

# Selection Procedures for Module Discovery: Exploring Evolutionary Algorithms for Cognitive Science

**Janet Wiles** (*j.wiles@csee.uq.edu.au*)

**Ruth Schulz** (*ruth@csee.uq.edu.au*)

**Scott Bolland** (*scottb@csee.uq.edu.au*)

**Bradley Tonkes** (*bradley@csee.uq.edu.au*)

**Jennifer Hallinan** (*J.Hallinan@imb.uq.edu.au*)

The University of Queensland, Brisbane QLD 4072 Australia

## Abstract

Evolutionary algorithms are playing an increasingly important role as search methods in cognitive science domains. In this study, methodological issues in the use of evolutionary algorithms were investigated via simulations in which procedures were systematically varied to modify the selection pressures on populations of evolving agents. Traditional roulette wheel, tournament, and variations of these selection algorithms were compared on the “needle-in-a-haystack” problem developed by Hinton and Nowlan in their 1987 study of the Baldwin effect. The task is an important one for cognitive science, as it demonstrates the power of learning as a local search technique in smoothing a fitness landscape that lacks gradient information. One aspect that has continued to foster interest in the problem is the observation of residual learning ability in simulated populations even after long periods of time.

Effective evolutionary algorithms balance their search effort between broad exploration of the search space and in-depth exploitation of promising solutions already found. Issues discussed include the differential effects of rank and proportional selection, the tradeoff between migration of populations towards good solutions and maintenance of diversity, and the development of measures that illustrate how each selection algorithm affects the search process over generations. We show that both roulette wheel and tournament algorithms can be modified to appropriately balance search between exploration and exploitation, and effectively eliminate residual learning in this problem.

## Introduction: EC and Cognitive Science

Evolutionary computation (EC) is increasingly used in cognitive science, both for evolving cognitive models and for modeling evolutionary processes.

Many algorithms use evolutionary search in one form or another. No single search algorithm will be optimal for all tasks (a thesis colloquially known as “no free lunch”, Wolpert & Macready, 1996). In any simulation study, characteristics of the task need to be taken into account in the selection of algorithms. However, to many cognitive science researchers it is not clear which aspects of tasks are important in the design of a search process, and what properties of evolutionary search

algorithms need to be taken into account to select an appropriate design.

This study is part of a wider program of research whose goal is to enhance the effective use of evolutionary computation techniques in cognitive science domains. This program involves assessing the performance of popular evolutionary algorithms on tasks of interest to cognitive scientists.

Current areas in cognitive science that are utilizing EC methods include the direct modeling of evolutionary processes, such as the role of learning in evolution, learning as a local search technique in a genetic algorithm, the evolution of modularity, the evolution of cooperation, and the evolution and learnability of language (e.g., see the biennial “Evolution of Language” conferences, or the Evolutionary Computation “Special Issue on EC and Cognitive Science”, Wiles & Hallinan, 2001).

Other domains use evolutionary algorithms for optimization, for example, testing theories of infant development; modeling populations of individuals engaged in cognitive tasks; testing outcomes following damage in neural network models; and exploring the range of behaviors in a dynamic model of an artificial language learning task.

In all of the cognitive science domains mentioned, evolutionary algorithms have been tested on specific problems, but little work has been done at a methodological level to characterize the nature of the tasks per se, and the way in which they interact with the evolutionary algorithms chosen. Many factors affect the performance of evolutionary algorithms, including the choice of fitness function, representation of the genome, population size, selection technique, and genetic operators.

## Learning and EC

For this study, the area of interest is the interaction of learning and evolution known as the Baldwin effect, first formalized as a computational problem by Hinton and Nowlan (1987). Hinton and Nowlan’s simulation of the Baldwin effect provided insight into how learning can guide evolution within a Darwinian, rather than a Lamarckian evolutionary framework.

In Hinton and Nowlan's model, each individual consisted of a bit string representing a simple neural network with twenty connections, which must be set correctly via either learning or evolution. A network that achieves the correct settings has a fitness dependant upon the time required to achieve the correct settings, while all incorrect networks have equal, minimal fitness. This task has a single fitness peak, which is surrounded by a perfectly flat fitness landscape, making it a classic needle-in-a-haystack problem (henceforth referred to as the *haystack* problem). The task is analogous to finding the components of a module in which no partial credit is given for partial solutions. Issues of modular design were popularized by Dawkins in the *Blind Watchmaker* (Dawkins, 1986), and are particularly relevant to understanding the evolution of complex cognitive systems.

The haystack task requires exhaustive search if genetic operators alone are used (crossover and mutation). However, if each agent modeled in the search population is allowed to perform some local searching, then the task can be solved by a much smaller population.

Hinton and Nowlan used a population size set to approximately the square root of the size of the search space, with each agent able to search on average a portion of the search space also equal to the square root of the size of the space. The choice of population size and local search space balanced the need for a population to have sufficient diversity to cover the space, and sufficient flexibility to find the "needle" (maximum fitness) in almost every trial.

Computationally, each individual is implemented as a string of twenty "genes", each of which may be either 1, 0, or ? (question mark). The ? represents a learnable gene. The individual learns by guessing 0 or 1 with a probability of 0.5. The target pattern is a string of twenty 1s. The number of guesses required to achieve this target is recorded and used to calculate the individual's fitness. The next generation is created by repeatedly selecting two parents, to produce pairs of new individuals. Parents are probabilistically selected proportional to the individual's fitness relative to the total population fitness.

Hinton and Nowlan (1987) demonstrated that under these conditions, the ability to learn, represented by ?s, was replaced by appropriate instincts, represented by 1s. The number of 1s rose from an initial 25% of alleles in the population to 50-80% after 50 generations, with the remainder of the alleles ?s. Non-target alleles, represented by 0s, disappeared from the population.

An interesting feature of Hinton and Nowlan's simulation is the persistence of learnable genes in the population once it has stabilized. Hinton and Nowlan suggested that there is very little selective pressure in favor of genetically specifying the last few connections,

since learning can occur in very few guesses. Several researchers have studied the phenomena of the residual question marks in the haystack problem and identified a variety of factors, including selection pressure and drift as significant factors in the results (Belew, 1989, Harvey, 1993).

In a previous study, we analyzed multiple simulations of the haystack problem to identify the characteristics of two classic selection algorithms (one fitness proportional and the other rank based) with respect to exploitation and exploration of the fitness landscape (Wiles et al., in press). These simulations demonstrated that fitness proportional selection finds good solutions and the average fitness of a population rises quickly, but at high fitness levels the population drifts gradually to homogeneity (all the alleles in one position on the chromosome are identical for all individuals in a population). Residual learning is frequently a result of an interaction between a pseudo-founder effect (dominance by one early successful solution) and drift to homogeneity at one or more of the genes. Selection by rank has the opposite effect, with populations drifting initially, until a critical mass find good solutions (or until an allele becomes homogeneous in 0s, resulting in an unsuccessful trial). Of the successful trials, at high fitness levels, populations converge to homogeneity based on fitness, rather than drift. By comparing fitness level and number of homogeneous genes to generation number, the relative effects of drift and selection pressure can be monitored during evolution.

The analyses in our previous study provide tools to understand how selection pressures are working during trials. The two techniques produce very different characteristic performance. Fitness proportional selection has initially fast fitness increases followed by slow convergence, whereas rank-based selection has initially slow and erratic fitness increases followed by fast convergence.

For the haystack problem, neither of these selection methods can be considered optimal in balancing the exploration of possible solutions with the exploitation of good solutions. Fitness proportional selection has too strong an exploitation of early successful solutions, leading to a pseudo-founder effect, and insufficient pressure to optimize when most of the population have good solutions. In contrast, rank-based selection has insufficient exploitation of its good early solutions, allowing drift to reduce the diversity of alleles available before fitness pressures shape the search space.

## Method

In this study, we report three sets of simulations. The first set replicates our previous work on the classic fitness proportional (roulette wheel) and stochastic

rank-based (tournament) methods and is reproduced here for comparison.

The second set of simulations was designed to investigate other selection algorithms on the haystack task, and also to test whether the analysis would be useful for their evaluation. For this set, we designed two additional algorithms to combine the search characteristics of fitness proportional and rank-based selection. The first operator was designed to exhibit fast fitness rises and fast convergence, and the second to exhibit slow fitness rises and slow convergence.

The third set of simulations directly addressed the problems inherent in fitness proportional and rank-based selection using modifications suggested in the literature. To modify fitness proportional selection, the expected number of offspring for any one individual was scaled in proportion to its deviation from the mean fitnesses of other individuals, which balances the selection pressure over a trial. To modify rank-based selection, the two fittest individuals (replacing the offspring of one pair of parents) were copied to the next generation, thus preserving good solutions once found. In the next section, the algorithms are described, and then the results summarized and presented together for ease of comparison.

### Simulation details for the haystack problem

Hinton and Nowlan (1987) modelled the Baldwin Effect using a simple genetic algorithm, with no mutation and a crossover value of 1.0; each pair of parents undergoes crossover once during each reproduction event. The next generation is created by repeatedly selecting two parents for each pair of new individuals. The probability of selecting an individual as a parent is proportional to its fitness divided by the total population fitness. The fitness,  $f$ , of an individual is calculated using the recorded number of guesses,  $g$ , taken to find the target:

$$f = 1 + \frac{(L-1)(G-g)}{G} \quad (1)$$

where  $G$  is the maximum number of guesses allowed and  $L$  is the length of the chromosome. In Hinton and Nowlan's model,  $G = 1000$ ,  $L = 20$ , and the population size,  $N = 1000$ .

We implemented Hinton and Nowlan's model, and as in our previous work, selection of each parent was implemented using a fitness proportional algorithm. After selecting two parents, a crossover point was chosen at random, and two new individuals were then created. Parameters were set similar to Hinton and Nowlan (1987), with initial proportions of 1, 0 and ? alleles set to 0.25, 0.25 and 0.50 respectively. A minor change from their parameters was setting both population size and maximum number of guesses to

1024 instead of 1000. All trials were run until the population converged (homogeneous in all genes).

For each selection method, 100 trials were performed. We report the proportion of trials that successfully eliminated all zeros from the population, the average number of residual question marks at the end of each trial, and the average number of generations to homogeneity (see Table 1). The average fitness pressures in the early and late stages of trials were calculated as the average number of generations until the fitness rose to 50% of the maximum (Stage 1) and from the midpoint to final convergence (Stage 2). By defining these values, the relative selection pressures early and late in a run can be compared. The average fitness of the population when the first gene in each trial became homogeneous was also calculated (see Table 1, column 4). This measure shows the potential exploration available to the algorithm.

### Set 1: Original algorithms

*Traditional roulette wheel (fitness proportional) selection:* The fitness of an individual is determined using equation (1) given above, and the selection procedure for two parents is as described for Hinton and Nowlan's simulations.

*Tournament (rank based) selection:* In this algorithm two candidates are selected at random from the parent population, and the individual with the higher fitness becomes a parent. The probability of being selected as a parent for the next generation therefore depends on the relative rank of an individual within the population, rather than its proportional fitness. Under tournament selection, the reduced fitness differential later in evolution does not change the ranking of individuals and selection pressure is maintained as long as there are different fitnesses within the population.

### Set 2: Modified algorithms

*Roulette with ranking:* In order to produce a selection strategy that should both begin and end rapidly, ranking was added to roulette wheel selection. This algorithm has also been called stochastic tournament (attributed to Wetzel by Goldberg, 1989). Continued pressure after the initial fast start means that selection will force the population to converge, rather than simply drifting to homogeneity.

To select each parent, two candidates are chosen using roulette wheel selection. The fittest of these two individuals becomes one parent, as in tournament selection. A second parent is selected in the same way. The fitness-proportional selection of candidates enables very successful individuals to have many offspring, in a similar manner to roulette wheel selection. The addition of a tournament between two candidates ensures that as fitness differentials decrease later in trials, the selection

pressures continue. The strategy is identical in all other ways to the others that have been used previously.

*Probabilistic tournament:* The second variation is a strategy that is designed to start slowly and end slowly. For this strategy, tournament selection was modified to include the proportional elements of roulette wheel strategy.

For each parent, two candidates are chosen randomly from the parent population. The one that will become a parent is chosen using proportional selection based on the fitness of these two individuals. That is, the one that is fitter will be more likely to be chosen than the less fit individual, but both still have a chance of being a parent. The selection of candidates with equal probability means that each individual, even the fittest one, can expect on average to contribute genes to a maximum of four offspring. The second parent is chosen in the same way, and reproduction continues as in the other selection strategies.

### Set 3: Optimized algorithms

*Sigma Scaled Roulette:* Amongst the known problems with roulette wheel selection is the variable selection pressure between early and late stages in a trial and the premature convergence of populations with inadequate exploration of the search space (Mitchell, 1996). A variety of modifications of roulette wheel selection have been proposed. One such mechanism is to balance the selection pressures evenly throughout a trial. Sigma scaled roulette is a renormalized version of roulette wheel. We use the description given by Mitchell (1986, who credits an early unpublished manuscript of Forrest from 1985). The expected number of offspring,  $E$ , is calculated from the mean and standard deviation of the fitnesses of the population:

$$E = 1 + \frac{f(i) - m}{2\sigma} \quad \text{if } \sigma \neq 0$$

$$= 1 \quad \text{if } \sigma = 0$$

where  $f(i)$  is the fitness of individual  $i$ ,  $m$  is the mean fitness of the population and  $\sigma$  is the standard deviation. This means that an individual with a fitness equal to the mean will gain a slice of the roulette wheel proportional to one unit. An individual with fitness one standard deviation above the mean will (on average) gain a slice proportional to 1.5 units, and one with fitness one standard deviation below will gain a slice proportional to 0.5 units. If the expected value for an individual is less than 0.1, then the slice is set to 0.1. The total size of the wheel is the sum of the slices of all individuals in the population. The expected number of offspring is proportional to the size of the slice, with corrections for the very small slices. For each pair of parents selected, two offspring are produced. Using the standard

deviation of the fitness maintains a constant selection pressure in the population throughout a trial.

*Elite tournament:* The slow initial period of all trials during tournament selection is a known problem. Even when a good solution is found, recombination of parents results in disruption of the solution and drift (rather than selection) can lead to homogeneity in one or more of the genes. Many researchers use elitism to preserve good solutions (first introduced in the 1970s by de Jong, according to Mitchell, 1996). In this strategy, one or more individuals with the highest fitnesses are copied to the next generation unchanged. In our implementation, elite tournament is identical to tournament selection, except that the individuals with the two highest fitnesses are copied to the next generation.

### Results and Discussion

The results from all three sets of simulations have been collated in Table 1, which shows the performance of each selection technique, the average number of residual question marks, and the number of generations to convergence (i.e., all genes become homogeneous) over 100 trials. Figures 1-3 show the selection pressures during early and late stages over 10 trials. Stage 1 is the average number of generations taken to reach a fitness value of 10 (i.e., 50% of the maximum fitness), the point at which fitness increases most rapidly in this task. Low values (few generations) indicate high initial selection pressure and high values indicate low selection pressure. Stage 2 is the average number of generations from this level of fitness to convergence of the population and indicates the selection pressure after the initial increase in fitness. The mean population fitness when the first allele becomes homogeneous generally indicates how good the solution will be. If an allele is homogeneous before the population is very fit, the solution tends to be poor.

The simulations in Set 1 (roulette wheel and tournament) provide a benchmark for the later studies. Performance in these simulations was consistent with our previous results (Wiles et al. in press). Using roulette wheel selection, all trials eliminated zeros from the population, and at convergence, individuals had an average of 1.4 residual question marks (stdev. 0.9). The change in selection pressure is revealed by the average number of generations spent in Stage 1 (10) versus Stage 2 (1437). See Table 1 for the means and standard deviations of all results.

Using tournament selection, 85/100 trials eliminated all zeros. In the successful trials, individuals retained an average of 1.2 residual question marks, with a much higher variance (stdev. 2.2). The average time spent in Stages 1 and 2 is reversed in this case, with 185 generations in Stage 1 and 23 generations in Stage 2.

The interaction between Stages 1 and 2 for roulette wheel and tournament selection is clear in Figure 1.

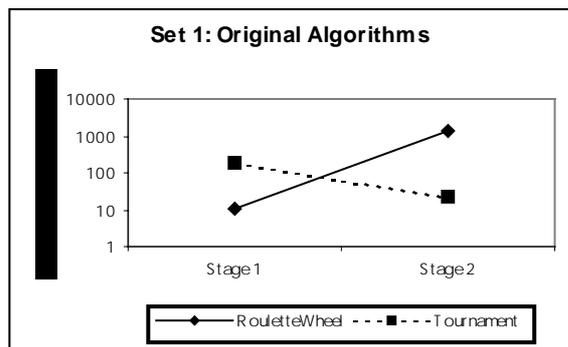


Figure 1. Time to convergence in Set 1, the original roulette wheel and tournament selection algorithms. Note that the y-axis is logarithmic. The algorithms show clear differences in behaviour, with roulette wheel faster in Stage 1, and tournament faster in Stage 2.

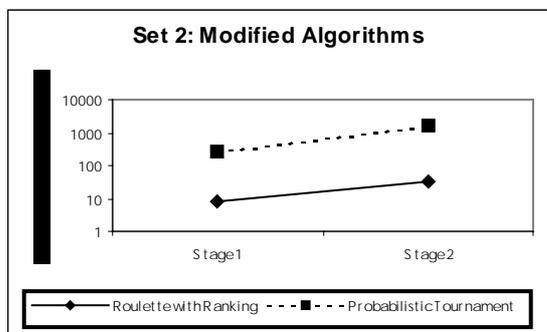


Figure 2. Time to convergence in Set 2, the modified algorithms. The Stages 1 and 2 components from Set 1 (see Fig 1), have been recombined as intended, to produce one algorithm in which both Stages 1 and 2 are fast (roulette with ranking), and the other algorithm in which both Stages 1 and 2 are slow (probabilistic tournament). Note that neither of these algorithms eliminate residual question marks.

In Set 2, the number of successful trials and residual question marks are similar to those from Set 1, but the time spent in Stages 1 and 2 differed markedly, as expected. Roulette wheel with ranking was fast in both stages (averages 7.6 and 33 generations respectively), and stochastic tournament was slow in both stages (average 249 and 1624 generations respectively, see Figure 2).

In Set 3, the original roulette and tournament selection procedures were modified to address their major known weaknesses, and both showed considerable improvement in optimization performance (as evidenced by the number of residual question marks). All trials eliminated zeros from the population, the time to convergence was short and very few residual question marks remained (an average of 0.02 in sigma scaled roulette and 0.06 in elite tournament).

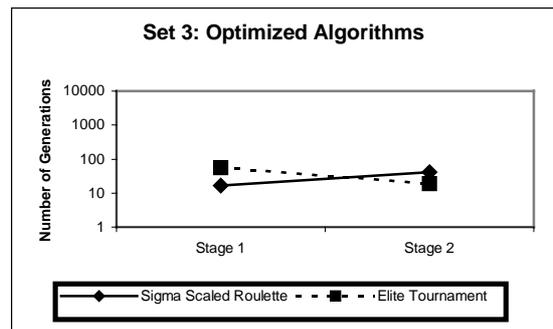


Figure 3. Time to convergence in Set 3, the optimized algorithms. Both sigma scaled roulette and elite tournament eliminate virtually all residual question marks and are much faster than the algorithms in Sets 1 and 2 (cf. Figs 1 and 2).

The average time spent in Stage 1 and Stage 2 was also much more balanced (16 and 42 generations respectively in sigma scaled roulette and 56 and 19 generations respectively in elite tournament, see Figure 3).

Premature convergence is a known problem for these evolutionary algorithms. Tracking progress towards homogeneity can therefore provide valuable information. The average fitness at which the first gene becomes homogeneous provides a quantifiable measure of diversity at a significant point in a simulation. This fitness value was recorded for each selection regime (see Table 1, column 5). Higher values (maximum is 20) indicate that higher levels of diversity are maintained in the population. For problems in which hitchhiking genes (sub-optimal genes that are carried by pseudo-founders in a population) are liable to cause problems such as in the haystack problem, this measure is a good indicator of potential problems with premature convergence. The tournament-based algorithms that have trials that do not eliminate all zeros show the lowest values with average population fitnesses at the first homogeneous alleles of 9.1 and 6.0 for Sets 1 and 2 respectively. Values for the corresponding roulette wheel-based algorithms are higher (16.7 and 13.6 respectively), but are not optimal. The best algorithms, those in Set 3 have the highest values (19.3 for sigma-scaled roulette and 19.9 for elite tournament) indicating that none of the trials suffered from premature convergence.

The combination of relatively balanced fitness pressures in Stages 1 and 2, short times to convergence, and high fitness before diversity is reduced indicate that both selection algorithms in Set 3 are well-adapted to the haystack task.

## Conclusions

One specific conclusion from these experiments is that residual learning is not an inherent aspect of the Baldwin effect. Rather, it is a consequence of the way

the fitness landscape is searched, and the application of selection pressures at different stages. The methodological studies presented in this paper are one way to explore such issues. Further work is needed to tie these results to biologically-plausible learning scenarios, but that is beyond the scope of this study.

At a more general level, the simulations show that the haystack task is one for which tailoring of the algorithm makes a qualitative difference to the behaviors observed. Specific issues addressed in this study concern the characteristics of the algorithms and the nature of the landscape.

The simulations of the original algorithms illustrate properties such as premature convergence in roulette wheel and the dangers of early homogeneity in one or more genes due to drift in tournament selection. With appropriate modifications, the optimized algorithms achieve a balance between exploration and exploitation, resulting in convergence to good solutions. Residual learning can be almost eliminated, and performance on the haystack problem can be near optimal.

These results illustrate the need for a characterization of task types in cognitive science, and a characterization of evolutionary algorithms and their performance on these tasks. Such a classification would facilitate the tailoring of algorithms to particular problems, and has the potential to significantly reduce artifacts due to implementation details.

### Acknowledgements

This project was supported by a CSEE summer grant to RS, and an APA to SB.

### References

- Baldwin, J. M. (1896). A new factor in evolution. *American Naturalist* 30: 441 – 451. Reproduced in (eds.) Belew, R.K. & Mitchell, M.,. *Adaptive Individuals in Evolving Populations, Proceedings Volume XXVI, Santa Fe Institute Studies in the Sciences of Complexity*. Addison-Wesley, Reading, MA.
- Belew, R. K. (1990). Evolution, learning and culture: Computational metaphors for adaptive search. *Complex Systems* 4 (1), 11-49.
- Dawkins, R. (1986). *The Blind Watchmaker: Why the evidence of evolution reveals a universe without design*,

- NY: Norton.
- Goldberg, D.E. (1989), *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, MA.
- Harvey, I. (1993). The puzzle of the persistent question marks: A case study of genetic drift. *Computer Science Research Paper Serial No. CSRP 278*, The University of Sussex. (Also published in S. Forrest, (Ed.) *Proceedings of the Fifth Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, 1993.)
- Hinton, G. E. & Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems* 1: 495 –502.
- Maynard Smith, J., (1998). *Evolutionary Genetics*, second edition. Oxford University Press: NY.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press: Cambridge, MA.
- Wiles, J. and Hallinan, J.S. (eds) *IEEE Trans. on Evolutionary Computation Special Issue on EC and Cognitive Science*. 5 (2), 2001.
- Wiles, J., Schulz, R., Hallinan, J., Bolland, S., and Tonkes, B., *Probing the Persistent Question Marks*, to appear in the Proceedings of GECCO, 2001.
- Wolpert, D.H. and Macready, W.G. (1997), No Free Lunch Theorems for Optimization. *IEEE Trans Evolutionary Computation*, April 1997, pp. 67-82.

Table 1. Summary of Numerical Results

Selection Strategy	All Trials	Successful Populations		
	Proportion of trials that eliminated 0s	Residual homogeneous ?s [Mean (SD) of 20 trials]	Generations to homogeneity [Mean (SD)]	Av fitness at 1 <sup>st</sup> homogeneous allele [Mean (SD) of 10 trials]
Roulette Wheel	100	1.44 (0.91)	1448.16 (734.66)	16.70 (1.68)
Tournament	85	1.24 (2.22)	208.67 (142.60)	9.13 (8.81)
Roulette with Ranking	100	1.27 (1.25)	41.45 (9.58)	13.62 (3.08)
Probabilistic Tournamt	80	2.08 (1.79)	1873.66 (1035.19)	6.01 (8.19)
Sigma Scaled Roulette	100	0.02 (0.14)	57.93 (5.41)	19.34 (0.53)
Elite Tournament	100	0.06 (0.34)	75.01 (20.12)	19.90 (0.08)