Proceedings of the 2007 IEEE Symposium on
Artificial Life (CI-ALife 2007)

Speciation through Exaptation

Lee Graham, Franz Oppacher
{lee@stellaralchemy.com, oppacher@scs.carleton.ca}
School of Computer Science, Carleton University
Tel: 1-613-520-4333, Fax: 1-613-520-4334
1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada

Abstract- **In this paper we present a simple genetic algorithm, possessing a notion of viability, in which a population of individuals in one niche spreads to populate a second niche with a different fitness function. The two niche populations then diverge genetically to such a degree that offspring produced by the crossover operator are inviable in both niches. We argue that this simple adaptive evolutionary behavior in the genetic algorithm can be likened to the concept of speciation in biology.**

OVERVIEW

The fields of evolutionary computation (EC) and artificial life (ALife) are replete with attempts to describe, and take advantage of a wide variety of evolutionary behaviors and dynamics in digital systems of Darwinian evolution. These include such phenomena as the Baldwin effect [1, 2], arms races [3], mimicry [4], parasitism [5, 6], co-evolution [7], and exaptation [8] among others. In this paper we describe experiments with a model of speciation that relies on the exaptation of evolved structures, a distinction between viable and inviable individuals, and is based on a biological notion of species. This contrasts with the more common notion of species and speciation in GAs, in which the distinction between viable and inviable interspecies hybrids is not made[1], and in which individual species are merely subpopulations exploring different regions of a common fitness landscape.

The concept of a species and the processes of speciation are central ideas in the field of evolutionary biology, and are vital to our understanding of the abundant diversity of life forms on Earth. In the biological sciences speciation is said to occur when two populations of organisms, having originated from a common ancestral stock, diverge from one another either genetically or behaviorally to such a degree that members of one population either cannot or do not interbreed with members of the other population. Once speciation has occurred the two populations are said to be reproductively isolated, and continue to diverge from one another via genetic drift, selection, or both, increasing the diversity of life. The process of speciation is thought to be triggered either by the geographical isolation of a group of organisms from the original stock, in which case the phenomenon is called allopatric speciation, or by behavioral changes that prevent interbreeding despite both lineages living in the same location. This second alternative is referred to as sympatric speciation. Numerous instances of speciation have been observed both in the lab and in the wild, and are documented extensively in the biological literature [9].

Just as the biological phenomenon of speciation is a source of diversity and innovation in the biosphere, it may be that a comparable notion of speciation, which is mostly lacking in extant EC systems, will yield benefits. As one population speciates, the system in question applies previously learned or evolved knowledge and structures to newer and possibly more difficult problems. An ability to harness behavior of this sort could conceivably prove to be a useful addition to the EC design and optimization toolkit.

The genetic algorithm (GA) described in the next section consists of two populations of individuals, called “niches”. Each niche has its own fitness function by which individuals are evaluated and compared. The fitness functions are such that the standard GA practice of random population initialization will tend to produce viable solutions in the first niche but not the second. The second niche population remains effectively uninitialized until individuals from the first niche population are able to jump to the second and become founders of a new population of viable individuals.

The authors examine the data collected from thousands of runs of the GA and provide a brief analysis. These show that once a viable population of individuals begins to evolve in the second niche the two populations rapidly diverge to the point of effective speciation – hybrid offspring produced by the crossover operator are inviable under the fitness functions of both niches.

THE GENETIC ALGORITHM

The GA used in our speciation experiments operates in a fairly standard manner as far as GAs are concerned, but with some minor noteworthy differences. The fitness functions for both populations (or “niches”) are maximization problems. Both map GA individuals to non-negative integer values and both have a minimum attainable fitness of zero. In this GA individuals with zero fitness are considered inviable solutions and are forbidden from reproducing. In this respect the GA operates more like many ALife systems with their endogenous fitness imposing a distinction between viable and inviable individuals. A second noteworthy difference is that, as will be seen in the next section, one of the fitness functions (that of the first niche) will appear to rapidly converge and level off,

---

[1] It should be noted that the notion of viable and inviable individuals comes into play in some GAs in which certain regions of the search space are considered infeasible (such as repeating cities in a candidate TSP solution), and in some GAs with zero-fitness individuals under fitness-proportional selection regimes.

1-4244-0701-X/07/$20.00 ©2007 IEEE

yet the population will continue to undergo significant evolutionary change driven by selection pressure (as opposed to mere pseudo-clone genetic drift as would be the case in many GAs after fitness convergence).

GA Parameters

The same GA parameters and operators are applied to both niches and are as follows. The population size is 50 with single individual elitism. Selection is done by tournaments of size two, and only viable individuals are eligible for selection. Ten percent of offspring are produced by uniform crossover, with the rest produced by mutating a copy of a single chosen parent. Chromosome length is variable and is modified by mutation and crossover operators. Since the experiments described herein are concerned with the high-level speciation-like behavior of the GA, and not with the efficient generation of highly fit solutions, as would be the case with most GAs, no effort has been made to tune these parameters.

Representation

Both niches' fitness functions assign a non-negative integer value to individuals' phenotypes. Those phenotypes are two-dimensional structures constructed within a grid of 30 rows and 14 columns, called the "game board". Each grid cell in the game board is blank by default unless the individual's genes specify otherwise. Nonblank grid cells can have one of four types of "boxes" placed in them, designated by the letters S, A, L, and R. The meaning of each box type depends on which fitness function is being evaluated. Each of the two fitness functions is explained below under the "Fitness Problem One" and "Fitness Problem Two" sections.

Individuals' chromosomes are composed of three strands of equal length. This length can vary between individuals but not between strands in the same individual. Strand one is a sequence of game board row indices with values ranging from 0 to 29 since the game board has 30 rows. Strand two is a sequence of game board column indices with values ranging from 0 to 13 since the game board has 14 columns. The third strand is a sequence of letters (either S, A, L, or R) representing box types. The phenotype is decoded from the genotype by simply reading the three strands in lockstep from left to right and placing boxes of the specified types in the specified cells on the game board. If the same cell is specified multiple times only the final box assignment will dictate the type of box that ends up in the cell. This decoding creates a two-dimensional structure on the game board which is then examined for fitness evaluation.

Chromosome length is variable from individual to individual but is constrained to have a minimum length of one and a maximum length of 400.

Crossover Operator

The crossover operator employed is a simple uniform crossover for variable length chromosomes. Each row-column-box assignment in the chromosome has a 50% chance of coming from parent one and a 50% chance of coming from parent two. In the case of unequal parental chromosome lengths, row-column-box assignments in the longer parent that have no counterparts in the shorter parent are appended to the offspring's chromosome 50% of the time, at random.

Mutation Operators

When offspring are produced by mutation, one of three mutation operators is chosen with uniform probability. The first is a point-mutation operator which chooses a locus and a strand from the chromosome uniformly at random and replaces it with a different value from the range of legal values for the strand. The second is a deletion mutation operator which chooses a chromosome locus at random and deletes it from all three strands thereby shortening chromosome length by one. The third is an insertion mutation operator which chooses a chromosome insertion point at random and inserts a new value into each of the three strands thereby increasing the chromosome length by one. If a mutation operator would result in a violation of the chromosome length constraints, one of the remaining two operators is chosen uniformly at random instead.

Fitness Problem One: A Ball Game (BG)

Fitness in the first niche is calculated by interpreting the two-dimensional phenotype on the game board in the following manner. A game ball falls through the board, entering from above at column index 5. The ball carries with it a point value which is initially set to 15 points. As the ball falls downward through the game board from row 0 to row 29, it is manipulated by any boxes it encounters. The four different types of boxes can modify the number of points assigned to the game ball, its direction of travel through the board, and even the number of balls in play. The manner in which each type of box affects game balls that pass through it is explained below. Any game balls emerging from column 8 at the bottom of the game board contribute to a total tally of points collected during the game. The goal of the game, then, is to steer and multiply the game ball(s) on the game board in such a way as to funnel as many high-point-value balls out of the bottom of column 8 as possible. This point tally contributes to the BG niche fitness value according to the fitness function

$$\text{fitness} = \max\{\, 0, p - (c * P)\,\}, \qquad (1)$$

where p is the number of points collected, c is the chromosome length, and P is a penalty value which increases during a run in order to make the game more difficult as the population evolves[2]. The value of P in the niche begins at zero and is increased at the end of every generation by the minimum (integer) amount required to keep the population's maximum fitness value at or below a fixed threshold of 1000. An example of a highly evolved individual in this niche, from the final generation of a run, is shown on the left side of Fig. 2.

---

[2] The rationale for P stems from its effect on certain aspects of evolved solution complexity for this problem, which is the subject of other research by the authors.

The Effect of Each Box Type on the Game Ball

The four types of boxes specified by the third strand of an individual's chromosome, designated by the letters S, A, L and R, affect the game ball in the following ways. The Split Box (or S-Box) splits the game ball into two balls, each with a point value one less than the original. The two reduced-point-value balls exit the Split Box in different directions. The second type of box, called the Add Box (or A-Box) allows the game ball to pass through and exit traveling in the same direction in which it entered. At the same time the point value assigned to the ball is increased by one point. The third and fourth box types, called Left Boxes and Right Boxes (or L-Boxes and R-Boxes) modify only the ball's direction of travel. The Left Box will cause the ball to make a left turn, as seen from the ball's point of view. The Right Box will cause the ball to make a right turn. In the cases in which such turns would cause the ball to exit the boxes traveling upwards, the ball exits traveling downward instead. The upward direction is forbidden in order to ensure that the time required for game play is finite. When a game ball enters a blank cell on the board, it always exits traveling downward. The rules for the S-, A-, L-, and R-Boxes are depicted in Fig. 1.

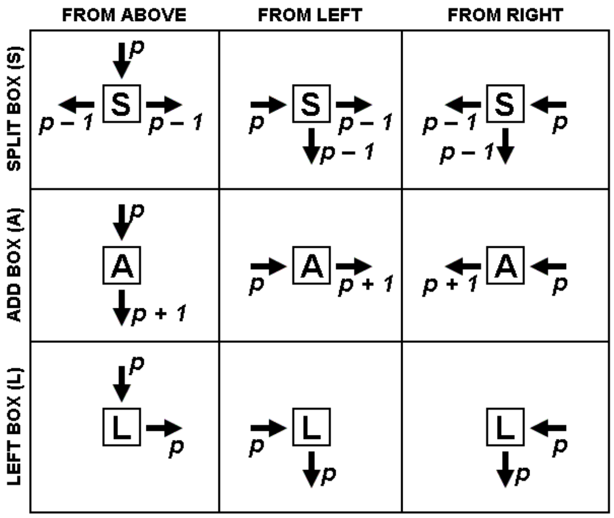

Fig. 1. Rules governing the passage of the game ball(s) through S-, A-, and L-Boxes (R-Box rules are a mirror image of L-Box). Labeled arrows indicate game balls entering and exiting the boxes carrying particular point values.

Fitness Problem Two: Largest Enclosed Area (LEA)

The fitness function in the second niche makes no distinction between the four different types of boxes placed on the game board. To attain nonzero fitness in the LEA niche, the two-dimensional structure created by an individual's phenotype must create an unbroken path of boxes from the top of the board (row 0) to the bottom (row 29). Such a path can be established between successive neighbors in the Moore Neighborhood (neighbors in any of the eight cells surrounding a box). Provided such a path exists, fitness is then equal to the largest contiguous region of blank cells which is completely enclosed by a circuit of boxes. Like the top-to-bottom path, this circuit follows the von Neumann Neighborhood of a box at each step. If an individual's phenotype does not possess a path connecting the top and bottom of the board, or does not possess a circuit enclosing an empty region consisting of at least two blank cells, the individual is assigned zero fitness and is considered inviable. An example of a highly evolved individual from the final generation of a run is shown on the right side of Fig. 2.

The LEA fitness function operates in such a way that randomly generated individuals in the first generation are inviable with extremely high probability. The LEA niche only becomes populated when speciation occurs in the GA and structures evolved in the BG niche become viable in the new niche. All randomly generated individuals, whether in the BG niche or the LEA niche, place their boxes only on a restricted region of the board (explained in the next section on population initialization). This prevents a top-to-bottom path from ever appearing in the first generation and renders all first-generation individuals inviable in the second niche. Nevertheless, even without this restriction, viable individuals are almost never produced at random for the second niche. To test the claim that this restriction does not materially change this property of the niche with regards to the inviability of random individuals, the authors generated 8,000,000,000 random individuals without the box placement restriction and not one was viable in the LEA niche.

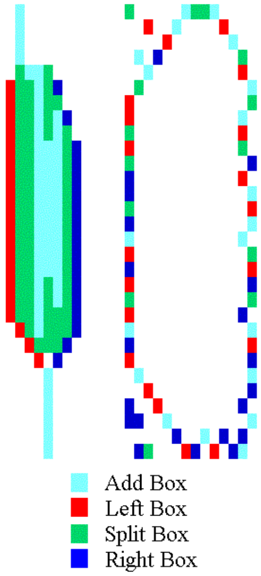

Fig. 2. Two examples of highly evolved individuals from the final generation of a run ; on the left is the phenotype of an individual from the BG niche, and on the right from the LEA niche.

Population Initialization

Since the population sizes in these experiments are quite small at 50 individuals per niche, population initialization has been biased in two ways in order to start each run off with a better sample of random individuals. The first bias, as was mentioned in the previous section, is a row and column restriction on the placement of boxes for randomly generated individuals. Randomly generated individuals' chromosomes restrict box placement to rows 2 through 15, inclusive, and to columns 3 through 10, inclusive. This boosts the chances of random individuals collecting points in the ball game. Once a GA run begins, mutation can immediately violate this restriction. Unlike the LEA niche, random individuals in the BG niche are sometimes viable. Since inviable individuals are prevented from reproducing, this bias during initialization only ensures that nearly all runs will have at least one viable individual in the first generation in niche one instead of terminating immediately for lack of individuals from which to produce successive generations.

The second bias in population initialization is that every individual in the first generation is the best of ten randomly generated individuals. This was done for the same reason as was mentioned above, and applies to both niches. It is effectively of no benefit to the second niche, however, given the extremely low probability of generating viable individuals at random.

Pioneer Jumps

Since the LEA niche remains effectively empty after population initialization, some means of allowing the niche to become populated at a later time is required. This is accomplished by choosing a random viable individual from the BG niche at the end of every generation and evaluating that individual's fitness in the LEA niche. If the individual, called a "pioneer", is found to be not only viable in the new niche, but also more fit than the least fit viable individual already in the niche, then a pioneer "jump" event takes place, and a copy of the pioneer is placed in the new niche, replacing the least fit individual. In this manner, structures evolved in the BG niche can be exapted (co-opted for a function other than the one(s) for which selection adapted them) to allow individuals to survive in a new niche with new selection pressures. After the exaptation event(s), the newly-founded population in the second niche can undergo further adaptation in an entirely new direction. The two populations then diverge genetically to the point where hybrid offspring, produced by the crossover of a parent from each niche, are inviable in both niches. When this occurs, the initial population, which started in the BG niche, has speciated into two populations which are unable to interbreed. This GA setup is depicted in Fig. 3.

EXPERIMENTAL RESULTS

The authors performed 7500 runs of the speciation GA with the parameters and configuration described in the previous section. Each run lasted for a duration of 5000 generations. At the end of every twentieth generation the following statistics were logged.

The fitness of the best individual in each niche

The number of viable individuals in the LEA niche (between 0 and 50 individuals)

The number of successful pioneer jumps during the 20 generation interval (between 0 and 20)

The outcome of a crossover operator hybridization test between the best individual from the first niche and the best individual from the second niche (three possible outcomes, described below)

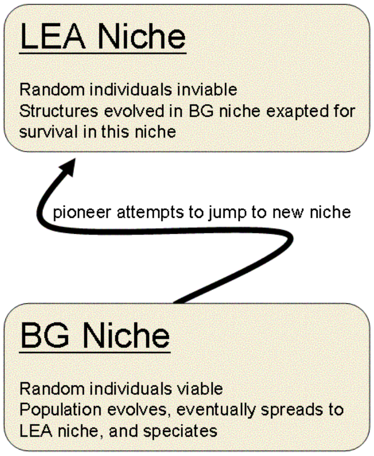

Fig. 3. GA niche configuration. Pioneers from the BG niche may jump to the LEA niche and, if found to be viable and competitive there, inhabit the new niche, possibly becoming founders of a new population.

In the case of the hybridization tests, the first possible outcome occurs when at least one of the two niches contains no viable individuals to take part in the crossover and no hybrid offspring can be generated. The second possible outcome occurs when the hybrid offspring is viable (has nonzero fitness) in at least one of the two niches. The third possible outcome occurs when the hybrid is inviable in both niches.

Fig. 4 shows the change over time in the average fitness value of the best individual in the BG niche during a run. The fitness threshold and chromosome length penalty create a fitness graph with negative slope in some regions, which is unusual for a GA employing elitism. The graph is somewhat deceptive, however, since the number of points collected by individuals in the board game continues to increase (exponentially) throughout the run, as reflected by the chromosome length penalty graph in Fig. 5. This is analogous to van Valen's Red Queen Effect [10] in biology, with the adjustments in chromosome length penalty playing the role of the second party in an arms race.

Fig. 6 and Fig. 7 show, for the LEA niche, the number of viable individuals and the average best fitness, respectively. Both of these graphs attest to the fact that the niche population is an offshoot of the BG niche population. The initial period (hundreds of generations on average) devoid of viable individuals at the beginning of the runs indicates that the final population in the niche cannot have evolved from the initial random stock since inviable individuals cannot reproduce. This is evident in the graph despite the fact that Fig. 6 averages over multiple runs in which pioneer jumps take place at different times or in some cases not at all. This has the effect of exaggerating variance during the period in which pioneer jumping takes place, and yet there is still a period of calm in the graph which is common to all the runs regardless of when the first pioneer jumps. It is during this initial period of calm that a co-optable structure is being evolved in the BG niche, which can then be used to seed the new niche.

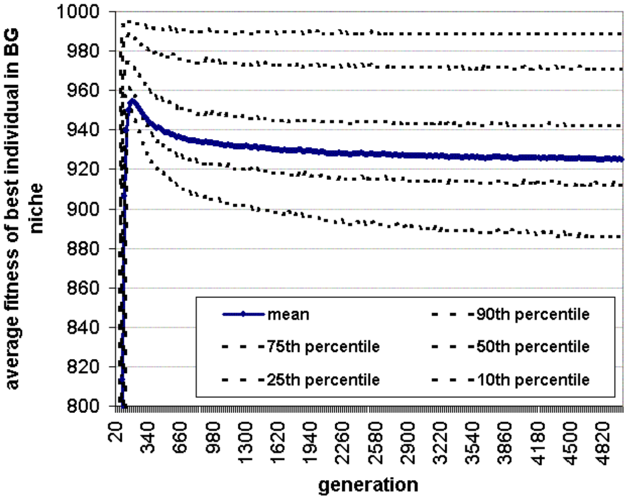

Fig. 4. Mean best fitness over time in the BG niche, averaged over 7500 runs, together with several percentile graphs.

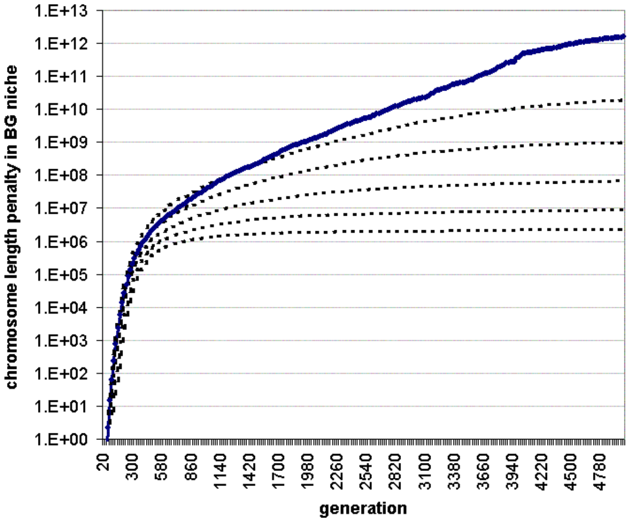

Fig. 5. Chromosome length penalty value in the BG niche over time, averaged over 7500 runs, together with several percentile graphs. Due to exponential growth, the mean is greater than the 90^{th} percentile, and the y-axis is scaled logarithmically.

Fig. 8 shows the behavior of the GA with respect to pioneers jumping from the first niche to the second. The sloped-spike shape of the graph reflects the fact that such pioneer jumps are made during the first half of the run, after the initial period of calm in which a co-optable structure is being evolved[3], and that, because of secondary adaptations in the new niche, the chance of another successful jump drops rapidly to a value near zero. The histogram inside Fig. 8 shows that, of those runs in which jumps occurred at all, often only a single individual seeded the new population, with subsequent potential pioneers having been unable to compete against the new population's rapid adaptation to the demands of the new fitness function.

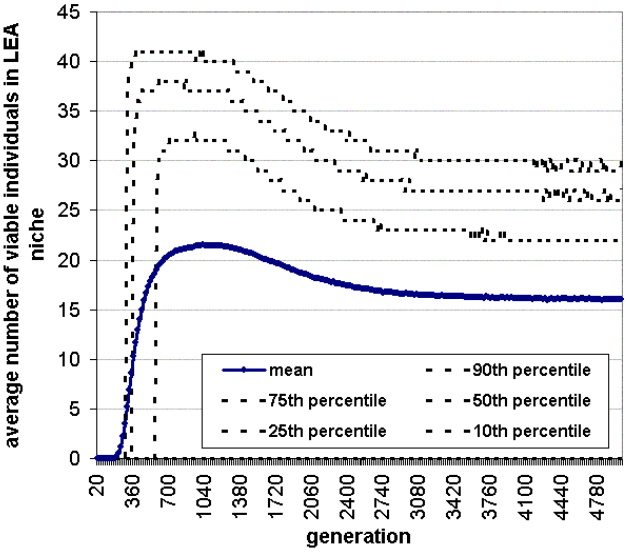

Fig. 6. Number of viable individuals in the LEA niche, averaged over 7500 runs, together with several percentile graphs which show that at least 25% of the runs never create a viable individual in this niche.

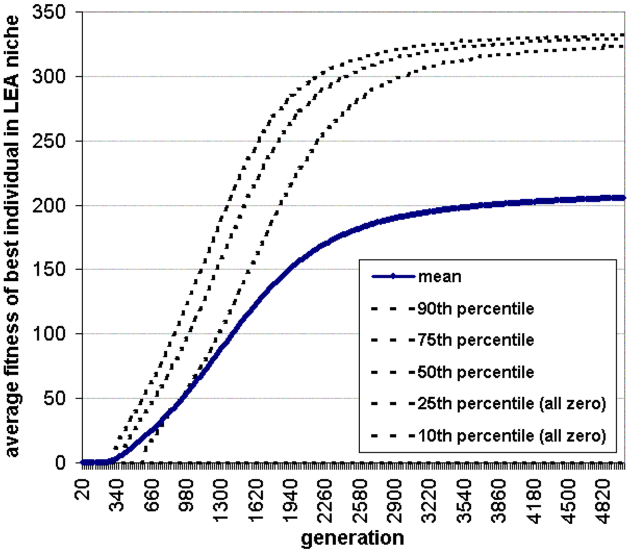

Fig. 7. Best fitness value in the LEA niche, averaged over 7500 runs, together with several percentile graphs which, like those in Fig. 6, show that at least 25% of the runs never create a viable individual in this niche.

---

[3] This ability to co-opt evolved systems from one domain for use in another reveals relationships between domains that would otherwise, it seems, go entirely unnoticed and unanticipated. The authors posit that these sorts of counter-intuitive connections between differing domains may be an important resource in future efforts to produce flexible evolutionary systems capable of producing perpetual and clever novelty.

Finally, Fig. 9 shows the results of the hybridization tests. The graphs show unambiguously that the offshoot population in the second niche becomes unable to interbreed with the original population in the first niche. This divergence is so rapid, in fact, that at the twenty-generation logging interval resolution of the graph, the hybridization outcomes in which the offspring are viable in at least one niche amount to only a small bump at the bottom of the graph.

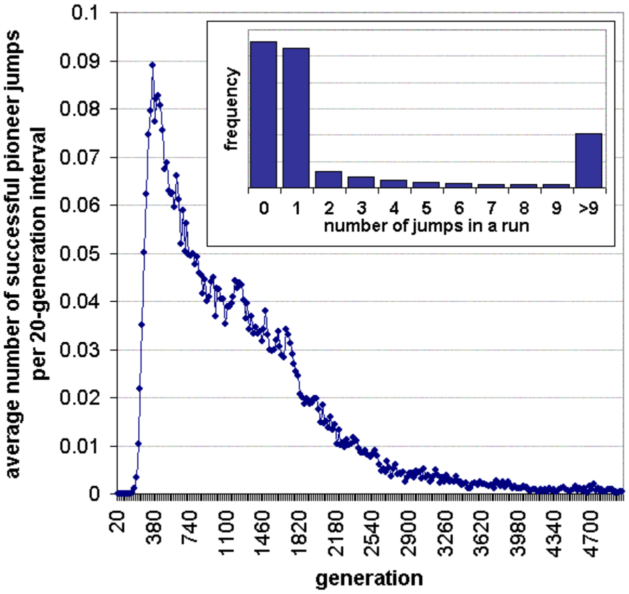

Fig. 8. Timing and frequency of successful pioneer jumps from the BG niche to the LEA niche in the GA, averaged over 7500 runs. The histogram at upper right shows the distribution in the number of such jumps during an entire run.

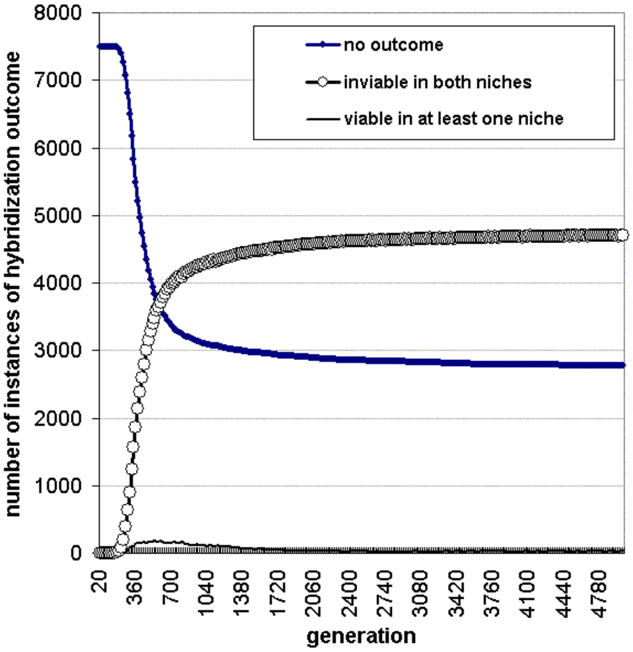

Fig. 9. Number of hybridization test outcomes over time in the GA, summed over 7500 runs. A "no outcome" result occurs when at least one of the two niches contains no viable individual to participate in the crossover operation.

CONCLUSIONS

Whereas the term "species" in GAs has often come to refer to isolated subpopulations diverging from one another but working toward the same goal in roughly the same fitness landscape, such as the creation of a highly-fit solution to a combinatorial optimization problem, the term is very often divorced from its biological counterpart's focus on interbreeding (or lack of ability to do so) between the two populations. Many researchers employing speciation in GAs do so in order to place greater emphasis on the exploration side of the exploration-vs-exploitation tradeoff inherent in such algorithms. The issue of barriers to interbreeding in a GA often arises from population isolation, and not from an inability to produce viable offspring. In many GAs any individual may cross with any other individual (although in some cases they may simply be forbidden from doing so) and from this point of view all individuals in all subpopulations belong to a single species. For these reasons, and because many GAs make no distinction between viable and inviable offspring, the terms "species" and "speciation" have only a tenuous connection to their counterparts in biology.

By introducing a notion of viability, and by allowing individuals with the same underlying genetic coding format to evolve under two distinct and very different fitness functions, we allow an original interbreeding population to split and diverge to the point where crossover can no longer produce viable offspring. In doing so the GA is able to make progress in a new niche which would otherwise have remained filled with inviable solutions.[4] Although simple in design, and very rudimentary, this GA reliably exhibits a speciation-like phenomenon, achieved through exaptation, which more closely adheres to the meaning of the term as it is used in the biological sciences.

REFERENCES

[1] Ackley, D., & Littman, M. (1992). "Interactions Between Learning and Evolution," in Langton, C.G., Taylor, C., Farmer, J.D., & Rasmussen, S. eds, *Artificial Life II*, Addison-Wesley.

[2] Hinton, G.E., & Nowlan, SJ. (1987). "How Learning Can Guide Evolution," *Complex Systems 1*, 495-502.

[3] Ficici, S., & Pollack, J.B. (1998). "Challenges in Coevolutionary Learning: Arms-Race Dynamics, Open-Endedness, and Mediocre Stable States," in Adami, Belew, Kitano, & Taylor, eds, *Proceedings of the Sixth International Conference on Artificial Life*. The MIT Press.

---

[4] Although this inviability of random individuals in the second niche is central to the speciation exhibited here, it is not true that this need necessarily be the case for this particular toy problem. Relaxation of restrictions on chromosome length and box placement during population initialization may yield viable random solutions at a workable rate in this instance. There may be some problems, however, where the random generation of viable or feasible solutions is a challenging problem in its own right, and for which some mechanism of exaptation may be an alternative avenue of exploration for the would-be EC practitioner.

[4] Franks, D.W., & Noble, J. (2002). “The Origins of Mimicry Rings,” in *Proceedings of the Eighth International Conference on Artificial Life*. 186-191. The MIT Press, Cambridge, MA, USA.

[5] Hillis, D. (1990). “Co-evolving Parasites Improve Simulated Evolution as an Optimization Procedure,” *Physica D, 42*, 228-234.

[6] Robbins, P. (1994). “The Effect of Parasitism on the Evolution of a Communication Protocol in an Artificial Life Simulation,” in Cliff, D., Husbands, P., Meyer, J.-A., & Wilson, S.W. eds. *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*. MIT Press/Bradford Books.

[7] Rosin, C.D., & Belew, R.K. (1995). “Methods for Competitive Co-evolution: Finding Opponents Worth Beating,” in *Proceedings of the Sixth International Conference on Genetic Algorithms*. 373-380. Morgan Kaufmann, San Francisco, CA.

[8] de Oliveira, P.P.B. (1994). “Simulation of Exaptive Behaviour,” *Parallel Problem Solving from Nature, 3, Lecture Notes in Computer Science 866*, 354-364. Springer Verlag, Berlin, Germany.

[9] Boxhorn, J. (1995). “Observed Instances of Speciation,” in *The TalkOrigins Archive*. www.talkorigins.org/faqs/faq-speciation.html

[10] van Valen, L. (1973). “A New Evolutionary Law,” *Evolutionary Theory 1*:1-30