# An Introduction to Cultural Algorithms

Robert Reynolds
Wayne State University

# AN INTRODUCTION TO CULTURAL ALGORITHMS

ROBERT G. REYNOLDS
*Department of Computer Science 431 State Hall Wayne State University*
*Detroit Michigan 48202*
*e-mail: reynolds@cs.wayne.edu*

## ABSTRACT

In this paper a computational model of the cultural evolution process is described. This model integrates several traditional approaches to modeling cultural evolution into a common conceptual framework. This framework depicts cultural evolution as a process of dual inheritance. At the micro-evolutionary level there is a population of individuals, each described in terms of a set of behavioral traits. Traits are passed from generation to generation at this level by means of a number of socially motivated operators. At the macro-evolutionary level, individuals are able to generate "mappa" that generalize on their experience. These individual mappa can be merged to form group mappa and these group mappa can be generalized and specialized using a variety of generic and problem specific operators. A specific implementation of Cultural Algorithms is described using Genetic Algorithms to represent the population space and Version spaces (or lattices) to represent the set of possible schemata that can be produced via generalizations on the population space. Individual and group mappa are defined as subspaces within the lattice. It is shown how the addition of a belief space to the traditional Genetic Algorithm framework can affect the rate at which learning can take place in terms of the modifications that it produces in the traditional schema theorem for Genetic Algorithms.

## 1. Introduction

Evolutionary Computation is the metaphorical use of concepts, principles, and mechanisms extracted from our understanding of how natural systems evolve to help solve complex computational problems. Currently much of this work has focused on the processes of natural selection and genetics[1]. However, it has frequently been suggested that cultural evolution enables societies to evolve or adapt to their environments at rates that exceed that of biological evolution based upon genetic inheritance alone. It is the goal of the work described here to describe a computational framework in which to express different models of cultural evolution in such a way that the reasons for this increased rate of change can be isolated, and harnessed in order to solve very large-scale computational problems.

Culture is defined by Durham as a "system of symbolically encoded conceptual phenomenon that are socially and historically transmitted within and between populations" [2]. It has been suggested by Renfrew that over time humans have evolved a unique set of capacities that support the formation, encoding, and transmission of cultural information[3]. A key factor underlying all of these capacities is the formulation of categories. Although the categories are represented by symbols, he suggested that the ability to formulate the categories via experience is the critical factor in the development of a cultural capacity. The symbolization of an individuals past experience and forecasts concerning future experiences is internalized in what Renfrew refers to as a world map, or "mappa" in his THINKS model. These individual mappa can be merged, generalized, and specialized in order to form group mappa. These group mappa serve to direct the future actions of the group and its individuals. Most approaches to modeling cultural evolution presuppose individuals with the kind of capabilities suggested by Renfrew. It should be noted that these qualities are not unlike those given to interacting agents in a Distributed Artificial Intelligence application.

In recent years, researchers have attempted to model the cultural evolution process from both a micro-evolutionary perspective in terms of the transmission of behaviors or traits between individuals in a population and a macro-evolutionary perspective in terms of the formation of generalized beliefs based upon individual experiences. These generalized beliefs can serve to constrain the behaviors of individuals within the associated population. A dual inheritance cultural system supports the transmission of information at both the individual and group level. Cultural Algorithms are a class of computational models of cultural evolution that support such a dual inheritance perspective. This approach provides a framework in which to describe all of the current models of cultural evolution from a computational point of view since any of the single inheritance systems can be produced as a special case.

In section two, a general description of Cultural Algorithms is provided. Specific versions can be implemented based upon the representation of the population space and the belief space. One such instance is termed Version-Space Guided Genetic Algorithms (VGA). In this approach, the micro-evolutionary process is modeled using genetic algorithms while the belief space represents schemata or generalizations of the individual chromosomes based upon their behaviors. The schemata are structured in terms of a lattice or Version Space. The basic framework for representing the VGA is given in section 3. The paper concludes with a discussion of how the Schema Theorem for GA's is modified in the VGA. What these modifications mean in terms of the relative rates of search is discussed.

## 2. Cultural Algorithms

In this section, an informal overview of Cultural Algorithms is presented. The goal of the section is to give the basic properties that need to be supported within the Cultural Algorithm framework. While it is possible that a given version may not utilize all of the features, it is important that they all be supported potentially. The model described here is an expression of Renfrew's THINK model in terms of a dual inheritance framework that includes a belief space consisting of individual and group mappa and a trait-based population space.

In the model presented here, each individual can be described in terms of 1) a set of traits or behaviors and 2) a mappa or generalized description of their experiences as suggested by Renfrew. Traits can be modified and exchanged between individuals by means of a variety of socially motivated operators. Likewise, individual mappa can be merged and modified to form "group mappa". Various merging and modification operators are possible and will be discussed later. Operators for the modification of traits and mappa can be either generic (problem independent) or problem specific.

The symbols used to characterize traits and mappa can also be modified over time based upon experience. It is possible that traits can be lost from the population or added. In addition, symbols that are used to represent mappa can be forgotten and new symbols added. Thus, the representation of the trait sequences and the mappa can themselves evolve as a result of the groups experiences.

At any given time step in the model there are a set of individuals in the population space, each described in terms of currently applicable traits. The performance of each individual in solving a set of selected problems is EVALUATED. In addition, each individual will produce a generalized map of their experience during that time period. This process is called OUTLINING. The most general belief in a generated mappa is called its dominant belief. An individual's mappa can then be merged with currently existing group mappa in the belief space if the conditions for one or merging operations are met. If it cannot merge, it remains separate for that time step in the belief space. When mappa are merged, the performances of the individuals associated with them are COMBINED in some fashion. If the combined performance of a mappa is less than some ACCEPTABLE level then that mappa is discarded or PRUNED from the belief space. The ADJUSTED belief space at a given time step is the set of currently ACCEPTABLE group mappa. Discarded mappa can be enforced or not at the population level. If the discarded mappa is enforced, then no individuals possessing the associated beliefs are allowed in the group in the future. If the discarding of mappa is not enforced by the group, then individuals with associated beliefs can reappear in future populations.

The current state of the belief space can then be used to modify the performance of individuals in the population, modify the set of allowable traits, enforce discarded mappa etc. The population is then used to generate a new population through the SELECTION of individuals to be parents for the next generation. These parents are used to EVOLVE a new population via the application of various modification operators. As such, the process is inherently parallel since it is possible that there can be many group mappa residing in the belief space, each supported by a subset of the current population and exploiting some niche or portion of the current problem solving environment. The processes of OUTLINING, and MERGING and PRUNING together with the COMBINE and ACCEPTANCE functions determine how the space of beliefs will be searched in parallel by individuals from the population.

How the current belief space affects the population of individuals and how individuals in turn affect the belief space is mediated by the nature of the communication channel or PROTOCOLS that interconnect them. There are several possible protocols. The standard protocol, VOTE-INHERIT-PROMOTE, supports the process of associating the performance of an individual with a mappa in the belief space(VOTE), then allowing the mappa to INHERIT the individuals performance, and finally PROMOTING those individuals in the population associated with current group mappa. Other protocols involve changing trait and belief representations.

The pseudo code description of the above informal model is given below. The next section describes a particular implementation of this general model.

```
CULTURAL ALGORITHM
begin
      t=0;
      Initialize Population POP(0);
      Initialize Belief Network BLF(0);
      Initialize Communication Channel CHL(0);
      Evaluate (POP(0));
      t=1;
      repeat
            Communicate (POP(0), BLF(t));
            Adjust (BLF(t));
            Communicate (BLF(t), POP(t));
            Modulate Fitness (BLF(t), POP(t));
            t = t+1;
            Select POP(t) from POP(t-1);
            Evolve (POP(t));
            Evaluate (POP(t));
      until (termination condition)
end
```

## 3. Version Space Guided Genetic Algorithms (VGA)

Various Cultural Algorithms can be produced depending upon how the population space and the belief space are represented. For example, various computational frameworks can be used to represent the population space such as Evolutionary Programming, Evolution schemas, and Genetic Algorithms among others. Various symbolic representations can be used to describe the belief space. Such possibilities include semantic networks, logic, and set theory among others.

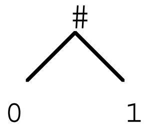
In the example described here Genetic Algorithms were selected as the framework in which to describe the population. One reason for this choice is that the rate at which Genetic Algorithms can explore a space of possibilities is described in terms of a specific formulation, the Schema theorem developed by Holland[4]. One of our goals in the computational characterization of cultural evolution is to investigate the reasons why the rate of evolution may be accelerated by the process. The presence of the Schema theorem will provide us with a benchmark for our investigations.

A natural way in which to express generalizations about the performance of individual population members in a Genetic Algorithm is in terms of schema. A basic way of organizing these generalizations from a set theoretic point of view is a lattice or Version Spaces as described by Mitchell[5]. Each schemata will represent a subset of possible individuals and therefore the schemata can be organized relative to the size and composition of their subsets as a lattice. The root of the lattice is the schema that describes all possible individuals. This representation has a nice correspondence with Renfrew's mappa paradigm. He suggests that one of the keys to the development of culture is the ability to classify objects into groups. That is the fundamental basis for information within the lattice. Some subsets of objects will be more useful in solving problems than others. The ability to identify the useful collections of objects and manipulate them abstractly is the basis for cultural evolution.

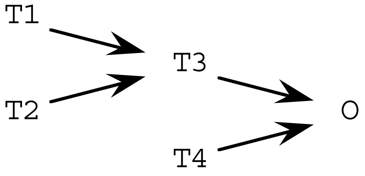
One of the important features of Cultural Algorithms is the ability to shift the way in which traits are represented as well as beliefs. In other words, the system must have the capability of building up a language of traits and belief that corresponds with the populations experiences in a systematic fashion. Therefore, the way in which chromosomes in a population of individual trait sequences are structured must be able to change over time based upon the groups experience. In order to do this, the structure of chromosomes in the Genetic Algorithm population must be specified in an algorithmic fashion. Therefore changes to the representation can be explicitly specified in terms of changes to the generative algorithms and the data structure of traits that it uses. How this is done specifically for this system will be discussed in the next section.

### *3.1 Representing the Population and Belief Spaces*

Each individual in the population is characterized in terms of a set of properties or traits. A trait is a value taken over a hierarchically structured collection of terms, where the most general term is found in the root, and the most specific terms found at the leaves. This specification can be viewed as a semantic network. Each hierarchy is well structured in that there is a path from the root to any given term in the hierarchy. The term hierarchy for a binary valued trait is given below where a # corresponds to a don't care condition, or a set containing both values.



Given that certain traits require other traits as prerequisites, one can define a prerequisite graph structure where an arc from trait i to trait j exists if trait i is required for the development or specification of trait j. An example of such a prerequisite structure is given below.

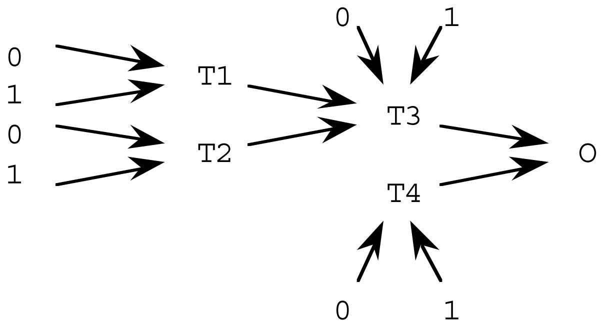


In this case there are two independent prerequisite trait structures connected together where O is a dummy node. Here, T3 is supported by two prerequisites, T1 and T2, while T4 has no prerequisites. Note that the standard chromosomal structure for Genetic Algorithms without prerequisite information can be modeled by linking all of the traits directly to the dummy node in a star-like configuration.

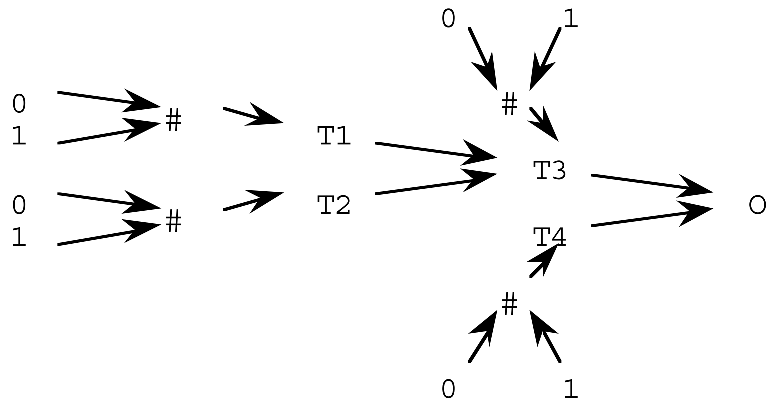
The set of chromosomes used to specify individuals are produced from this data structure using an algorithm that performs a topological sort on the prerequisite structure for the given of traits. A topological sort is the assignment of a linear ordering to the vertices of a directed acyclic graph so that if there is an arc from a trait i to a trait j in the graph then i must appear before j in the linear ordering. The exact ordering of traits in the chromosome depends upon the precise graph traversal algorithm used. Here, we assume a basic post-order traversal of the tree. In the example above the trait sequence that would be generated via the post-order traversal is T1, T2, T3, T4.

It is important to note that specification for a chromosome is specified in terms of the traversal of a semantic net data structure by a specific traversal algorithm. New representation of individual chromosomes can now be produced by changing either the semantic network data structure or the traversal algorithm used. Such changes can now be triggered by particular events in the problem solving process for a group. These events can relate to properties of the belief space, the population space, or both.

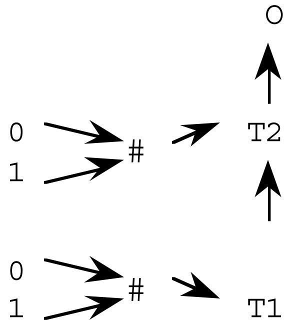
Once a trait sequence has been determined, all of the possible chromosomes for those traits can be generated by augmenting the prerequisite graph with the leaf nodes from the term hierarchy associated with each trait. This is called the trait network for the prerequisite graph. Assuming the prerequisite structure given above and a binary term hierarchy for each trait, the trait network is given below.

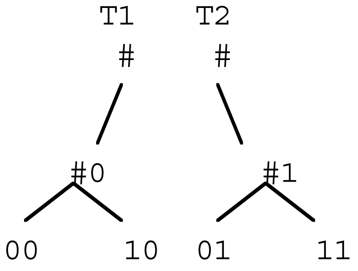


All allowable chromosomes can be generated by traversing the trait network for a given topological sort, and visiting one of the possible trait values for each trait visited. For the topological sort that produces the linear ordering of T1, T2, T3, T4, a chromosome of 1001 would be produced if a 1 was visited for T1, a 0 for T2, a 0 for T3, and a 1 for T4. All other possible bit sequences can be produced for a given trait sequence in this manner. The belief space can be generated by augmenting the prerequisite graph with the complete term hierarchies for each of the traits. This allows the positioning of more generalized terms in the representations. This augmented graph is termed the belief network. The belief network for the above example is given below.



Just as the trait network can be used to generate the population of chromosomes, the belief network can be used to generate the set of all possible beliefs that can be produced as generalizations of the trait sequences. The belief space for a given topological ordering corresponds to the set of all unique combinations of terms assignments that result from the traversal of the belief network  using a given topological sort. Each of the generated beliefs corresponds to a set of individuals in the chromosome population. These beliefs can be organized as a lattice, with the most general belief at the root and the most specific beliefs at the leaves.  In the above example the root node will be ####. Its children in the lattice will be those beliefs produced by replacing one of the #'s with a term value immediately below it in the term hierarchy for the selected trait. The leaf nodes of the lattice are the possible chromosome sequences themselves. To illustrate, the belief network below yields the corresponding belief space.





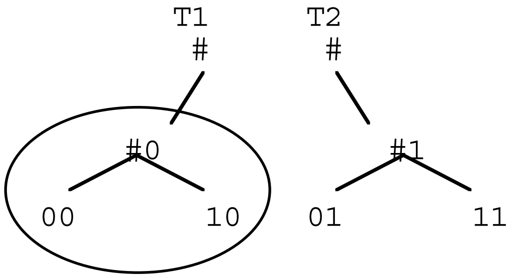
Note that only a portion of the belief space is shown here. In this case T2 is elaborated upon first, followed by T1. Elaborating on the #'s in this order will augment the space accordingly. However, one can restrict the size of the belief space by requiring that traits be elaborated in a top-down fashion in order of their precedence. In this case T2 is of higher precedence than T1, so it is elaborated first.

Mappa are viewed as being subsets of the belief space. Individual mappa are constructed in a data driven, bottom up fashion. The process of producing an individual mappa is called OUTLINING. It presupposes an ACCEPTABILITY function and a PERFORMANCE function. For a given population of individuals, a given individuals performance can be assessed. Then it is classified as being "acceptable" or "unacceptable" by the ACCEPTANCE classification function. The goal of the OUTLINING process is to identify an equivalence class of beliefs in the network that are adjacent to the original individual. In particular, it identifies a sub lattice of adjacent beliefs that have the same value for the ACCEPTABILITY function as the original individual. Thus, mappa are viewed as being sub lattices in the belief space.

An individual mappa can be described in terms of two sets. The first set contains the most general belief that is supported by the mappa. This is called the G-set or Generalization set using Mitchell's terminology. Beliefs in this set are known as DOMINANT beliefs for that mappa since they subsume all of the others within the sub lattice. The second set describes the set of leaf nodes that were searched in order to produce the generalizations in the G-set. This is called the S-set or Specialization set. All of the nodes in the sub lattice between and including the root and leaves, by definition, the same acceptance value. This motivates an alternative name for mappa, stable class. This is because the sub lattice is a set of subsets, all of which are stable, have the same value, with respect to the acceptance function.

The OUTLINING process is basically a local search activity. It generates modifications to the individual in a systematic manner in order to explore the local population space in order to see how perturbing the values for specific traits affects the ACCEPTANCE function. Numerous search heuristics can be employed in order to guide the local search. The sophistication of the search heuristic used is a function of the problem size. For small lattices, simple breadth first and depth first searches are satisfactory[5]. For large spaces, more sophisticated heuristics such as parallel bi-directional A* have been employed[6].

The basic idea can be illustrated using the portion of the belief space described earlier and shown again below. Assume that one of the leaf nodes, 00, is a member of the current population and has an associated performance value of f. Also assume that f is determined to be "acceptable" relative to the ACCEPTANCE function. Assume that the OUTLINE function used here is a breadth first search of the population space from 00. Each of the two bits is changed one at a time to see if the new configuration produces an unacceptable solution. 10 produces an acceptable solution as 00 did. Therefore, there is no evidence that the first trait affects the acceptability of the solution. On the other hand, changing the second bit to a 1 results in an unacceptable solution. Thus, the second bit is fixed at 1 and the most general schemata that supports all of these experiments is #0. The G-set or dominant belief for the mappa is #0 and the S-set is 00 and 10. The performance for the mappa is the combined performance of its individuals. The mappa produced by the OUTLINING process here is the highlighted sub lattice below.



This mappa can then be compared to existing group mappa to see if there is any basis for merging them. There are many different ways and reasons to merge mappa. One basic one is that one is subsumed by another in some fashion. Other problem specific merge operations are possible. Much of the power of the belief space stems from the ability to manipulate the mappa expressed within it. Once all possible merging operations have been performed then the performance of the remaining mappa can be transmitted back to the individuals that correspond to them in the population. This performance information can be used to augment the individual performance information for each. After updating the performance of the current population, a new population of individuals by copying successful ones in proportion to their performance and applying the operations of crossover and mutation to produce new individuals.

## 4. Modifying the Schema Theorem

One key question is the extent to which the VGA described above is able to speed up the learning process over that of a GA alone? The basic schema theorem associated with GA's developed by Holland is given below.

$$m(H,t+1) >= m(H,t)\frac{f(H)}{\bar{f}}\left[1 - p_c\frac{dlen(H)}{len - 1} - p_m\ o(H)\right]$$

For a given schema, H, the number of instances of it in the next population, m(H, t+1), will be a function of the current number of instances, m(H, t), its fitness relative to the average fitness of the current population, and the likelihood that its structure will be disrupted by the actions of the two genetic operators. The likelihood of disruption is expressed as 1 minus the probability of changing one of the fixed bits and that of replacing a sequence via crossover.

How is this relationship between schemata modified in the VGA implementation of Cultural Algorithms? In order to answer this question certain assumptions must be made. First, as in the Schema Theorem it is assumed that the standard operations of mutation and crossover are used, and that reproduction is proportional to the fitness of an individual. In addition it will be assumed that the standard Vote-Inherit-Protocol links the belief and population components, and that the performance of each individual chromosome will be the sum of the performance for each of the active mappa that it represents. Also, assume that the performance for a mappa is the average performance of its population elements. The performance of any given schema is that of the dominant belief in the active mappa that contains it. In addition, it is assumed that the ACCEPTANCE function enforced a specific lower bound on the performance of acceptable mappa. Mappa with unacceptable performances are deleted.

For the VGA configuration given above, the only difference visible to the GA component is the value for the overall performance, f. The key here is that it has now been set to the accumulated average of all active mappa over all time steps. In fact, that value is guaranteed by the pruning operation to be at least as high as the lower bound on the acceptable performance set in the ACCEPTANCE function, assuming that at least one acceptable instance has been found. Thus, f, rather than being the current average for the population is instead the average for all active mappa and must be at least as high as the bounds set in the ACCEPTANCE function. In fact, active pruning of unacceptable mappa will effectively raise the value to levels higher than that lower bound.

In the GA version of the Schema Theorem there was no restriction on f so it was possible that if the average was low in a given time step, then lesser performing schemata would be afforded a free ride. In terms of the VGA, the performance level is always constrained to be at least a certain amount by the ACCEPTANCE function so this will put more selective pressure on existing schema to perform well. In addition, the overall performance is not computed over just a single time step but accumulated over all time steps. Thus, it is less likely that there will be major fluctuations in the average over time. This will also put more selective pressure on the active schemata.

## 5. Conclusions

From a computational point of view, the basic reason why cultural evolution can proceed at an increased rate is that it is able to 1) Provide selective pressure on the population by placing constraints on their performance, and to 2) maintain a history of individual performance that is separate from that individual. Both of these characteristics are key factors in influencing the performance speedup associated with the specific version of Cultural Algorithms described here.

For example, in terms of the VGA the presence of the belief space has the potential to speedup the rate of evolution for the GA by influencing the overall fitness of active schema. The presence of an ACCEPTANCE function can set a lower bound on the overall performance of active mappa and their associated schema. In addition, the presence of pruning can effectively raise this lower bound over time. Both of these activities serve to provide stronger selective pressure on schema in the population and effectively guides the search of the GA population towards regions of the belief space associated with high performing mappa. The extent to which pruning of the belief space can affect the rate of learning in a Cultural Algorithm is a topic for future research.

## 6. References

1. D. B. Fogel, "An Introduction to Simulated Evolutionary Optimization", IEEE Transactions on Neural Networks, Volume 5, Number 1, 1994.

2. W. Durham, Co-Evolution: Genes, Culture, and Human Diversity, Stanford University Press, Stanford, CA, 1994.

3. A. C. Renfrew, "Dynamic Modeling in Archaeology: What, When, and Where?", Dynamical Modeling and the Study of Change in Archaeology, S. E. van der Leeuw, ed., Edinburgh University Press, 1994.

4. J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan, 1975.

5. T. M. Mitchell, Version Spaces: An Approach to Concept Learning, Ph.D. Dissertation, Stanford University, Stanford, CA, 1978.

6. P. C. Nelson, L. J. Henschen, "Parallel Bidirectional Heuristic Searching", Proceedings of Canadian Information Processing Society, Toronto, 1987, pp. 117-124