it'. But clearly, when the placement of related genes on the genome is unfavorable this heuristic is not valid. See Figure 3.

A: 0**1**010**0**1**1**
B: **1**0**0**0**0**110
C: **110**1**00**1**1**

Figure 3. Parents, A and B, each contain a useful subset of genes (three genes per parent, shown in bold). The desired offspring, C, should take the good genes from both parents as shown. But simple crossover can not achieve this.

Without additional information about epistatic linkage, only uniform crossover could possibly create the offspring

[1] the unqualified term linkage shall refer to *genetic* linkage

[2] the representation used by MGAs is sometimes referred to as *position-independent* coding - however, this is a little misleading - although the coding is not dependent on the original ordering of genes (or variables in the fitness function), the recombination operators *are* sensitive to the position of genes on the *re*ordered genome.

shown from these two donors. But uniform crossover is equivalent to assigning random allele values to all conflicts (to each question mark in Figure 2). Uniform crossover only has an advantage over random guessing where genes are already in agreement. This defeats the intent of recombination in the Building-Block Hypothesis.

Moving-Locus / Partial Commitment. The Messy GA response is to use a moving-locus representation. Each gene is represented as a locus/allele pair - and an individual consists of a variable number of these pairs. The two parents from Figure 3 might be represented as below[3]:

```
A: ((2,1), (6,0), (8,1))
B: ((1,1), (3,0), (5,0))
-----------------------------------
C: ((2,1), (6,0), (8,1), (1,1), (3,0) (5,0))
```

Figure 4: 'Splice' recombination in the Messy GA.

It is then a simple matter to create an offspring C which is the sum of good genes from both parents. This is the *splice* operation of the Messy GA [Goldberg 1989]. But note that the moving-locus aspect of this operation, i.e. the fact that these genes have been brought next to each other on the genome, is beside the point. The splice operation shown is *linkage-invariant* - the position of genes within the parents is not relevant. The important aspect of recombination in this example is the fact that only the desirable genes are specified in the parents. Accordingly, we may look at the problem described in Figures 2 and 3, as not so much a problem of there being *distance* between the desired genes, but rather as the problem that there are *garbage genes* between the desired genes. In the MGA individuals can specify for a good schema without having to specify all the remaining genes. Thus recombination can combine the good schemata from two individuals without 'garbage' genes causing conflicts with desirable genes.

Such garbage bits have variously been referred to as 'hitch-hikers' [Mitchell and Forrest 1993], or 'free-loaders' - the idea being that they 'catch a ride' on the fitness of the good genes in the same string without contributing fitness themselves. Goldberg et al voice a concern for the same principle when found in the Messy GA where they refer to them as 'parasitic bits'. In the case where it is not known exactly how many genes a good schema should have an over-estimate leaves room for garbage bits. "During [recombination] these parasites will tend to prevent the expression of other more useful bit combinations" [Goldberg et al 1989]. We shall return to the question of how we could ever know just how many bits a schema should specify. But for now, we shall grant there is some way of knowing, at least an upper limit, and we take a moment to emphasize that it is this knowledge that has the most profound affect on recombination in the MGA.

Figure 4 showed the desirable genes of two parents in the MGA representation. Figure 5 shows that the desired genes might just as well be spread out along the genome so long as we can identify them.

```
A: -1---0-1
B: 1-0-0---
-----------
C: 110-00-1
```

Figure 5: An alternate representation of the splice operation in Figure 4. Here we represent unspecified genes, or *don't cares,* by "-" and the offspring is created by taking specified genes from either parent where available.[4]

Thus far it has been taken as a given that if an algorithm has a mechanism that is *capable* of expressing something advantageous then selection will see that it does. For example, in the MGA, a moving-locus scheme plus selection is assumed to reorder genes in an advantageous manner. Similarly, in our examples of partial commitment, we have selected out the desirable genes and discarded the garbage genes. Later we shall show how we need to engineer selection to achieve this in practice.

To recap, the significant feature of Figure 4 is that a subset of epistatically linked genes is represented by inclusion in an individual rather than by their adjacency on the genome. The point we wish to stress is that successful recombination in Figure 4 is not a result of moving-locus representation.

Dominant splice. We now address the case where the parents use partial specification, as before, but here there are conflicts in the genes that they specify. Figure 6 shows two parents that each code for a subset of bits. Unlike Figure 5 (and Figure 4), these parents specify for non-disjoint sets of genes and, at some of the common loci, allele values conflict.

```
A: -1--00-1
B: 100-0--0
-----------
C: 1?0-00-?
```

Figure 6: Partial specification with conflicts.

Although much reduced, we still have a problem of conflict resolution like the one that we started with in Figure 3. It is for these occurrences that another feature of the MGA is introduced. In the MGA, when gene values are over-specified an "intrastring precedence rule" is invoked: e.g. a first-come-first-served rule is proposed which expresses only the first occurrence of each gene encountered in a left to right scan.

A: ((2,1), (5,0), (6,0), (8,1))
B: ((1,1), (2,0), (3,0), (5,0), (8,0))

C: ((2,1), (5,0), (6,0), (8,1), (1,1), **(2,0)**, (3,0), **(5,0)**, **(8,0)**)

Figure 7: A first-come-first-served rule determines which over-specified genes are ignored (shown bold).

[3] There are other ways to represent the same individuals with messy coding but the details are not relevant to our discussion.

[4] We shall shortly address what should be done in the case that specified genes conflict.

Notice that the expression of genes is only dependent on *which* parent donated the genes. So far, we have consistently allowed the genes of string A to be transferred to the offspring first, to be followed by the genes of B. If the choice is reversed the resultant string has a different expression - specifically, all conflicts would be resolved in B's favor instead of in the favor of A as shown in the example. But, either way, the position of genes *within* each parent is still not relevant. To this extent the splice event shown here remains linkage-invariant and we can still express the splice operation using our fixed-locus representation - Figure 8.

```
A: -1--00-1
B: 100-0--0
___________
C: 110-00-1
```

Figure 8: Partial specification with conflicts resolved by allowing parent A to dominate. Together with the specification that parent A is the more fit, we call this recombination operator 'dominant splice'.

Adding the specification that parent A shall be the more fit of the two parents, we call this operation "dominant splice": it is the recombination operator that we shall use in the following experiments. Notice that simply allowing conflicts to be resolved entirely in the favor of one parent does not make sense in a fully-specified representation. We would simply reproduce the first parent. Partially committed strings allow the good genes from one string to be transferred intact, and allow additional schemata to be supplied by the second parent where they do not conflict with the first. This operation is entirely insensitive to genetic linkage, yet resolves conflicts in a consistent manner. The purpose of the Incremental Commitment GA (ICGA), presented in the Section 3, is to separate this form of recombination from the other features of the MGA.

Cut. It is important to clarify that we do not use the 'cut' operator found in the MGA. Cut takes a subset of genes from each parent in a manner similar to one-point crossover. The division of genes is thus sensitive to the adjacency of genes on the (reordered) genome. In so doing the cut operation destroys the linkage-invariance quality of recombination in the MGA. Without cut, the moving-locus aspects of recombination in the MGA are irrelevant. This is not to say that they are not valuable - but we will demonstrate that they are not essential. By removing cut we can separate the moving-locus aspects from the partial-commitment feature found in the MGA.

3 Tidying-up the Messy GA

Goldberg et al [1989 pp.508-509] provide the following list of MGA features:

1) *Use of variable-length strings that may be under- or overspecified.*
2) *Use of intrastring precedence rules (first-come-first-served or other)*
3) *Use of cut and splice instead of crossover.*
4) *Division of evolutionary processing into two phases: primordial and juxtapositional.*
5) *Use of variable-size population* [appropriate to phase].
6) *Use of partially enumerative initialization to encompass the longest misleading non-linearity in the problem.*

There are a few other features besides, not included in this list:

7) Use of competitive templates to fill-in for unspecified genes for the purposes of evaluation.
8) Use of "thresholding"; a diversity maintenance method to prevent premature convergence.

Space does not permit us to detail all of these features - suffice to indicate that it is difficult to determine which features of the MGA are critical to its operation. The intent of the ICGA is to reduce this feature set so we can illustrate one aspect of the algorithm. Specifically, we wish to determine whether the use of partial commitment is sufficient for successful recombination. We shall be describing the Incremental Commitment GA used in the following experiments in detail, but in overview: the ICGA will incorporate underspecification, splice and some form of diversity maintenance mechanism and, in addition, we introduce a size penalty into the fitness function which provides a mechanism to regulate the growth of strings incrementally and replaces the two-phase aspect of the MGA (hence, incremental commitment). Other features of the MGA are not used.

We now recap the genetic operators we shall employ and describe the mechanism we use to maintain diversity. Incremental commitment will be detailed in Section 5.

Representation and recombination. As mentioned earlier, we shall not use the moving-locus representation of the Messy GA but we shall keep the underspecification and splice recombination. We shall also dispense with overspecification and, as mentioned, when conflicts occur between two parents, the more fit parent will be favored for all conflicts in that mating, and conflicting genes from the less fit parent will be discarded permanently. Accordingly we shall simply use a fixed length ternary genome - {0,1, and null} (see Figure 8). The resultant offspring may then optionally be mutated. Thus overspecification, cut, and intrastring precedence rules are eliminated.

Note that the use of underspecification requires us to evaluate partial strings and accordingly, in the general case, we shall need to use some mechanism to replace competitive templates. However, the test problem used in the following experiments naturally evaluates partial strings and so a competitive template will not be necessary.

Diversity maintenance. The diversity maintenance mechanism we employ is not the same as that used in the MGA. We use a resource-based fitness-sharing mechanism as used in our earlier work [Watson et al 1998]. This mechanism uses considerable domain knowledge - specifically, a resource-level is maintained for every building-block in the problem - so it is not intended as a general solution to diversity maintenance. However, it enables us to focus on the operation of partial commitment in the algorithm without significant concern of premature convergence that would defeat effective recombination. Our previous work has shown that the fitness sharing method does not make our test problem easy - the simple GA cannot solve the shuffled H-IFF (that we will detail shortly) with, or without, this fitness sharing method. We are not as yet able to verify whether the original mechanism for diversity maintenance detailed by Goldberg et al [1989] will provide the same support for the ICGA.

This leaves the features of the MGA numbered 4 through 6 which we replace with a more general mechanism to limit the length of strings. But, in order to properly explain our motives for this change we shall first explore the motives behind the two-phase operation of the MGA: this requires discussion of the problem class to which these algorithms are applied.

4 A Test Problem

Concatenated Trap Functions. Evaluation of Messy GAs has usually been performed using *concatenated trap functions* [Goldberg et al 1998, Deb & Goldberg 1992, Goldberg et al 1993, Harik 1997, Kargupta 1997]. These are simple concatenations of difficult (often fully-deceptive) fixed-size subfunctions. These problems are therefore delineable at some order, k, being the size of the subfunctions. The rationale for this and the design of the Messy GA are mutually consistent. For example, if a problem is order-k delineable and the gradient information below order-k is entirely misleading then it is appropriate to use partially enumerative initialization. "In partially enumerative initialization, at least one copy of all possible building-blocks of a specified size is provided, where the size is chosen to encompass the highest order deceptive nonlinearity suspected in the subject problem" [Goldberg et al 1989]. In this manner, the gradient information below order-k is ignored - initialization simply produces all possibilities randomly, and subsequent selection picks out the good guesses (the primordial phase).

Similarly, if the gradient information above order-k is reliable ("many high-order nonlinearities in problems encountered in practice are weak" [Goldberg et al 1989 p.505]) or perfect, as in the case of the test functions, then it is appropriate to allow accumulation of genes to occur unchecked after the population has been properly pruned (as per the juxtapositional phase).

These two phases of the MGA are the mechanism by which partial commitment is regulated. The primordial phase searches for highly-fit length-k individuals where k is the size of the trap function used in the test problem. Several iterations of selection are performed before their lengths are allowed to increase by recombination in the juxtapositional phase.

Clearly, these mechanisms are appropriate for the assumptions that are made of the problem. In large part, these assumptions are made because it is not clear how a problem could be solvable at all if deception were not bounded, and how it could be at all hard if it were not deceptive to some level. However, previous work [Watson et al 1998, Watson & Pollack 1999] introduces a test problem which is consistent in the difficulty it creates at all scales of resolution.

Hierarchical if-and-only-if. The canonical form of the problem defined in [Watson et al 1998] is called Hierarchical if-and-only-if (H-IFF). In H-IFF the fitness of a string, B, is defined as follows:

$$F(B)=\begin{cases} 1, & \text{if } |B|=1, \text{ and } (b_1=0 \text{ or } b_1=1) \\ |B| + F(B_L) + F(B_R), & \text{if } (|B|>1) \text{ and } (\forall i\{b_i=0\} \text{ or } \forall i\{b_i=1\}) \\ F(B_L) + F(B_R), & \text{otherwise.} \end{cases} \quad \textbf{Eq.1}$$

where B is a block of bits, $\{b_1,b_2,...b_k\}$, $|B|$ is the size of the block=k, b_i is the ith element of B, B_L and B_R are the left and right halves of B (i.e. $B_L=\{b_1,...b_{k/2}\}$, $B_R=\{b_{k/2+1},...b_k\}$). The length of the string evaluated must equal 2^p where p is an integer (the number of hierarchical levels). Notice that this function gives no reward to nulls and therefore naturally evaluates partially specified strings.

This function interprets the string as a binary tree and recursively decomposes the string into left and right halves. At every level a string is rewarded if all its bits are identical - either all ones or all zeros. From a bottom-up point of view, we can see H-IFF as defining a hierarchy of building-blocks which may be assembled together in pairs, doubling in size and reward at each level. Since both ones and zeros are rewarded each partition contains two equal-fitness competing schemata. Both are rewarded and both contain one of the two global optima (at all ones and all zeros) - yet the problem still contains local optima since only compatible blocks are rewarded at the next level. For example, the best fitness strings besides the two global optima are N/2 ones followed by N/2 zeros, or vice versa. Both these local optima are maximally distinct from the global optima in Hamming space and make H-IFF very hard for any kind of hill-climber or non-population-based method [Watson et al 1998].

Unlike the concatenated trap functions, the recursive subfunction used to build H-IFF creates a problem with building-blocks that are not separable. The hierarchically consistent structure, [Watson & Pollack 1999], requires the searching of bit combinations at the bottom-level, and the searching of schema combinations of successively higher order, at all subsequent levels. Thus H-IFF is not order-k delineable for any k (except the trivial case where k=N, the number of bits in the problem). That is, it does not consist of separable subfunctions. Incidentally, this means that, unlike the concatenated trap functions, H-IFF cannot be solved by a macro-mutation hill-climber even when linkage is tight, [Jones 1995, Watson et al 1998]. Yet given that linkage is good it is easy for a GA to solve [Watson et al 1998].

Our goal now is to solve the *shuffled* H-IFF where linkage is random. The shuffled-H-IFF uses a random but fixed reordering of the variables from standard H-IFF so that epistatically linked genes are scattered along the genome (in just the same manner as is used for the concatenated trap functions in previous MGA research). The shuffled H-IFF cannot be solved by the GA because genetic linkage and epistatic linkage do not correspond, and it cannot be solved by the MGA because it is not delineable into separate subfunctions.

5 Incremental Commitment

Goldberg et al state, "Messy GAs that use tightly linked building-blocks are analogous to simple GAs processing alleles on easy problems" [1993] - but this is not true in a hierarchically consistent problem like H-IFF. In fact, the class of problem facing an algorithm after identifying the linkage of order-k building-blocks is exactly the same class as it was after discovering the linkage of order-k/2 building-blocks (by definition, [Watson et al 1999]). Since H-IFF (shuffled or otherwise) is not delineable it is clear that the two-phase method of the MGA cannot be applied. It is not sufficient to use partially enumerative initialization to find building-blocks for any particular order. Nor is it appropriate to perform combination only in a separate phase (i.e. the juxtapositional phase). Accordingly, we shall dispense with the distinct phases of the Messy GA and return to the seamless integration of recombination and selection found in the simple GA.[5]

The Persistent Parasites. If we are to do away with a separate first phase that conveniently limits the length of strings we need some other mechanism to prevent the otherwise unchecked growth of the number of specified genes per individual. Without a restriction, recombination will accumulate genes at an exponential rate, and selection will promote larger sub-optimal composites rather than small-but-perfect building-blocks. This means that the string will predominantly consist of garbage bits that prevent successful recombination as they do in the GA in Figure 3.

When Goldberg et al refer to the issue of parasites, mentioned earlier, they suggest the use of "nulls" to fill up the non-essential gene values, and in our terminology, postpone commitment for these genes. They also suggest using the number of nulls to resolve ties between strings of equal fitness: i.e. for any given fitness value, the string with the most nulls is preferred. However, Goldberg et al do not take this idea to its natural conclusion. By augmenting the fitness function with a size-penalty we can use this idea as the sole method of regulating string growth. The size-penalty will be proportionate to the number of specified genes in the individual. See Equation 2.

[5] An 'outer loop' that repeats the process of selection and juxtaposition has been proposed for the Messy GA but apparently this still assumes a fixed ceiling on the order of deception.

Comparing Two-Phase with Incremental Commitment. Both the MGA two-phase method and the incremental commitment (IC) method require domain knowledge to regulate the growth of strings. In the MGA a researcher must know the order of the highest-order deceptive non-linearity in the problem. This is used to inhibit commitment in the primordial phase. In the IC method a researcher must use knowledge of how the fitness of strings grows with their size. The MGA method is not applicable to hierarchically consistent problems where there is no limit to deceptive non-linearities. In this respect the IC method is the more general.

Nevertheless, the applicability of the IC method is dependent on finding an appropriate balance between size and fitness. In our current implementation this is complicated by the choice of fitness sharing method which depresses the value of previously discovered building-blocks. To adjust for this the fitness contribution of building-blocks is first exponentiated and then re-scaled after fitness contributions are summed to bring the result back into line with the size of the individual. This has the effect of exaggerating the importance of the largest building-block discovered. Specifically, in our ICGA implementation, the fitness of a string B, is given by

$$F(B)= \log_2(F'(B))-\text{size}(B) \text{ where,}$$

$$F'(B)=\begin{cases} 1, & \text{if } |B|=1, \text{ and } (b_1=0 \text{ or } b_1=1) \\ 2^{|B|} + F'(B_L) + F'(B_R), & \text{if } (|B|>1) \text{ and } (\forall i\{b_i=0\} \text{ or } \forall i\{b_i=1\}) \\ F'(B_L) + F'(B_R), & \text{otherwise.} \end{cases} \quad \textbf{Eq. 2}$$

The meaning of variables is as per Equation 1, and size(B) is the number of non-null genes in the string.

We acknowledge this is not a general method but, in a similar spirit to the fitness-sharing method, it enables us to focus on the purpose of the experiment, specifically the investigation of partial commitment, without the complication of the two-phase features of the MGA.

6 Experimental Results

The following experiments apply the ICGA to the 64-bit shuffled H-IFF problem. In summary, our implementation of the ICGA uses the following features:

- Partially specified representation of genes (0, 1, null) (see Figure 8)
- Dominant splice recombination operator (see Figure 8)
- Diversity maintenance (resource-based fitness-sharing)
- Fitness function is augmented with size-penalty (Equation 2)

In all other respects the ICGA is as per a simple GA with the following parameters: a population of 1000 individuals, mutation with bit-wise probability of 2/64 of assigning a new random value (0,1,null), recombination (dominant splice) applied with probability 0.3, generational exponentially-scaled rank-based selection (with scaling factor, p=0.01), 50% elitism (i.e. the best 50% of the population are copied to the new population without genetic variation). Individuals are initialized to all-null strings (i.e. they have no specified genes).

Figure 9 shows the performance of the ICGA on 64-bit shuffled H-IFF as indicated by the best individual in the population in each generation.

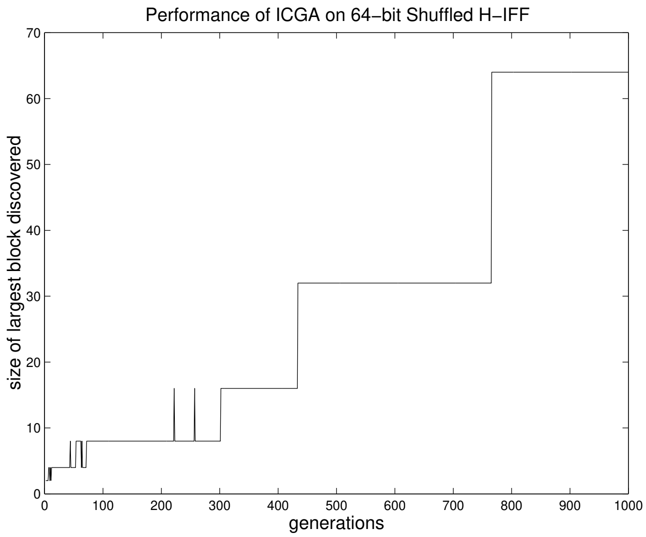

Figure 9: Performance of ICGA on shuffled H-IFF.

Rather than show performance as fitness of best individual we use the size of the largest correct block discovered. This shows clearly how the ICGA discovers the building-blocks over successive levels as they double in size. Despite the rather high elitism used there is still a little 'forgetting' occurring in the early stages of search. Nonetheless, the ICGA successfully finds a complete 64-bit solution before 1000 generations. Actually, it finds both the all-ones and the all-zeros global optima. (not shown).

The curves in Figure 10 indicate the proportion of building-blocks of each size discovered - e.g., discovery of size-16 blocks begins between 200 and 300 generations and by approximately 450 generations about half of the size-16 all-one blocks have been discovered. There are N/size all-one blocks of each size.

We stress that the shuffled H-IFF problem used in these experiments is more difficult than the concatenated trap functions usually used to test MGAs in that the building-blocks are not separable. H-IFF uses strong non-linear interdependency between competing building-blocks at all hierarchical levels. And shuffled H-IFF has random genetic linkage. The results shown therefore demonstrate that, with appropriate regulation, incremental commitment is sufficient for solving multi-level building-block problems with strong epistatic linkage and random genetic linkage. And in particular, the results demonstrate that moving-locus aspects of the MGA are not required since the ICGA uses linkage-invariant recombination. Accordingly, the ICGA performs identically regardless of whether the genes are shuffled or not.

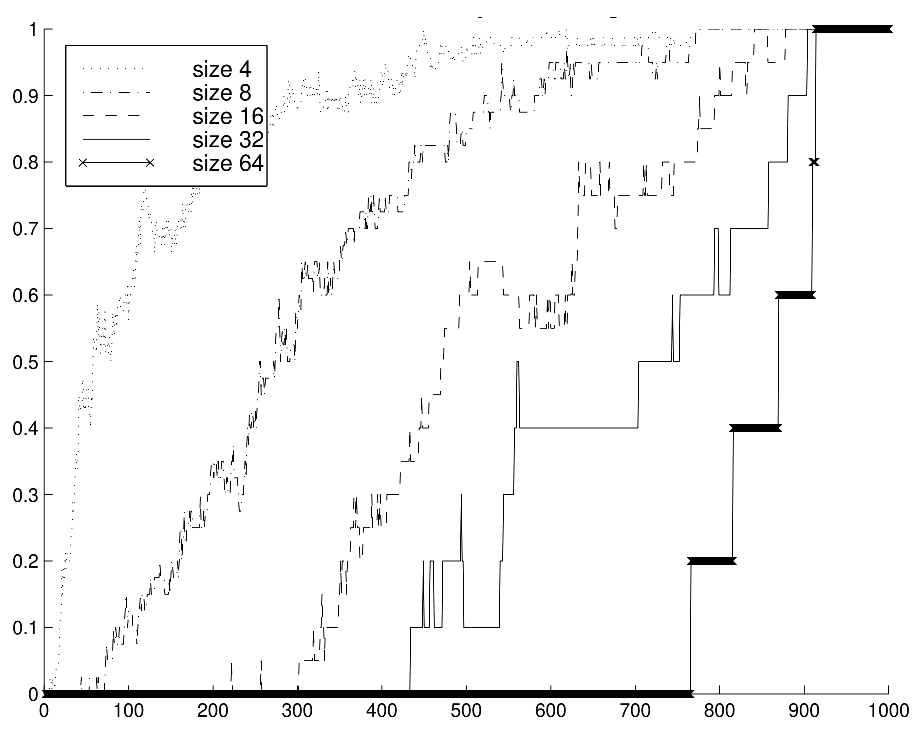

Figure 10: The discovery of (all-one) building-blocks at intermediate levels (blocks of size 4,8,16,32 and 64) averaged over 5 runs of the ICGA: vertical axis shows the proportion of the number of all-one blocks of that size.

7 Conclusions

"From square one, we would like to be able to evaluate the fitness of a part without possessing a whole." [Goldberg et al 1989]. Evaluating parts, together with an appropriate mechanism to (re)combine them, is the heart of the MGA. The virtue of this intuitive concept has been difficult to assess amongst the numerous other features of the MGA.

The Incremental Commitment GA has attempted to extricate the feature of partial specification from the other features of the MGA. It demonstrates the use of a fixed-locus partially-specified genome instead of a moving-locus scheme. And incremental commitment using a size-penalty augmentation to the fitness function instead of the two-phase method of the MGA.

Our experiments have addressed a hierarchical building-block problem, shuffled H-IFF [Watson et al 1998]. This problem has random linkage and therefore cannot be solved by the simple GA [Watson et al 1998]. It is also not delineable into separable subfunctions of a fixed order and therefore the Messy GA cannot be applied. But, with the proviso that diversity in the population is maintained, and an appropriate balance between fitness and the growth of strings is found, the ICGA solves the shuffled H-IFF successfully. It is incremental commitment, inspired by the MGA, that enables the ICGA to discover and recombine the schemata successfully. Thus we have shown that a key feature of the MGA, the use of under-specification, provides the basis for a successful algorithm for this class of problem.

Unlike the MGA the ICGA does not reorder genes on the genome so as to provide correspondence between genetic linkage and epistatic linkage. The ICGA it is not sensitive to the order or proximity of genes on the genome – its mechanisms are linkage-invariant. Instead the ICGA uses the individuals themselves, rather than the adjacency of genes within an individual, to represent sets of epistatically linked genes. Partially-committed genomes enable individuals to specify a schema without having to specify for all the remaining genes. And, assuming selection does its job – finding fit schema and discarding garbage genes – this enables recombination to combine good schemata from two individuals without garbage genes causing conflicts with desirable genes.

Thus we have illustrated that the recombination of partially-specified strings need not suffer from linkage-dependent operators like crossover or cut, and to this extent, we have demonstrated that the moving-locus aspects of the MGA are subsidiary. Nevertheless, linkage learning algorithms, and the original MGA, are a valid alternative to this approach.

The difficulty of balancing fitness with size, as required in the ICGA, may prove difficult to use in practice. In the ICGA a gene is either committed fully or not at all, and the number of committed genes is regulated by the string's fitness. A method that is more subtle in deciding a *degree* of commitment to gene values may be more robust. The Selective Crossover algorithm [Vekaria & Clack 1998] offers such an approach. We would also like to investigate a 'group evaluation' method that uses other members of the current population to temporarily fill-in for unspecified genes during evaluation [Watson & Pollack 1999b] and act as the competitive template for problems where partial evaluation is not provided naturally.

Acknowledgments

We are grateful to the members of DEMO at Brandeis, especially Sevan Ficici, and also to David Goldberg and Kanta Vekaria for valuable discussion and insight.

References

Altenberg, L, 1995 "The Schema Theorem and Price's Theorem", 3, eds. Whitley & Vose, pp 23-49, Morgan Kaufmann, San Francisco.

Deb, K & Goldberg, DE, 1992, "Analyzing Deception in Trap Functions", In FOGA 2, ed. Darrell Whitley, Vail, CO.

Forrest, S & Mitchell, M, 1993 "Relative Building-block fitness and the Building-block Hypothesis", in FOGA 2, Morgan Kaufmann, San Mateo, CA.

Goldberg, DE, 1989 "Genetic Algorithms in Search, Optimisation and Machine Learning", Reading Massachusetts, Addison-Wesley.

Goldberg, DE, Deb, K, Kargupta, H & Harik, G, 1993 "Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms", IlliGAL Report No. 93004, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, Il.

Goldberg, DE, Korb, B, & Deb K, 1989 "Messy Genetic Algorithms: Motivation, Analysis and First Results" Complex Systems 1989, 3, 493-530.

Harik, GR, & Goldberg, DE, 1996, "Learning Linkage" in FOGA 4, Morgan Kaufmann, San Mateo, CA.

Harik, GR, 1997 "Learning Gene Linkage to Efficiently Solve Problems of Bounded Difficulty Using Genetic Algorithms" PhD dissertation, Dept. of Comp. Science and Engineering, Uni. of Michigan.

Holland, JH, 1975 "Adaptation in Natural and Artificial Systems", Ann Arbor, MI: The Uni. of Michigan Press.

Jones, T, 1995, Evolutionary Algorithms, Fitness Landscapes and Search, PhD dissertation, 95-05-048, University of New Mexico, Albuquerque. pp. 62-65.

Kargupta, H, 1997, "Gene Expression: The Missing Link In Evolutionary Computation" Genetic Algorithms in Engineering and Computer Science, Eds. Quagliarella, Q Periaux, J, and Winter, G.: John Wiley and Sons.

Vekaria, K, & Clack, C, 1998, "Selective Crossover in Genetic Algorithms", Procs. of PPSN V, Eds. Eiben, Back, Schoenauer, Schweffel: Springer. pp438-447.

Watson, RA, Hornby, GS, & Pollack, JB, 1998, "Modeling Building-Block Interdependency", Procs. of Parallel Problem Solving from Nature V, Eds. Eiben, Back, Schoenauer, Schweffel: Springer. pp97-106.

Watson, RA, & Pollack, JB, 1999, "Hierarchically-Consistent GA Test Problems", Procs. of 1999 Congress on Evolutionary Computation, to appear.

Watson, RA, & Pollack, JB, 1999b "How Symbiosis Can Guide Evolution", 5th European Conference on Artificial Life, submitted.