Acknowledgements

This research was supported by the National Science Foundation IRI-9224917.

References

C. Bridges & D. Goldberg (1987), "An analysis of reproduction and crossover in a binary-coded genetic algorithm", *Proc. 2nd int. conf. Genetic Algorithms*, J. Grefenstette Ed., Lawrence Erlbaum Associates, 1987

T. Davis (1991), "Toward An Extrapolation Of The Simulated Annealing Convergence Theory Onto The Simple Genetic Algorithm", Dissertation presented to the University of Florida.

A. Nix & M. D. Vose (1992), "Modeling Genetic Algorithms With Markov Chains", *Annals of Mathematics and Artificial Intelligence* 5, 79-88.

M. D. Vose (1990), "Formalizing Genetic Algorithms", *Proc. IEEE wksp. on G.A.s, N.N.s, & S.A. applied to problems in Signal & Image Processing*, May 1990, Glasgow, U.K.

M. D. Vose (1993), "Modeling Simple Genetic Algorithms", *Foundations of Genetic Algorithms - 2*, D. Whitley Ed., Morgan Kaufmann, 1993

M. D. Vose (1993), "A Closer Look At Mutation In Genetic Algorithms", To appear in *Annals of Mathematics and Artificial Intelligence.*

D. Whitley (1992), "An Executable Model of a Simple Genetic Algorithm", *Foundations Of Genetic Algorithms - 2*, D. Whitley Ed., Morgan Kaufmann.

Thinking about schema competitions is generally a waste of time when trying to explain mutation effects like these. On the other hand, explanation in terms of fixed points of $\mathcal{G}$ is not difficult. Although some results concerning fixed points and their relevance to genetic search have been obtained (Vose, 1993), much remains to be accomplished.

Returning to the question "why schema?", the explanation given in a section above was the historical one. That tradition may well be unconvincing. Schema bear no special relationship to either selection or mutation, and these latter operators effect genetic search profoundly.

The main argument against schema is that given $\mathcal{G}$, they are unnecessary; they do not simplify analysis, and the effort of mapping a problem concerning populations into schema and back out again is wasted. Moreover, in the absence of the information that $\mathcal{G}$ provides, it is unclear that schema analysis could have significant logical consequences given the severe limitations of the classical schema theorem.

Summary

The classical schema theorem is not sufficiently powerful to make predictions concerning the direction of genetic search. It says nothing about if or how a schema might be sampled at strings which are not in the current population. In particular, it can have no logical consequence which would be stronger than a conclusion based on zero mutation and zero crossover rates. In other words, the schema theorem is only a statement concerning how selection focuses the initial population. The occurrence of crossover and mutation rates in its conclusion serve only to weaken the resulting inequality.

If this were not enough (though it surely is), the type of information provided by the classical schema theorem – an inequality – is often inadequate to make decisions even if subsets of the initial population were the only ones of interest.

To the extent that the GA community has appealed to the schema theorem to justify conclusions involving populations containing strings not present in the original population, the community has been in error.

The GA community should not be appealing to the schema theorem to justify conclusions concerning the direction of Genetic Search.

Fortunately, there are extensions of the classical schema theorem, referred to generically as $\mathcal{G}$, which remedy the classical schema theorem's inability to look beyond the initial population. In fact, creating the next generation by sampling from the distribution which $\mathcal{G}$ provides is indistinguishable from the stochastic transition taken by a genetic algorithm.

Since $\mathcal{G}$ deals with populations directly, schema are unnecessary when using it. If not used, it is unclear that schema analysis could have significant logical consequences given the severe limitations of the classical schema theorem.

Since the function $\mathcal{G}$ encodes all relevant information – anything which ever could be proved about the GA must correspond to a property of $\mathcal{G}$ – perhaps schema are moving towards historical artifacts.

To the extent that the GA community has appealed to the schema theorem in order to justify conclusions involving populations containing strings not present in the original population, the community has been in error.

The GA community should not be appealing to the schema theorem to justify conclusions concerning the direction of Genetic Search.

The reader will please appreciate that *claims* or *conclusions* based on the schema theorem are not being called into question. What is being disputed is whether the schema theorem would *justify* claims which are made and conclusions which are reached concerning the direction of Genetic Search.

GA Variations

There have come to be endless variations of the simple genetic algorithm. For most of these, the function $\mathcal{G}$ does not apply. The situation is analogous for the schema theorem; the version of it which is appropriate for ranking and uniform crossover is not appropriate for two point crossover with proportional selection.

Just as the schema theorem needs amending appropriate for alternate GA forms, the same is true of $\mathcal{G}$. In the case of the schema theorem it has been seductively simple, but that is only because the information content of the schema theorem is extremely low. In contrast, obtaining information concerning the sampling of strings outside the current population will require effort.

The community is urged to take up the task of amending $\mathcal{G}$ for GA variants. Once exact sampling distributions are in hand, a solid foundation for attaining provable results will have been laid.

Are Schema Necessary?

As was pointed out in the introduction, knowing the schema is equivalent to knowing the population; either can be computed from the other. This observation justifies the position that the population is the primary object of interest.

Since the function $\mathcal{G}$ deals with populations directly, schema can therefore be dispensed with.

What would replace schema, and how would arguments traditionally made using them be carried out? Since the algorithm in the previous section is indistinguishable from a simple GA, the function $\mathcal{G}$ must encode all relevant information; anything which ever could be proved about the GA must correspond to a property of $\mathcal{G}$.

For example, deception – in the sense of evolutionary trajectories leading away from the optimum – manifests itself in properties of the fixed points of $\mathcal{G}$. Without doubt, arguments based on schema utilities (uniform or nonuniform) are suspect. There exist examples where mutation rates (which leave schema utilities unchanged) of 0.006 reverse the outcome of genetic search!

Note the *dissimilarities* between this and Theorem X. It is evident that the second event could very well be Heads even if the first were Tails. Unlike Theorem X, this probabilistic description is able to indicate that exploration takes place. In fact, it can be used to prove that (with probability 1) coin tossing will explore beyond the possibility given by the first toss.

Next consider the competition between the competing events {Heads, Tails}. The distribution offers no inequalities unable to answer the question, but provides the definitive response: *it is a tie*.

Perhaps this same approach could be applied to genetic algorithms. In fact, it already has. The chronology is

Author(s)	Date	Comments
Bridges & Goldberg	1987	Valid only for mutation rate zero
Vose	1990	Complete model
Davis	1991	Complete model
Whitley	1992	Valid only for mutation rate zero

Much has been done beyond what will be indicated here;[3] it is the simplist aspects that are of present concern. In particular, the authors above have constructed a function, call it $\mathcal{G}$, which given the current population x produces the exact sampling distribution $\mathcal{G}(x)$ determining the next generation. In other words, the following algorithm is indistinguishable from a simple genetic algorithm:

1. Generate a random population x.
2. Compute the distribution $\mathcal{G}(x)$.
3. Determine the next generation by sampling strings according to the distribution in step 2.
4. Replace x by the result of step 3 and go to step 2.

Lest there be any possible confusion regarding this point, "indistinguishable" means what is says. For *all* finite population sizes, creating the next generation by sampling from the distribution which $\mathcal{G}$ provides is indistinguishable from the stochastic transition taken by a genetic algorithm.

Note the *dissimilarities* between this and the schema theorem. This probabilistic description is not trapped within the initial population. It is able to indicate that exploration/recombination takes place because the exact distribution of what new strings are generated and what building blocks are assembled is provided by $\mathcal{G}$.[4]

A consequence of this probabilistic model is that $\mathcal{G}(x)$ is also the expected next generation. Unlike the schema theorem, no inequalities are offered. The expected outcome of every competition – if schema competitions are of any interest – is answered exactly. Each of the 62,712 schema competitions for the population listed in the previous section is decided by $\mathcal{G}(x)$ compared to the schema theorem's zero.

At issue is not just whether $\mathcal{G}$ decides what the schema theorem cannot.

[3] Of particular significance is the Markov Model of Davis, and of Vose.

[4] If one is interested in schema, summing $\mathcal{G}(x)$ over elements of the schema provides that information.

String	Fitness
101101111110	11.000
011101011000	12.000
010010101110	12.000
101110110000	10.000
111111100101	10.000
001011011010	9.000
001001100111	13.000
110001111010	11.000
000001100111	9.000
101001101101	11.000
111101011101	10.000
110010001110	11.000
010010101000	8.000
011101111000	9.000
000100111100	9.000
101101111011	11.000
010101100001	12.000
000011101100	9.000
000100100010	10.000
101101111101	11.000

The objection could be raised that the problem is extremely difficult, or that this population is atypical. Yes it is hard, but the virtual inability of the schema theorem to decide competitions is not unusual for this problem. A typical run will show less than 1 out of every 2,000 competitions decided.

Another possible objection might be that the schema theorem says virtually nothing because almost nothing can be said. This is an important issue to which the next section is devoted.

The function $\mathcal{G}$

There are certain realities which must be faced. Perhaps the most difficult is the fact that genetic algorithms are nondeterministic. Asking any theorem to predict the next generation is an impossibility because the next generation is determined stochastically.

Although the flip of a fair coin cannot be predicted, it can be described. The distribution assigning probability 0.5 to Heads and 0.5 to Tails is the cornerstone for whatever predictions are possible (i.e., laws of large numbers, the law of the iterated logarithm, the Arc sine law, etc.). What makes this distribution important is the fact that the following algorithm for generating a sequence of Heads and Tails is indistinguishable from that generated by tossing a coin:

1. Determine the next outcome by uniformly sampling from the set {Heads,Tails}
2. Go to step 1.

So far, the schema theorem is looking a lot like Theorem X. It satisfies property 1.

Next consider the schema theorem's conclusion, a lower bound. Suppose it is guaranteed that for a pair of loaded dice

- The expectation of rolling an even total is at least 0.47
- The expectation of rolling an odd total is at least 0.42

Which then is more likely, rolling even or odd? In actuality, the expectations could be 0.48 for even and 0.52 for odd. These probabilities are compatible with the given lower bounds. There is nothing to prevent an odd total from being the more likely event.

Just as for this dice example, the type of information provided – lower bounds – may be inadequate to decide schema competitions. Only when lower bounds indicate increasing representation is a schema's growth insured. Only when lower bounds indicate increasing representation of its complement is a schema's decline guaranteed.

At this point, Theorem X has been identified, it is the schema theorem.

The following attempt at making the schema theorem explain genetic search will clarify the severity of its limitations. Let P_0 be the initial population. Apply the schema theorem to P_0 and collect the information concerning schema competitions which it is able to provide. Next, use that information to produce the expected next generation P_1.

This is where the difficulties begin. What if very few schema competitions were decided? For the sake of argument, assume almost all were. O.K., but for those schema with increasing/decreasing representations, how much of an increase/decrease? The theorem provides an inequality, not an expected value! For the sake of argument, assume that somehow doesn't matter. O.K., but for an increasing schema, from where should the additional strings be sampled? Should they all come from the original population?[2]

A little thought reveals that the distribution describing how strings outside of the current population are sampled is crucial to the evolutionary trajectory followed by a GA. Even if the previous two "for the sake of argument" assumptions are granted in its favor, the schema theorem cannot describe how new strings are to be sampled.

Each of these problems alone challenges the idea that the schema theorem can explain genetic search. When combined, the belief is destroyed.

It is interesting to reconsider the first difficulty. Perhaps a large number of competitions are always decided. The following initial population is for a problem similar to autocorrelation (string length 12, population size 20, crossover rate 0.8, and mutation rate 0.05). This population was not contrived, it was simply chosen randomly. However, the schema theorem cannot decide a single competition.

[2]That is the assumption behind the information provided by the schema theorem.

While it is true that selection will not move out side of the initial population, a genetic algorithm with zero mutation and zero crossover rates is an uninteresting degenerate case. Genetic search should not be trapped within the confines of the initial population. Exploration and recombination should create strings beyond those boundaries. New strings should be generated and new building blocks assembled.

Next consider the second property. If the representation of schema were unimportant to the direction of genetic search, then Theorem X attempts (without complete success) to addresses the wrong issue. On the other hand, if schema matter and if – for the benefit of theorem X – the generous assumption is made that competition between schema for representatives in generation n matters only for those strings contained in generation 0 (remember property 1), then Theorem X may not be able to provide relevant information since it can not predict representation for an arbitrary schema.

Unable to address search beyond the boundaries of the initial population and unable to decide arbitrary schema competitions involving those strings originally present; Theorem X does not appear to have significant value. In light of these considerations, it would be strange if Theorem X were regarded as an important theorem. To imagine it as the fundamental theorem of a discipline might be stretching the thought experiment too far.

The Classical Schema Theorem

The classical schema theorem relates the representation of a schema in the current population to its expected representation in the next generation. To gain understanding of what the schema theorem actually says, a brief review of its proof is in order.

The salient points are few and simple. At some point attention is focused on a string belonging to the current population. By considering its relative fitness, its expected number of copies (in the gene pool) is obtained. A lower bound on the probability of schema survival follows from recognizing that any schema which the string currently represents might still be represented by its child. The principle used is that a schema will be represented in the next generation when represented in the gene pool and not disrupted.

The important point to notice is that it is only strings in the current population and how their sub strings may survive to the next generation which is being kept track of. This information is expressed in aggregate form (schema are aggregates of stings) and an inequality concerning schema is thereby obtained.

The classical schema theorem says nothing about if or how a schema might be sampled at strings which are not in the current population (there are no "source terms"). The possibility that strings outside of the current population could be produced is treated as nonexistent. In fact, the theorem could never imply that exploration/recombination took place. Its proof is valid when mutation and crossover rates are zero – when exploration and recombination are certain not to exist! In particular, it can have no logical consequence which would be stronger than a conclusion based on zero mutation and zero crossover rates. In other words, the schema theorem is only a statement concerning how selection focuses the initial population. The occurrence of crossover and mutation rates in its conclusion serve only to weaken the resulting inequality.

Very well, but why schema? Let $\pi_j : \{0,1\}^\ell \to \{0,1\}$ be projection to the j th coordinate:

$$\pi_j(d_1 \ldots d_\ell) \quad = \quad d_j$$

It is a simple exercise to verify the following. Given any (nonempty) collection $\mathcal{H}$ of strings, define (the denotation of) a schema s by

$$s_j \quad = \quad \begin{cases} * & \text{if} \quad \pi_j(\mathcal{H}) = \{0,1\} \\ 1 & \text{if} \quad \pi_j(\mathcal{H}) = \{1\} \\ 0 & \text{if} \quad \pi_j(\mathcal{H}) = \{0\} \end{cases}$$

The set of strings that may be generated by crossover beginning from $\mathcal{H}$ is exactly s. In other words, it is precisely schema which are explored by crossover. Observe that "explored" is the right word in this context. For example, if $\mathcal{H} = \{001011, 011111\}$, then every member of $0*1*11$ can be generated, including the strings 011011 and 001111 which are not originally contained in $\mathcal{H}$.

Crossover is traditionally regarded as the central exploratory mechanism of the genetic algorithm. While mutation may be important and/or effective, the tendency has been to label algorithms which omit crossover as something other than "genetic algorithms". If crossover is the definitive operator for genetic algorithms, and if schema characterize the subsets which crossover explores, perhaps that answers the question "why schema?".

There is another reason however. Not only does crossover explore schema, it recombines them.[1] For example, crossover may produce 001111 from the parents 011111 and 001011. This is an example of (members of) the schemas $***11*$ and $00****$ being recombined to form (a member of) the schema $00*11*$. Thought of in this way, crossover can assemble the "building blocks" represented by $***11*$ and $00****$ into into something new, the composite $00*11*$.

Theorem X

Given that *exploration* and *recombination* of schema are definitive properties of genetic algorithms, perhaps a theorem concerning schema could address these issues and relate them to the direction of genetic search. Indeed, this is what the classical schema theorem is widely believed to be all about.

Consider the following thought experiment. Imagine a theorem, call it "Theorem X", having the following properties:

1. Theorem X *can not* indicate that exploration/recombination takes place.
2. Theorem X relates the representation of schema in the current population to the representation in the next generation. However, it *can not* decide if an arbitrary schema's representation is expected to increase or decrease.

Given property 1, Theorem X could be some statement concerning the initial population. However, if the scope of Theorem X was limited to only those strings originally present – which is forced by no exploration or recombination – then it would not be very useful.

[1] These are actually two aspects of the same thing; crossover can not explore without recombining.

A Critical Examination Of The Schema Theorem

Michael D. Vose

Computer Science Dept., 107 Ayres Hall
The University of Tennessee
Knoxville, TN 37996-1301
vose@cs.utk.edu

Abstract

The classical schema theorem is examined with the aim of discerning what it really has to say. The conclusion reached is probably controversial, but inescapable. Alternatives to the classical schema theorem are discussed in general terms.

Introduction

The classical schema theorem is usually regarded as describing the propagation of schema from one generation to the next. However, if the composition of the population is unknown, then determining the representation of schema is impossible. Conversely, if the schema are not known, then neither is the population.

The fact is that knowing the schema is equivalent to knowing the population; either can be computed from the other. This is natural enough. Otherwise, frequent appeals to the schema theorem so often heard concerning the direction of genetic search would have little force or effect.

By carefully examining the relationship between schema and population, this paper uncovers what the classical schema theorem really does say about genetic search. Our conclusions may at first seem controversial. However, careful consideration will show them inescapable.

Alternatives to the classical schema theorem are discussed in general terms. Their relationship to schema suggest that the direction taken by the theoretical development of the discipline may be making schema into historical artifacts.

Why Schema?

First, what are schema? Classically, one takes them to be collections of strings of a special kind. Assuming binary strings of fixed length ℓ, a schema is usually denoted by an element of $\{0,1,*\}^\ell$. The correspondence between the denotation s of a schema and the collection of strings represented is given by

$$s_1 \ldots s_\ell \quad \longleftrightarrow \quad \{d_1 \ldots d_\ell \,:\, \forall j \,.\, s_j \neq * \implies d_j = s_j\}$$