An Approach For Evolving Novel Organizational Forms

Kevin Crowston

crowston@umich.edu

School of Information Studies

Syracuse University

Appeared in *Journal of Computational Organization Theory*

This paper is based on a version originally published in K. M. Carley and M. J. Prietula (Eds.), *Computational Organization Theory*. Hillsdale, NJ: Lawrence Erlbaum, 1994, pp. 19–38.

The author gratefully acknowledges the support of the Ameritech Foundation, through the Information and Organizations Program of the University of Michigan Institute for Public Policy Studies, the University of Michigan Center for Parallel Computing, which is partially funded under NSF Grant CDA-92-14296 and the Massachusetts Institute of Technology Center for Coordination Science, which is partially funded under NSF Grant IRI–92–24093.

The paper has benefited from discussions with Michael Gordon, Helen Klein, Michael Cohen, Brian Pentland, Martha Feldman, Tom Finholt, Ojelanki Ngwenyama and Kathleen Carley and from the comments of anonymous reviewers.

An approach for evolving novel organizational forms

Abstract

A key problem in organization theory is to suggest new organizational forms. In this paper, I suggest the use of genetic algorithms to search for novel organizational forms by reproducing some of the mechanics of organizational evolution. Issues in using genetic algorithms include identification of the unit of selection, development of a representation and determination of a method for calculating organizational fitness. As an example of the approach, I test a proposition of Thompson's about how interdependent positions should be assigned to groups. Representing an organization as a collection of routines might be more general and still amenable to evolution with a genetic algorithm. I conclude by discussing possible objections to the application of this technique.

1 Introduction

It is common place to note that the environment in which organizations operate is changing rapidly. Businesses are facing increased pressures to design and produce products with higher quality, more rapidly yet more cheaply. At the same time, new technologies are rapidly relaxing fundamental constraints on organizational design, for example, by making communication and data storage much cheaper or by facilitating previously impractical interactions across time or space.

Such rapid changes may pose problems for organizations that have evolved to fit the old environment. In particular, the changes may make novel organizational forms more appropriate. Organizational ecologists explain the diversity of forms by analogy to biological species competing for resources (Singh and Lumsden, 1990). Organizations with forms more appropriate for the environment are more successful at acquiring resources and thus tend to survive while those with less effective forms tend to fail. Over time, this selection results in an observable match between organizational forms and the environment in which they operate. Organizational ecologists tend to minimize the role of organizational learning and adaptation and instead see organizations adapting to their environments only as a population.

Unfortunately, these theories only suggest that if the environment changes, new forms may eventually replace existing ones and describe the dynamics of this replacement. They do not provide much insight into exactly what kinds of new form might become desirable. Indeed, Romanelli (1991) calls the question of the origin of new organizational forms, "one of the critical unaddressed issues in organizational sociology."

In this paper, I will discuss the use of computer simulations to search for novel organizational forms by reproducing some of the mechanics of organizational evolution. The metaphor of organizations competing and being selected on the basis of their fitness is a compelling one. Natural selection is also the basis of a heuristic search technique known as the genetic algorithm (GA) used to search large problem spaces (Holland, 1992). If we can describe the space of organizational forms, we can use this algorithm to search it. Note that the GA is simply a search technique. The approach can work even if you happen not to believe that organizational ecology accurately describes how real organizations adapt to their environments, although clearly the mechanism will seem more reasonable if it happens that you do.

Using the GA to search for possible organizational forms has the advantage that variations in organizational forms explored are not restricted by social factors, institutional pressures (DiMaggio and Powell, 1983) or human ingenuity. However, this advantage is simultaneously a major concern about simulations: the potential lack of external validity. Computer models necessarily abstract from real organizations; the features simulated must be chosen carefully to ensure that conclusions drawn from the models are generally applicable (Burton and Obel, 1980).

In the rest of this paper, I describe the GA and discuss some issues in applying it to organizational forms. To illustrate the potential of this approach, I then present a simple model of an organization and some preliminary results from using the GA. I conclude by discussing possible bases for a more general organizational model and general caveats about the approach.

2 Genetic Algorithms

A simple genetic algorithm works as follows. Given a problem, a random population of possible solutions is generated. For example, in the simulation to be presented, the population was of organizations represented as various arrangements of positions into groups. Solutions in the population are evaluated and the most promising ones used in proportion to their fitness as the basis for the next generation of possible solutions. In my simulation, fitness was determined by the number of tasks the organization performed, as explained below. New solutions are created by cross-over, that is, by taking two existing solutions, dividing each in two parts and exchanging parts to create two hybrid solutions. In my simulation, the parts are partial assignments of positions to groups. As well, a small number of mutations are introduced, although in most GAs these changes play only a minor role. In my simulation, a position was moved from one group to another in approximately 1% of the organizations. The process is repeated until a solution successfully solves the problem, until a certain number of generations pass with no improvement in the solutions or, as in my simulation, for some fixed number of generations.

Note that organizations do not change as they are evaluated; rather, the population changes as individuals are replaced by their (hopefully more fit) off-spring. An interesting extension is to allow agents to learn as they perform a given task and then pass some of this learning on to their off-spring (i.e., inheritance for organizations could be Lamarkian instead of Darwinian).

While the GA seems random, it turns out to be an effective method for searching a large search space. It has been used to evolve solutions to many kinds of problems, including strategic games, optimization problems, mechanical design and image classification (Booker, et al., 1987) and even LISP programs (Koza, 1992). It can be shown that the individuals in a population encode large amounts of potentially useful information about different combinations of feature, called schemata (Holland, 1992). These schemata implicitly represent numerous similar individuals not actually present in the population. For example, a single individual with (say) four features implicitly represents all individuals with any one of those features, with any two of those features, etc. Furthermore, as the individuals are selected and bred, schemata are also reproduced in the population in proportion to their fitness. In essence, by manipulating a modestly sized population, a GA implicitly searches in parallel a much larger portion of the space.

As the above discussion makes clear, there are two main issues in applying the GA to a problem such as organizational design.

- *Representation.* First, we need a representation of the organizations to be evolved, general enough to represent any organizational form of interest. Representing only different kinds of hierarchies, for example, would not allow us to consider market mechanisms for the same transactions (Williamson, 1975).

- *Fitness.* Second, we need to represent the environment in which the organizations perform and, explicitly or implicitly, a fitness function to identify "good" organizations. By varying the environment, we can look for forms that may be desirable under different conditions.

These two elements will be discussed in detail in the remainder of this section.

2.1 Representation of Organizations

The first question is how organizations should be represented, or alternatively, what constitutes an organizational form? Romanelli (1991, p. 82) notes that researchers have proposed several approaches to this question but concludes that there is no general agreement or even a common classification scheme for organizations. Instead, this question is answered primarily by researchers' theoretical commitments and the kind of research questions they hope to investigate. For example, an interest in the use of information technology suggests a focus on factors that are directly influenced by the application of such systems, such as communication patterns, costs or capabilities.

The use of the GA does place some constraints on the representation, however. First, we require a representation in which cross-overs between individuals make sense. This requirement implies that an organization is described as a collection of substitutable features that can be combined and recombined in various ways—what Romanelli (1991) and Hannan and Freeman (1986) describe as organizational genetics. Most characterizations of organizations do not have this property. For example, Hannan and Freeman (1986) note that much of organizational research has used conventional classifications of types of organizations—as they say, "We routinely distinguish hospitals, prisons, political parties, universities, stock exchanges, coal mines and fast-food chains" (p. 54)—but it is unclear, what a cross between a hospital and a coal mine is (for example) or how it could be represented.

Hannan and Freeman (1986) and Romanelli (1991) review several candidates for organizational features, including organizational building instructions, transactions and routines. Describing an organization at this level of abstraction has several advantages. First, representing an organization as a combination of simpler features may reveal similarities between seemingly different organizations and vice versa. In other words, the diversity of organizations may result from the combination of a smaller set of building blocks, much as the diversity of living creatures results from combinations of a smaller set of genes. Second, understanding the building blocks may guide empirical study by providing a list of elements to look for and a framework for putting them together. Again, to draw on the analogy to biology, some scientists seek to explain differences between individuals (e.g., risk of a disease) by looking for an underlying genetic difference. This strategy does not always work, but when it does, it provides considerable insight into the causes and implications of the difference.

Finally, such a representation is amenable to experimentation with a GA. For example, given a set of features, an organization can be represented simply as a string of bits encoding the presence or absence of a feature. Organizations can be easily crossed: the two parents are split at a randomly chosen cross-over point (e.g., after the 10th bit) and the first half (e.g., bits 1 to 10) of the first concatenated with the second half of the second (e.g., bits 11 and on) and vice versa to create two new hybrid strategies. As well, solutions can be mutated by changing one or more of the bits. A significant disadvantage is that this representation often determines in advance the size and shape of the final solution, i.e., it is not sufficiently dynamic (Koza, 1992).

A second requirement is that we would prefer that the space of possible organizations be "dense," that is, modifications to the representation of one organization should usually result in another viable organization. Simply representing the formal structure of an organization might be unsatisfactory, for example, because the effects of replacing one department (marketing say) with another (manufacturing) will usually be a non-functioning organization (one with two manufacturing divisions but no marketing). This requirement is not mandatory, but if most crosses are not viable, the GA will have trouble finding a solution. For example, Liepins and Potter (1991) used a GA to diagnose underlying problems from a set of symptoms. They modified the fitness function to return a low value even if the proposed problems were inconsistent with the symptoms. Without this modification, they found that the GA found the optimal solution less than 10% of the time; with it, the optimal was found more than 85% of the time.

A possible approach to this problem is to develop a translation from organizational structures of interest to a representation on which the GA can operate. Similarly, Michalewicz (1992) suggests that for the GA to perform better it is necessary to incorporate, "more problem-specific knowledge in the chromosomes' data structures" (p. 7). Bean (1994) has experimented with several techniques for mapping a dense and simple space into a solution space, e.g., mapping a sequence of N random numbers into a route for the travelling salesman's problem or a schedule for a machine shop. More complex representations, such as the hierarchical LISP programs used by Koza, could allow for a wide variety of forms, including forms with complex internal structures as are found in organizations.

2.2 Fitness

Second, we need a mechanism to assess the fitness of individual organizations. The most straightforward approach to this problem is to choose a task and measure the success of the organization at performing it (e.g., the time required to perform the task, number of tasks performed in unit time, total cost, etc.). Picking a good task is key—it must be abstract enough to simulate but have clear application to a real situation. Better yet, it should be one that could be done in many ways, with no clear dominant best form, and for which the features of the organization of theoretical interest are thought to make a difference in performance. For example, if we chose to model communication patterns within an organization, then the task evaluated should be one for which theory suggests how those patterns affect the outcome (e.g., by affecting how long it takes to perform the task).

Characteristics of the environment also determine in part the fitness of organizations. As with organizational forms, there are many characteristics of the environment that could be represented. Freeman and Hannan (1983), for example, examined the effects of environmental variability and patchiness. Again, the characteristics of the environment modelled are those that are believed to affect the performance of the organization on the task.

The environment may also include the other individuals in the population, making the success of a particular individual dependent on the behaviour of the other individuals. For example, Axelrod (1987) used the GA to evolve strategies for playing prisoners' dilemma games, which were evaluated by their success in playing against other strategies. Such models exhibit behaviours that depend on the interaction between individuals, such as symbiosis or parasitism. In such a model, an individual's fitness may be defined only implicitly by its relative success in acquiring resources, as in Holland's Eco models (Holland, 1992).

3 Example: Assignment of Positions to Groups

In this section, I will provide an example of the use of GAs for organizational design. In this example I model entire organizations by encoding an organization's formal structure. The GA is then used to modify these structures, eventually identifying particularly fit organizational forms.

At this initial stage of research, I believe it is most useful to address questions for which there is already some theoretical agreement. This replication will allow us to develop some confidence in the technique before applying it to novel problems. As well, a theoretical base is necessary to suggest what features of organizations and the environment should be modelled. The experiment presented here was developed to test a proposition of Thompson's (1967) about how interdependent positions should be assigned to groups. Thompson suggested that, "In a situation of interdependency, concerted action comes about through coordination" (p. 55) and further noted that coordination requires decisions and communication (in varying amounts), which make coordinating costly. He therefore proposed that, "Under norms of rationality, organizations group positions to minimize coordination costs" (Thompson, 1967). In other words, the problem of organizing is determining which positions should be assigned to which groups.

To develop a model, we need to consider two other factors that influence organizational design and which are implicit in Thompson's formulation. First, we must identify the source of coordination costs. I interpret these as arising from the size and number of groups. There must be a cost to large groups, or else the easy answer is to simply put all positions in the same group. Similarly, there must be some restriction on the number of groups to which a position belongs, or else there could simply be one group per task. Second, we need to specify the benefit of coordinating. In Thompson's statement, this benefit is "concerted action," which I will interpret as increased task performance. If concerted action is not required, no coordination cost is worth paying and no groups should form; in other cases, coordination might be necessary do the tasks at all and group are therefore required.

In general, there will be a trade-off between the costs and benefits of coordinating. Taking these two factors into account, Thompson's proposition can be restated as, "Under norms of rationality, organizations group positions to optimize the tradeoff between the benefit of concerted action and the cost of being in large groups or in multiple groups." It is this restatement that I will test in this experiment.

3.1 Experimental design

To design the simulation, I addressed the two issues discussed above, representation and calculation of fitness.

3.1.1 Representation

To begin simply, I created a model with positions restricted to a single group. In this model, an organization is a list of N positions; each position belongs to one of 2^M groups. An organization is therefore simply a list of N numbers, where each number is the group to which the corresponding position belongs. Group numbers are represented as M-bit binary numbers, so an organization is an NM-bit string. Representing numbers in binary allows cross-over to generate novel group numbers (e.g., by taking the first two bits from one parent and the last two from the other).

In these experiments there were 10 positions and 16 possible groups (0000 to 1111 in binary), so each organization was represented by a string of 40 bits, for a total of 2^{40} or approximately 10^{12} possible organizations. Sixteen groups was chosen to allow each position to be in its own group.

3.1.2 Fitness

The model is characterized by a number of other parameters, as shown in Table 1, but for the experiments reported here most were held constant. Two of the parameters were varied and form the experimental conditions for the experiments: the *interdependency between positions* and *the relative benefit of coordinating*.

Insert Table 1 about here.

Interdependency between positions. To model interdependencies between positions, positions were assigned overlapping sets of tasks. In these experiments the degree of interdependency was varied from none (each position works on a unique set of tasks) to total (all positions work on the same set of tasks). The overlapping sets of tasks were created by deterministically assigning each position some number of tasks and randomly assigning others. For example, if five of ten tasks were deterministically assigned, then there would be 50 tasks total (5 tasks times 10 positions) and the interdependency would be 5. Position 0 would be assigned tasks 0 through 4 plus five tasks chosen randomly from the remaining 45 tasks assigned to other positions; position 1 would be assigned tasks 5 through 9, plus five other randomly chosen tasks, and so on. If all ten tasks are assigned deterministically, then each task would be assigned to only one position and the interdependency would be 0; if only 1 task is assigned deterministically, then each position would perform the same set of ten tasks and the interdependency would be 10. A typical set of positions is shown in Table 2. Table 3 shows the interdependency between the tasks performed by these positions, calculated as the number of common tasks divided by the total number of tasks.

Insert Tables 2 and 3 about here.

Relative benefit of coordinating. In order to calculate the tradeoff between the benefits and costs of coordinating, we must state them in a common metric. In this model, the fitness of each organization is defined to be the sum of the number of tasks each position performs, which in turn is proportional to the time spent on those tasks. Each position was assumed to start with an initial allocation of time. Conceptually, each position then talks to each other position in its group, which diminishes the time available to work on tasks by a constant factor per position. For this experiment, this cost was fixed at $^1/_9$ of a position's initial time so that a position that talked to every other position would have no time left to work on any tasks (i.e., a group that included all positions would get no work done in the absence of other effects).

To model the benefit of coordination, the time a position spends on a task is increased for each other position in the group that is assigned the same task. The amount of the increase is called the coordination benefit and is stated as a percentage of a position's initial allocation of time. The coordination benefit was varied from condition to condition as shown in Table 1. (Conversely, the benefit could have been left fixed and the coordination cost varied from condition to condition.) Negative values of the coordination benefit were included primarily as a sanity check on the algorithm.

3.2 Hypotheses

Organizations in this model are subject to two countervailing forces: as positions are assigned to the same group, they can take better advantage of the synergies between tasks (depending on the degree of interdependency between positions and the value of the coordination benefit), but suffer a reduction in the time available due to the need to communicate with other group members.

Interdependency. For conditions with no interdependency between positions, there is no benefit to being in a group; therefore, for this condition there should be one group for each position. On the opposite extreme, when all positions perform the same tasks, the benefit for being in a group with others is high; if the benefit is high enough, one group should form that includes all positions.

Coordination benefit. When the coordination benefit is zero or negative, again, there is no benefit to being in a group, so again each position should be in its own group. On the other hand, when the coordination benefit is high, all positions should be in the same group, since the benefit will outweigh the cost. For intermediate values of the coordination benefit, groups of intermediate size should form, grouping positions with high interdependency.

To summarize:

Hypothesis 1: When the interdependency is 0, the number of groups will be 10.

Hypothesis 2: When the coordination benefit is negative or zero, the number of groups will be 10.

Hypothesis 3: When the coordination benefit is high and the interdependency is high, the number of groups will be 1.

Hypothesis 4: For intermediate values of interdependency and coordination benefit, groups will form that group positions with relatively high levels of interdependency.

These forces and predictions about the number of groups are summarized in Table 4.

Insert Table 4 about here

3.3 Implementation

The simulation described here was written in C and run on a 64-node KSR2 parallel computer. As Grefenstette (1991) notes, GAs map easily on to a coarse-grained multiprocessor. In particular, evaluation of individuals in a generation can proceed in parallel, as can the breeding of members of a new generation, resulting in nearly linear speedup on these phases. The program has also been run on a single processor Sun-3 and on a Macintosh using Symantec Think C.

There are many variations on the GA. The one used for this paper is what Davis (1991) describes as a "traditional GA" (p. 35). Individuals were represented as strings of bits, as discussed above. Fitnesses were normalized using linear normalization (Davis, 1991, p. 33), that is, organizations were sorted in decreasing order of evaluation and fitnesses assigned starting at a maximum value and decreasing linearly to a minimum value. Davis points out two benefits to the use of this technique. First, normalizing the evaluations spreads out closely spaced organizations, thus heightening competition in a close race. Second, using linear normalization allows a "super" individual to be strongly selected, but not so strongly that it entirely dominates the population. On the other hand, as Michalewicz (1992, p. 57) notes, "selection process based on ranking violate the Schema Theorem."

One-point crossover and mutation were used to generate new individuals. As well, reproduction was elitist (Davis, 1991, p. 34), meaning that the two best individuals from each generation were simply copied to the next, thus preventing them from being eliminated by the vagaries of random selection. Davis notes that this strategy usually improves the performance of the GA, although it may actually hinder performance in the some cases by allowing a high performing yet non-optimal individual to dominate the population too quickly. It was used here first to ensure that the best performing organization would survive to the final generation and because it seemed best to stick to the "traditional GA" for my initial experiments.

3.4 Results

For this experiment, the GA was run 30 times with a population of 1000 organizations for 200 generations for each set of conditions. In each run, therefore, a total of 2×10^5 organizations were considered, which is significantly less than the total number of different forms possible (approximately 10^{12}). The best performing organization seen in each run of the simulation was saved for analysis. Each run of 49 conditions took approximately one hour to complete. A sample of the output from one run for one set of conditions is shown in Table 5. Each position is assigned to a group; in this case, all positions are in separate groups. Table 6 presents the average number of groups formed, by interdependency and coordination benefit; Table 7 presents the average size of the largest group formed. These quantitative results (i.e., the exact number of groups for a particular pair of parameters) are somewhat arbitrary, since they depend on the particular parameter values chosen, but I believe that the general pattern of the results (i.e., that the number of groups is lower for conditions in the lower right corner) is robust.

3.5 Discussion

Hypothesis 1, that when the interdependency is zero, there would be one group per position, is supported. As Table 6 shows, for these conditions, the average number of groups is 10. The relevant question is whether this is a significant result, that is, one that is unlikely to be a product of chance. The role of traditional statistics is to answer such questions by comparing a result to a theoretically derived distribution of the measure in order to determine how likely the result is. Often, the assumption is that the underlying data is normally distributed. Unfortunately, for this experiment the underlying distributions are not known and are almost certainly not normal. Techniques have been developed to deal with such measures; for example, Friedman and Friedman (1995) suggest using the bootstrap method, which involves resampling the data generated to approximate the underlying distribution. Fortunately in this case, the underlying distributions are directly calculable. To test Hypothesis 1, the distribution of the number of groups was determined empirically by simply generating 5000 organizations randomly and counting the number of groups. According to this analysis, organizations with 10 groups occur by chance less than 3% of the time (134 times out of 5000). In other words, the result is significant, and Hypothesis 1 is supported.

Hypothesis 2, that when the coordination benefit is negative or zero, no groups would form, seems to be supported. Again, for these conditions, the average number of groups is 10, which again is unlikely to occur by chance.

Hypothesis 3, that when the coordination benefit is high and the interdependency is high, there will be only 1 group, seems to be contradicted by the results in Table 6, which show an average around 2. Even so, this number of groups is significantly fewer than expected by chance: of the 5000 randomly generated organizations, only 6 had 4 groups and none had more than that.

I believe that this result is due mostly to the asymmetry of the representation. There are only 16 isomorphic representations of organizations with 1 group, while there are $16 \times 15 \times \ldots \times 7$ or approximately 3×10^{10} ways to represent organizations with 10 groups. As a result, it is much easier for the GA to find organizations that satisfy Hypotheses 1 and 2 than Hypothesis 3. Closer examination of the data shows that organizations with only 1 group are found in some but not all runs. Table 7 shows that the average size of the largest group is quite large. It appears that in the remaining runs most of the positions are gathered into one group, with a few stragglers in their own groups.

Additional analyses were necessary to test **Hypothesis 4**, that for intermediate values of interdependency and coordination benefit, groups will form that group positions with relatively high levels of interdependency. A matrix of interdependencies between positions is calculated by counting the number of tasks they have in common, as in the example in Table 3. Once positions are collected in groups, a subset of the interdependency matrix can be identified that includes interdependencies only between positions in the same group. To test Hypothesis 4, it is necessary to determine the similarity between these matrices. Usually a measure of similarity would be chosen to allow statistics to be calculated, based on properties of the underlying data. The data in these matrices are not well distributed, making traditional statistics difficult, but again, traditional statistics are unnecessary since the actual distributions can be directly calculated for any measure chosen. I therefore chose to simply calculate the Pearson's correlation between the upper diagonals of the matrices. Table 8 shows the average correlation for the intermediate results, that is, those with fewer than 10 but more than 2 groups; presumably any other measure would give essentially the same results.

Insert Table 8 about here

The significance of this measure was determined by comparing the calculated correlation to an empirically derived distribution developed by randomly generating 5000 pairs of environment and organization for each level of interdependency and calculating the correlation between the resulting two matrices. According to this analysis, these correlations are much greater than would be expected by chance. In fact, as Table 9 shows, in all but one case the significance of the *minimum* correlation found in the 30 runs is in the upper 5% of the empirical distribution. In the exceptional case (interdependency 5 and coordination benefit .30), the benefits of coordinating just outweigh the costs and in 5 of the 30 runs, no groups were formed, resulting in a correlation of zero, which is not significant. The significance of the correlation in the remaining 25 cases is in the upper 10%.

In other words, for intermediate cases, the groups that formed grouped together positions with higher interdependencies than would be expected if they were formed by chance, that is, Hypothesis 4 is also supported.

Insert Table 9 about here

The model has several possible extensions. First, positions can currently belong only to a single group; the model could be extended to allow positions to be in multiple groups simultaneously. Second, positions could be allowed to form a hierarchy, further testing Thompson's propositions.

4 Example: Interacting Problem Solving Agents

To apply the technique described in this paper more broadly, a more general model of what organizations do is necessary. For this purpose, I will discuss how an organization can be represented as a collection of routines, which Nelson and Winter (1982) likened to *memes*. Winter described organizations as, "packets of routinized competencies" (Winter, 1990, p. 280). Similarly, McKelvey (1982) described an organization as a collection of *comps*, the "base units of knowledge and skill that make up what the organization knows how to do" (Romanelli, 1991).

Routines are a good basis for a representation because organizations may share routines yet perform differently because of the environment or interaction of other routines (Winter, 1990, p. 275). As well, routines are combinable in the way required by the GA. For example, the behaviour of the hypothetical cross between a hospital and a coal mine discussed earlier is precisely defined by its collection of routines, half from one and half from the other (e.g., extracting coal, diagnosing and treating it and monitoring its health). Such an organization may not be adaptive, but it is interpretable. A disadvantage is that identifying routines may be problematic; as Romanelli (1991) points out, routines are "empirically elusive" (p. 87).

4.1 Representation

The most direct approach is to apply the GA to a population of organizations, each a collection of routines. Routines might be modelled as rules or productions indicating the appropriate action to be taken in a situation. Rule-based representations seem particularly appropriate for use with the GA because of the claimed robustness of a rule-base to additions or deletions of individual rules (i.e., rules should be easily substitutable for each other). Each organization would therefore be represented as a collection of rules.

These rules might encoded as a bit string, where the bits indicate the presence or absence of a particular rule. These bit strings can then be crossed and evolved as in the example in the previous section. Alternatively, a collection of rules could be represented as a classifier system (Holland, 1992) or even a program that takes the current state of the world as input and outputs the actions to take. Koza (1992) developed techniques for evolving LISP programs, crossing them by exchanging randomly chosen subtrees (i.e., entire subexpressions) from each. Wong and Leung (1995) similarly evolved logic programs, which are essentially collections of rules. These approaches have the advantages that they can develop new rules if a cross occurs in the middle of the representation of a rule.

A second strategy would instead model the subunits within an organization and their interactions—what Carroll (1984) calls the organizational level. Each subunit would be represented as a collection of rules (or even a single rule), as discussed above. In this simulation, the behaviour of the organization and indeed the organization itself would emerge from the interacting behaviours of the individual subunits. In this model, criteria would be needed—perhaps related to patterns of interaction—to determine when an organization has emerged and which subunits are included. As well, it may be desirable to allow subunits to differentiate and develop specializations.

4.2 Fitness

A key difficulty with models based on routines will be calculating fitness. One approach is to simulate the performance of a collection of routines in performing some organizational process. For example, inputs might periodically appear and be processed by the appropriate routines. These routines would in turn produce intermediate goods, which are processed by other routines and so on until a good is delivered to an external customer or the environment. Unfortunately, there is a danger that a simulation of a single organization will be all or nothing: either the necessary routines are present or they are not. As well, developing these simulations will be time consuming (at least at first).

Evolving individual subunits of an organization (the second approach discussed under representation) might avoid at least the first problem. Most organizations (i.e., most populations of subunits) will include all necessary routines, but different patterns of distribution of this know-how will result in differences in performance. However, to evaluate the subunits there must be some mechanism to distribute payoffs among the subunits that contribute to the result, which is essentially the credit assignment problem discussed by Booker et al. (1987, p. 20). The subunit that actually delivers the final good can receive a payoff from the environment, but this payoff must be shared with the other agents that "set the stage" for its success.

The bucket-brigade algorithm developed for classifier systems (Booker, et al., 1987, pp. 20–22) may be an appropriate solution. In a classifier system, each rule has a strength that affects how likely it is to be executed. The strength of a rule is reduced when it executes; it is then increased again if other rules match its output and execute themselves. A rule that produces a desired output is strengthened by the environment. Similarly, a subunit in a simulation might pay those subunits whose output it used in proportion to its success. Subunits that contribute more to an organization's success can then be selected and used to breed the next generation of organizational subunits, thus changing and hopefully improving the entire organization. As well, the distribution of payoffs provides a measure to determine which subunits are part of an emerging organization.

4.3 Proposed Experiment

To illustrate the possible applications of these ideas, I will briefly discuss how they could be used to represent an organizational model such as Malone's (1987) hierarchies and markets. (Again, it seems useful initially to attempt to reproduce findings for which there is some theoretical prediction.) In Malone's model, the problem faced by an organization is processing tasks (e.g., building cars). Tasks arrive at some point in the organization but must be decomposed into subtasks to be processed by specialized processors. Organizations that process more tasks are more successful. Malone compared the performance of four pure forms, namely, functional and product hierarchies and centralized and decentralized markets, each composed of a number of actors of different types.

Each type of actor behaves in a characteristic fashion. Although Malone did not describe them in this way, we can analyze each actor's behaviour as a set of routines for primitive operations and for interacting with other actors, as shown in Table 10. Malone's analysis considered only pure organizational forms and therefore only a limited variety of organizational actors: by mixing these basic capabilities we may be able to generate a wide variety of intermediate forms, such as a cross between a processor and product manager, which performs some tasks on its own and delegates the rest to another actor.

Insert Table 10 about here

In the GA, each actor starts with a random selection of routines and connections to other actors. As well, each actor would have a fixed set of routines for basic interactions, such as "if you want something that you know someone else has, one way to get it is to ask for it." In each generation, the behaviour of the actors is simulated and their interactions determine the performance of the organization. Actors that contribute to the success of the organization, weighted perhaps by some measure of their cost to the organization, are then selected and bred to form the next generation of actors.

Using such a model, the effect of changes in underlying parameters could be assessed. For example, Malone's models included parameters for various costs, such as performing a subtask, maintaining a unit of production capacity and sending a message and he predicted the type of organizational form that would be most efficient for different combinations of the parameters. A GA model can be validated against these predictions as well as used to identify hybrid forms that might be more appropriate for intermediate parameter settings.

4.4 Conclusion

To summarize, I have proposed using a GA on organizations represented as collections of rules for action. In the simplest approach, each organization is represented as a collection of rules. The fitness of an organization is calculated by simulating the performance of those rules on a problem of interest. The GA is applied to a population of such organizations to evolve organizations that are particular good at solving the problem. Alternatively, an organization might be modelled as a population of subunits. The fitness of an subunits is determined by its contribution to the overall success of the organization. The GA is applied to evolve a population of subunits that compose an organization or even multiple organizations.

By substituting a different set of routines, entirely different types of organizations could be modelled. For example, Crowston (in press) modelled the activities performed by participants in engineering change processes and Pentland (1992) modelled the *moves* made by software support hotline specialists.

5 Conclusions

The use of GAs to study Thompson's theory presented above illustrates several possible results when using GAs for studying organizational questions. First, the process of formulating the model required a more explicit statement of the proposition of interest. Second, running the model illuminated the relative balance between the underlying factors. For this simple model, the trade-off might have been directly calculated, but for more complex models, such calculations are likely to be intractable. Finally, the GA can find suitable forms for intermediate values of environmental parameters where the theory makes no or contradictory predictions.

Three objections may be raised to this approach to the study of organizations. First, groups of positions (as in the first example) is a rather limited view of an organization. This is undeniable. Identifying appropriate tasks and organizational features is key to the utility of any kind of model. It should be noted, however, that these choices are not determined by the use of the GA, but rather depend on the organizational theories of interest. Thompson's (1967) proposition is also only about positions and groups, although a natural language presentation provides linkages to other concepts. In principle, any set of interesting features could be used, although the GA does require that they be recombinable in various ways. As the second example suggests, it may be possible to develop quite general models of organizations that can be used with a GA.

Second, as with any method, there are technical issues that must be addressed in applying the GA. The problem representation, the fitness function and the details of the algorithm interact and affect the results found. Therefore, the way the GA is implemented must be carefully chosen to fit the nature of the problem. In the example presented in Section 3, I used the "traditional GA," but that combination of techniques may be inappropriate for many problems (perhaps even for this one). Fortunately, there is a growing body of work that can be used as models (see for example Davis, 1991).

Finally, even if the GA does successfully identify factors that contribute to the performance of an organizational form, it may be that the performance *per se* is only part of the reason for a form's success. For example, based on simulations of the evolution of competing forms, Carroll and Harrison (1992) suggested that long term success of a form may be due as much to chance as actual fitness because of the path dependent nature of the competition. Hannan and Freeman (1986) argued that institutionalization is important for the success of a form: when other powerful actors endorse a particular form's claims for resources or when it becomes unquestioned that one form is the right one to use, the difficulty of starting an organization and mobilizing resources is greatly reduced. Therefore, organizations may adopt forms without regard to their inherent performance; indeed, as Stinchcombe (1965) noted, "forms tend to incorporate and retain packages of characteristics that were fashionable or legitimate in the period when the form takes shape" (p. 53). In the terms of my example, the arrangement of positions into groups may be done for historical reasons rather than to meet the demands of the current task structure. Finally, organizational forms may differ in ways beyond those used to calculate fitness, for example, in the quality of work life they provide for their employees, how much position holders like each other and want to be in the same groups, etc.

These caveats are certainly significant for empirical studies that attempt to explain an observed distribution of forms and such factors will certainly affect the final implementation of novel designs. For the task of suggesting new forms, however, these objections are far less damaging. Indeed, computerized implementations of organizational evolution are best seen as powerful tools for the imagination, used to help conceive novel organizational forms.

References

Axelrod, R. (1987). The evolution of strategies in the iterated prisoner's dilemma. In L. Davis (Eds.), *Genetic Algorithms and Simulated Annealing*. London: Pitman.

Bean, J. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, *6*(2), 154–160.

Booker, L. B., Goldberg, D. E. and Holland, J. H. (1987). *Classifier Systems and Genetic Algorithms* (Technical Report No. 8). CSMIL, University of Michigan.

Burton, R. M. and Obel, B. (1980). A computer simulation test of the M-form hypothesis. *Administrative Science Quarterly*, *25*, 457–66.

Carroll, G. R. (1984). Organizational ecology. *Annual Review of Sociology*, *10*, 71–93.

Carroll, G. R. and Harrison, J. R. (1992). *Chance and Rationality in Organizational Evolution* (Unpublished manuscript). University of California at Berkeley.

Crowston, K. (in press). A coordination theory approach to organizational process design. *Organization Science*.

Davis, L. (Eds.). (1991). *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold.

DiMaggio, P. and Powell, W. (1983). Institutional isomorphism. *American Sociological Review*, *48*, 147–60.

Freeman, J. and Hannan, M. T. (1983). Niche width and the dynamics of organizational change. *American Journal of Sociology*, *88*, 116–45.

Friedman, L. W. and Friedman, H. H. (1995). Analyzing simulation output using the bootstrap method. *Simulation*, *64*(2), 95–100.

Grefenstette, J. J. (1991). Strategy acquisition with genetic algorithms. In L. Davis (Eds.), *Handbook of Genetic Algorithms* (pp. 186–201). New York: Van Nostrand Reinhold.

Hannan, M. T. and Freeman, J. (1986). Where do organizational forms come from? *Sociological Forum*, *1*(1), 50–72.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems* (2nd ed.). Ann Arbor, MI: University of Michigan.

Koza, J. R. (1992). *Genetic Programming*. Cambridge, MA: MIT Press.

Liepins, G. E. and Potter, W. D. (1991). A genetic algorithm approach to multiple-fault diagnosis. In L. Davis (Eds.), *Handbook of Genetic Algorithms* (pp. 237–250). New York: Van Nostrand Reinhold.

Malone, T. W. (1987). Modeling coordination in organizations and markets. *Management Science*, *33*, 1317–1332.

McKelvey, B. (1982). *Organizational Systematics: Taxonomy, Evolution, Classification.* Berkeley: University of California.

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer-Verlag.

Nelson, R. R. and Winter, S. G. (1982). *An Evolutionary Theory of Economic Change*. Cambridge, MA: Harvard.

Pentland, B. T. (1992). Organizing moves in software support hotlines. *Administrative Science Quarterly*, *37*, 527–548.

Romanelli, E. (1991). The evolution of new organizational forms. *Annual Review of Sociology*, *17*, 79–103.

Singh, J. V. and Lumsden, C. J. (1990). Theory and research in organizational ecology. *Annual Review of Sociology*, *17*, 161–193.

Stinchcombe, A. L. (1965). Social structure and organizations. In J. G. March (Eds.), *Handbook of Organizations* (pp. 153–193). Chicago: Rand McNally.

Thompson, J. D. (1967). *Organizations in Action: Social Science Bases of Administrative Theory*. New York: McGraw-Hill.

Williamson, O. E. (1975). *Markets and Hierarchies*. New York: Free Press.

Winter, S. G. (1990). Survival, selection and inheritance in evolutionary theories of organizations. In J. V. Singh (Eds.), *Organizational Evolution: New Directions* (pp. 269–297). Newbury Park: Sage.

Wong, M. L. and Leung, K. S. (1995). Inducing logic programs with genetic algorithms: The Genetic Logic Programming Systems. *IEEE Expert*, *10*(5), 68–76.

Tables

Table 1. Parameters of the simulation and settings in the reported experiments.

Parameter	*Setting in reported experiments*
Generations	200
Population	1000
Number of positions	10
Number of groups	16
Number of tasks/position	10 selected (see text)
Interdependency between positions	0, 1, 3, 5, 7, 8, 10 tasks overlap (see text)
Coordination cost	$^1/_9$ (see text)
Coordination benefit	–.5, 0, .3, .7, 1.2, 1.7, 2.25 (see text)
Mutation rate	$^1/_{40}$ % chance of a mutation per bit
Maximum and minimum normalized fitnesses	100 and 1
Number of runs	30

Table 2. Example of positions and tasks with 5 tasks assigned deterministically.

```
         Assigned task
Position
       | 1111111111222222222233333333334444444444
       | 012345678901234567890123456789012345678901234567
       | 89
-------+-------------------------------------------------
       | 11111    1    1        1 1            1
   1   |    1 11111 1               1       1        1
   2   |           11111   1 1   1       1     1
   3   |      1 11      11111                        1 1
   4   | 1  1     1          11111                1
       | 1
   5   |         1      1        111111         1       1
   6   |        1         1            11111      1   1 1
   7   |  1             1      1       1    111111
   8   |                         11 1    1      111111
   9   | 1   11  1                          1
       | 11111
```

Note: a 1 indicates the task is performed by the position; assigned tasks are underlined.

Table 3. Overlap between actors' assigned tasks (# shared tasks/total # tasks) for positions in Table 2. (For clarity, only the upper diagonal is shown.)

Position	0	1	2	3	4	5	6	7	8	9
0	1.0	0.2	0.2	0.0	0.4	0.1	0.0	0.2	0.1	0.2
1		1.0	0.1	0.4	0.2	0.2	0.1	0.1	0.2	0.3
2			1.0	0.1	0.2	0.1	0.1	0.1	0.2	0.0
3				1.0	0.0	0.2	0.2	0.1	0.1	0.3
4					1.0	0.1	0.1	0.1	0.2	0.2
5						1.0	0.1	0.2	0.4	0.2

6						1.0	0.1	0.2	0.2
7							1.0	0.2	0.1
8								1.0	0.0
9									1.0

Table 4. Summary of conditions, hypothesized forces and expected organizational forms.

		Coordination benefit		
		Negative or zero	*Low*	*High*
Inter-dependence	*Favoured result*	Many groups	Few groups	One group
Zero	Many groups	Many groups	Many groups	Many groups
Medium	Few groups	Many groups	Few groups	One group
High	One group	Many groups	Few groups	One group

Table 5. Example final organization form (5 task overlap, no coordination benefit).

Position 0, Group: 6
Position 1, Group: 0
Position 2, Group: 13
Position 3, Group: 4
Position 4, Group: 9
Position 5, Group: 15
Position 6, Group: 5
Position 7, Group: 12
Position 8, Group: 10
Position 9, Group: 2

Table 6. Average number of groups by interdependency and coordination benefit.

Interdependency		*Penalty* -.50	*No effect* .00	*Coordination benefit* *Low* .30	.70	1.20	*High* 1.70	2.25
Zero	0	10.00	10.00	10.00	10.00	10.00	10.00	10.00
	1	10.00	10.00	10.00	9.47	6.93	5.93	4.93
	3	10.00	10.00	9.93	6.23	4.17	3.23	2.80
	5	10.00	10.00	8.57	3.70	2.13	2.10	2.03
	7	10.00	10.00	5.33	2.00	2.07	2.00	1.97
	8	10.00	10.00	2.23	2.13	2.23	2.23	1.97
Total	10	10.00	10.00	2.43	2.33	2.23	2.23	2.03

Table 7. Average size of largest group by interdependency and coordination benefit.

Interdependency		*Penalty* -.50	*No effect* .00	*Coordination benefit* *Low* .30	.70	1.20	*High* 1.70	2.25
Zero	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	1	1.00	1.00	1.00	1.47	2.27	2.50	2.93
	3	1.00	1.00	1.07	2.33	3.90	4.67	5.23
	5	1.00	1.00	1.83	4.23	6.57	6.77	7.20
	7	1.00	1.00	3.33	7.60	7.07	7.23	7.17
	8	1.00	1.00	6.90	7.20	6.67	6.47	7.03
Total	10	1.00	1.00	5.73	6.50	6.73	6.87	7.60

Table 8. Average correlation between position interdependency and group assignment for intermediate conditions.

	Coordination benefit				
	Low			*High*	
Interdependency	.30	.70	1.20	1.70	2.25
1				.69	.74
3		.63	.72	.72	.73
5	.38	.65	.75		
7	.59				

Table 9. Significance of minimum correlation found.

	Coordination benefit				
	Low			*High*	
Interdependency	.30	.70	1.20	1.70	2.25
1				.96	.99
3		.97	.99	.99	1.00
5	.25	.99	.99		
7	1.00				

Table 10. Capabilities of different actor types in Malone's (1987) model organizations.

Actor type	*Capabilities and knowledge*
Processor	Perform assigned subtasks
	Respond to bids in a market
Product managers	Decompose tasks into subtasks
	Know one processor for each type of subtask
	Communicate with processors to assign subtasks
	Integrate results of subtasks
Functional manager	Know multiple processors for one type of subtask
	Pick best processor for a given subtask
	Communicate with processors to assign subtasks
General manager	Decompose tasks into subtasks
	Know one functional manager for each type of subtask
	Communicate with functional manager to assign subtasks
	Integrate results of subtasks
Buyers in a decentralized market	Decompose tasks into subtasks
	Know multiple processors for each type of subtask
	Request bids for each type of subtask
	Evaluate bids to pick best processor for a given subtask
	Communicate with processors to assign subtasks
	Integrate results of subtasks
Buyers in a centralized market	Decompose tasks into subtasks
	Know one middleman for each type of subtask
	Communicate with middlemen to assign subtasks
	Integrate results of subtasks
Middlemen in a centralized market	Know multiple processors for one type of subtask
	Request bids for one type of subtask
	Evaluate bids to pick best processor for a given subtask
	Communicate with processors to assign subtasks