On Neutral Networks and Evolvability

Marc Ebner[1], **Patrick Langguth**[1], **Jürgen Albert**[1], **Mark Shackleton**[2], and **Rob Shipman**[2]

[1] Universität Würzburg
Lehrstuhl für Informatik II, Am Hubland
97074 Würzburg, Germany
{ebner,plata,albert}@informatik.uni-wuerzburg.de

[2] BT Labs at Adastral Park
Admin 2-5, Martlesham Heath
Ipswich, IP5 3RE, UK
{mark.shackleton,rob.shipman}@bt.com

Abstract- Evolutionary algorithms apply the process of variation, reproduction and selection to look for an individual capable of solving the task at hand. In order to improve the evolvability of a population we propose to copy important characteristics of nature's search space. Desired characteristics for a genotype-phenotype mapping are described and several highly redundant genotype-phenotype mappings are analyzed in the context of a population based search. We show that evolvability is influenced by the existence of neutral networks in genotype space. The extent of the neutral networks affects the interconnectivity of the search space and thereby affects evolvability. Species evolving on a non-redundant mapping reach a state of stasis after a few number of generations. In effect, evolution comes to a halt. However, species evolving on a genotype-phenotype mapping with extensive neutral networks are continuously able to find adaptive mutations and are able to locate higher optima. The existence of highly intertwined neutral networks increases the evolvability of a population.

1 Introduction

Evolutionary algorithms apply the process of variation, reproduction and selection to look for a solution to a problem [7, 6, 15]. Although this is a general method, which can be used for a variety of different tasks, the method is usually used without regard to the characteristics of nature's search space. Nature's search space shows a high degree of redundancy in that many different genotypes map to the same phenotype [17] (Figure 1). Different genotypes that map to the same phenotype are randomly distributed over the search space. Some phenotypes occur more often than others. All common phenotypes are located in the vicinity of a randomly selected genotype and long paths of neutral mutations connect genotypes which code for the same phenotype. The presence of neutral mutations induces so called neutral networks [8, 9, 22]. A neutral network is a set of genotypes which map to the same phenotype and which are connected via single point mutations. According to the neutral theory of evolution, most mutations are neutral and only a small fraction of all mutations are actually beneficial [14].

Ebner [3] and Shipman [19] have previously argued that similar characteristics might also be useful for an artificial search algorithm. A number of studies were carried out to analyze the benefits of a highly redundant genotype-phenotype mapping which shows similar characteristics to nature's search space. Shipman et al. [20] analyzed the search space created by several redundant genotype-phenotype mappings with different amounts of redundancy. Shackleton et al. [18] introduced the concept of fitness and performed fitness adaptive walks for a single point moving through genotype space.

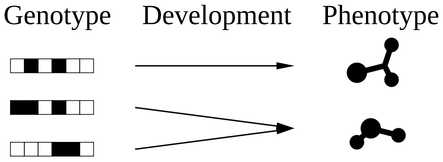


Figure 1: Nature's genotype-phenotype mapping is highly redundant in that many genotypes map to the same phenotype.

Other research exploring the effects of redundancy include the work of Levenick [16] and Julstrom [10]. Levenick has looked at the advantages of having introns for a dynamic environment. Julstrom established that redundancy is beneficial in looking for a solution to the problem of partitioning $3n$ points into 3-cycles of minimum total length. Banzhaf [1] studied a simple redundant map in the context of linear computer programs and Keller and Banzhaf [13] evolved a genotype-phenotype map for computer programs. The effects of a simple redundant mapping on the search space were analyzed theoretically by Kargupta [11].

In this paper, we investigate the characteristics of the search space for a population based search. In particular, we separate the search space into phenotypes which belong to different species. In this context, the following questions are of particular interest. To what extent is the evolvability (defined as the ability of random variations to sometimes produce improvement [23]) of a population dependent on the genotype-phenotype mapping used and to what extent are neutral networks a cause of punctuated equilibria (stasis punctuated by episodic events of speciation [4]). See Dawkins [2] for an introduction to the theory of punctuated equilibria. Does evolution eventually fall into a state of stasis if a highly redundant mapping is used? It is suspected that, as the individuals move *inside* the neutral networks, they are no longer able to discover beneficial mutations. However, this does not seem to be the case for a highly intertwined neutral network. Our results suggest that periods of stasis are mostly

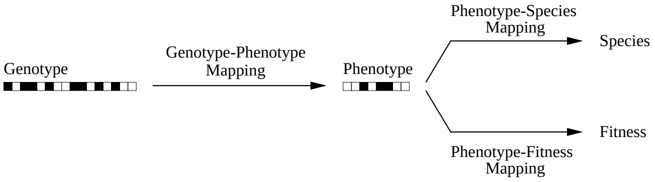

Genotype
Genotype-Phenotype Mapping
Phenotype
Phenotype-Species Mapping
Species
Phenotype-Fitness Mapping
Fitness

Figure 2: The genotype-phenotype mapping is used to determine the phenotype for each genotype. For highly redundant mappings the genotype space is much larger than the phenotype space. In addition to the genotype-phenotype mapping we have used a phenotype-species mapping, which determines the species of the phenotype and a phenotype-fitness mapping, which determines the fitness of the phenotype.

due to the scarcity of better fitness values as the species adapts to its environment and are not caused by the extent of the neutral networks. It appears thus, that the mappings which are analyzed in this paper possess neutral networks which are intertwined with a high degree of connectivity. If our search space has similar characteristics as nature's search space as described above, the individuals should have no problem discovering other species and thereby filling all available niches of the search space.

2 Mappings

In order to answer the above questions we looked at several different genotype-phenotype mappings. Each mapping specifies a phenotype for each possible genotype. The mappings differ in the amount of redundancy used. Two additional mappings separate the phenotypes into species and assign fitness values to the phenotypes. The phenotype-species mapping defines which phenotypes belong to a particular species. The phenotype-fitness mapping defines the fitness values for each phenotype. How the different mappings are used, is illustrated in Figure 2.

2.1 Genotype-phenotype mappings

We experimented with several different genotype-phenotype mappings: a standard binary mapping, a mapping based on a cellular automaton, and a mapping based on a random boolean network. The cellular automaton mapping and the random boolean network mapping are used here because we have shown previously that these highly redundant mappings also possess extensive neutral networks [20, 18].

Binary mapping

For the binary mapping, genotype space and phenotype space are equivalent. In this case the same number of bits are used for the genotype as well as for the phenotype. There exists a one to one correspondence between genotype and phenotype. The phenotype number is obtained by interpreting the genotype as a binary number.

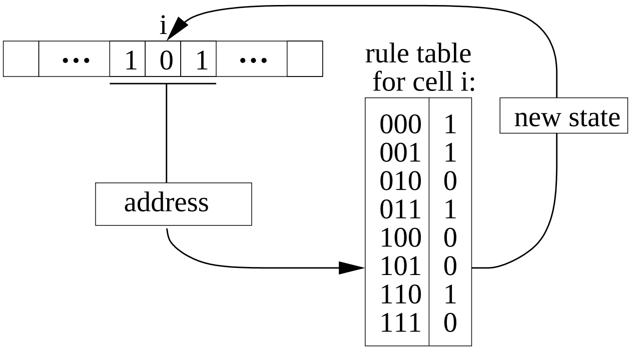


Figure 3: A linear non-uniform cellular automaton. The next state of the cell i is determined by the state of cell i as well as the state of its neighboring cells $i-1$ and $i+1$ (with circular boundary conditions). The state of these three cells specify an address in the rule table of cell i which determines the state of cell i at the next time step.

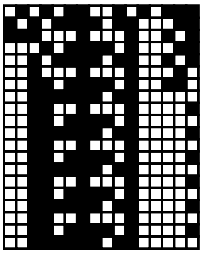


Figure 4: A sample run of a cellular automaton with 16 cells for 20 steps. The initial state of the automaton was a fixed random value.

Cellular automaton mapping

A linear cellular automaton consists of an array of cells. Each cell has two possible states: on or off. The state of a cell at the next point in time is determined by the state of cell at the current time step and by the state of its neighboring cells. See Wolfram [24, 25] for an introduction to cellular automata. For a non-uniform cellular automaton, each cell has its unique rule table which defines the new state of the cell for all possible combinations of state values of the cell and its neighbors [21]. We have used a neighborhood of 2. Therefore the rule table of each cell has 2^3 entries. How the cellular automaton works is described in Figure 3. A sample run of a cellular automaton with 16 cells is shown in Figure 4.

For the cellular automaton mapping, the genotype specifies the rule tables of the automaton. This mapping has been analyzed in detail by Shipman et al. [20] and Shackleton et al. [18]. The phenotype is determined by running the cellular automaton for a fixed number of steps (20 in our experiments). Initially, the automaton is placed into a fixed randomly chosen state. The initial state is the same for all genotypes. At the end of the run, the resulting state of the network is used as the number of the phenotype. Let n be the number of cells

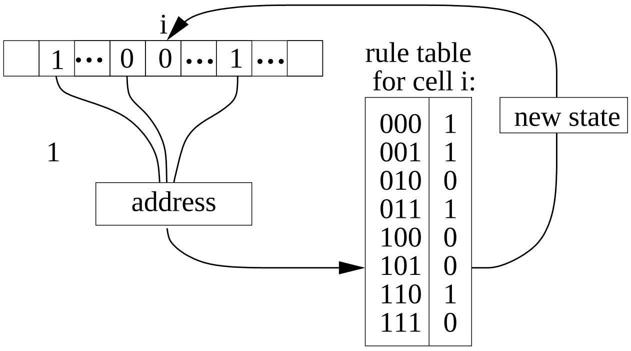


Figure 5: A random boolean network. The next state of cell i is determined by the state of the cells it is connected to. The state of these three cells specify an address in the rule table of cell i which determines the state of cell i at the next time step. For our experiments all cells are connected to three other cells.

of the cellular automaton. Then we need $8n$ bits to define the cellular automaton because 8 bits are needed for the rule table of each cell.

Random boolean network mapping

A random boolean networks consists of a fixed number of cells [12]. Each cell has two possible states: on or off. The state of a cell at the next point in time is determined by the state of the cells it is connected to. Each cell has a rule table which lists the new state of the cell for all possible combinations of state values of the cells it is connected to. For our experiments we have used a connectivity of 3. Therefore the rule table of each cell has 2^3 entries. How a random boolean network works is shown in Figure 5

For the mapping based on a random boolean network, the genotype specifies the initial state, its wiring and all of the rule tables. The mapping was introduced by Shipman et al. [20]. The phenotype is determined by running the random boolean network for a fixed number of steps (20 in our experiments). After that, the resulting state of the network is used as the number of the phenotype. Let n be the number of cells of the random boolean network. Each cell has n_c cells to which it is connected. Then we need $n(1 + n_c \log_2(n) + 2^{n_c})$ bits to describe the random boolean network. For each cell we need one bit to specify the initial state, $n_c \log_2(n)$ bits to specify to which cells it is connected to, and 2^{n_c} bits for the rule table.

2.2 Phenotype-species mapping

The phenotype-species mapping determines the species to which each phenotype belongs. Let n_s be the number of species and n_p be the number of phenotypes, then exactly $\frac{n_p}{n_s}$ phenotypes are associated with each species. The phenotype-species mapping is created by randomly distributing the species over the phenotypes.

2.3 Phenotype-fitness mapping

The phenotype-fitness mapping determines the fitness of each phenotype. Let n_s be the number of species and n_p be the number of phenotypes, then we need to define $\frac{n_p}{n_s}$ fitness values for each species. For each species we have used the fitness values 1 through $\frac{n_p}{n_s}$. The phenotype-fitness mapping is created by randomly distributing the fitness values for each species over the phenotypes which belong to the species. Thus, the redundancy is only in the genotype-phenotype mapping and not in the phenotype-fitness mapping because fitness values are separate for each species.

3 Experiments

For our experiments we have used a phenotype space of $n_p = 2^{16}$ phenotypes. Each phenotype belongs to one of $n_s = 64$ species. Therefore, we have used $\frac{n_p}{n_s} = 1024$ fitness values for each species. For the binary mapping the genotype and phenotype spaces are equivalent. The genotype consists of 16 bits and is interpreted directly as the number of the phenotype (no redundancy). For the cellular automaton mapping we need 128 bits for the genotype (redundancy of $2^{112} : 1$). For the random boolean network mapping we need 336 bits for the genotype (redundancy of $2^{320} : 1$). In order to investigate the influence of the genotype-phenotype mapping on evolvability we have made the following assumptions. The association between phenotype and species as well as the association between phenotype and fitness is fixed. That is, for each phenotype we are able to tell to which species the phenotype belongs and how fit the phenotype is.

Three experiments were performed. For the first two experiments we start with a single point moving though genotype space. The point represents a single species which later spawns new species which then move through genotype space in parallel. The first two experiments differ in how fitness is handled. For the first experiment we allow a speciation event to take place if the new species does not exist already. For the second experiment we make speciation events dependent on the environment. Finally, the third experiment uses multiple species right from the start.

3.1 Experiment 1 – Speciation by niche filling

First, we choose a random species and look for a genotype that belongs to this species which has the lowest possible fitness. This is our starting point in genotype space. Lowest possible fitness is chosen in order to better compare results across mappings. At each time step we randomly mutate the genotype. Next we determine the species to which the mutated individual belongs. If the mutated individual belongs to the same species we look at the fitness of the individual. If the fitness value is higher than or equal to the fitness of the original individual, then the mutated individual replaces its ancestor. If the fitness of the mutated individual is lower than the fitness of the original individual nothing happens. In

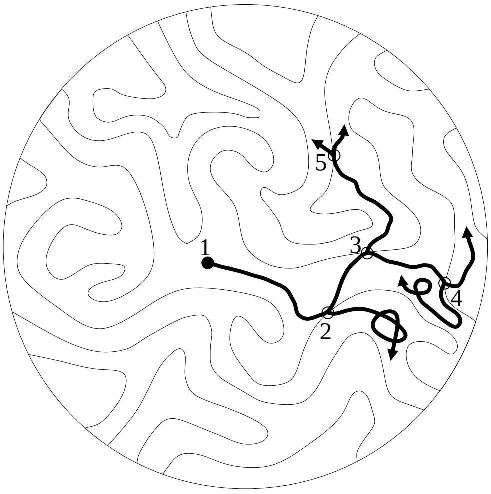


Figure 6: Genotype space is separated into different species by the genotype-phenotype mapping and the phenotype species mapping. Initially only a single point moves through genotype space. As new species are discovered, this point splits into a new point moving inside the newly discovered species and also continues inside the region belonging to the old species. The filled circle marks the starting point. Open circles mark points where new species are discovered.

case the mutated individual belongs to a new, currently not existing species then a new point is created which now also moves through genotype space. For this experiment we only use fitness to determine if a new genotype replaces an existing genotype after the species has been discovered.

Initially, only a single point moves through genotype space. As new species are discovered, the trajectory splits into a trajectory for the new species and a trajectory for the existing species. This process is shown in Figure 6. After a relatively short time all species have been discovered. This is due to the parallel search done by all of the points moving through genotype space. We are able to divide time into two periods. A period of speciation followed by a period of adaptation. After all species have been discovered they increase their fitness by adapting to their environment. The separation of phenotypes into species can be regarded as evolutionary niches that may be filled once they are discovered. After a species has been discovered, it may not be discovered again.

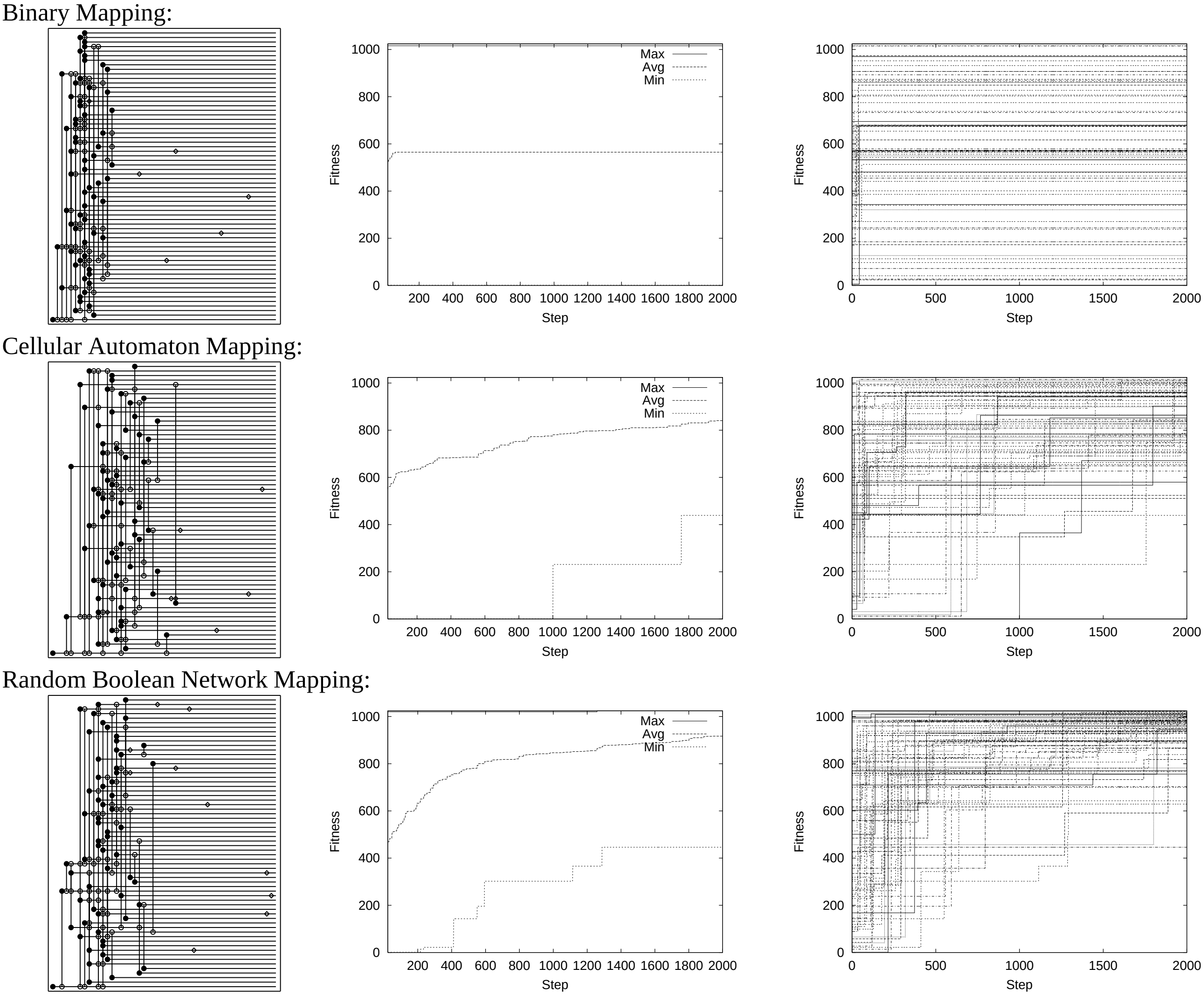
The results of this experiment are shown in Figure 7. Results for the binary mapping, cellular automaton mapping, random boolean network mapping are shown in the first, second, and third rows respectively. The first graph for each mapping shows how the species with which we started, speciates into the other species. Time is shown along the horizontal axis. Each species has its unique slot on the vertical axis. A point of speciation is marked by drawing a vertical line from the original species to the new species. The endpoints of the line are drawn with open circles on the original species and closed circles are used for the newly created species. Unfilled diamonds mark position at which a new phenotype with higher fitness is discovered. After all species have been discovered we look at the fitness values. The second graph shows minimum, average and maximum fitness values over all species. The third graph shows the individual fitness values of all species over time. It is interesting to notice that the time taken to fully populate the ecological niches is insignificant compared to the time taken for the species to better adapt to those niches.

As can be seen from the second graph, the larger the amount of redundancy, the better the species are able to adapt to their environment. In case of a binary mapping, the species don't evolve at all after a few adaptive moves at the beginning of the experiment. Instead, they become stuck in a local optimum. Average fitness is halfway between the lowest possible and the highest possible fitness value. For the cellular automaton mapping we get a slightly better ability to evolve. Higher average fitness values are reached than for the binary mapping. The mapping based on the random boolean network performs even better than the cellular automaton mapping. Higher average fitness values are reached. As can be seen from the fitness plots, there exists relatively long periods of stasis in between adaptive steps. It may be that these long periods of stasis are caused by movements along the neutral networks.

Most of the time, the species moves along the neutral networks. This results in the period of stasis. After a while a new adaptive mutation is discovered. As the species adapts to its environment fewer and fewer adaptive mutations are possible. That is, initially, at the lowest possible fitness, a species may move to one of 1023 possible adaptations. One expects that on average the next fitness value will be around 512. However, if a species has reached a fitness value of 512, only half as many adaptive moves are possible. Therefore it will take twice as long before the next adaptive mutation is discovered. Elena et al. [5] showed that punctuated evolution observed in bacteria is caused by selection of rare beneficial mutations. Therefore, it may be that the periods of stasis are actually caused by the decreasing number of available adaptive mutations. The next experiment was carried out to further investigate this hypothesis.

3.2 Experiment 2 – Adaptive speciation

For experiment 2(a), we first make the speciation event dependent on the fitness of the current species. A speciation event is only allowed if the new species has a higher or equivalent fitness as the current species. Other than that experiment 2(a) is equivalent to experiment 1. The results of experiment 2(a) are shown in Figure 8. We now see that speciation happens at a slower rate. As the fitness of the species improves, it takes longer and longer before a new species with higher fitness is discovered. A period of stasis results if the species move along the neutral network. After some time a new species with a higher fitness or equivalent value is discovered.


Binary Mapping:

Cellular Automaton Mapping:

Random Boolean Network Mapping:

Figure 7: Results of experiment 1 for the binary mapping, cellular automaton mapping, and the random boolean network mapping. The first graph shows how the initial point in genotype space speciates. Time is shown along the horizontal axis. Each species has its unique slot on the vertical axis. The beginning of each species is marked by a filled circle. Speciation events are marked with filled circles and a line connecting the original and the new species. Diamonds mark adaptive mutations. Speciation is shown for 50 time steps. The second graph shows the minimum, average and maximum fitness values of the different species for each generation. The third graph shows the fitness values of the individual species for each generation. It can be seen clearly how fitness improves in sudden bursts with periods of stasis in between adaptive mutations.

Next, we make speciation events dependent on the maximum fitness of all species (Experiment 2(b)). In this case, a speciation event is only allowed if the new species has a higher or equivalent fitness as the maximum fitness found so far. The results of this experiment are shown in Figure 9. Now we get an even slower rate of speciation. The behavior is similar to the adaptive evolution of fitness described above. At the same time, the advantage of the right kind of redundancy also becomes apparent. As can be seen from the graphs, instead of prolonging the search, neutral networks increase the reachability of new phenotypes and thereby aid speciation. Whereas the binary mapping becomes stuck right away, the cellular automaton mapping and the random boolean network mapping enable the population to discover additional species.

3.3 Experiment 3 – Population based search

The above two experiments are of interest if the mappings are to be used for an evolutionary algorithm. In this context it is of particular importance to know if the use of a redundant mapping is a possible cause of stasis. Since one is usually interested in finding a solution quickly the use of a redundant mapping would be a considerable drawback in this case. However, the above results show that the right type of redundancy actually improves evolvability and that periods of

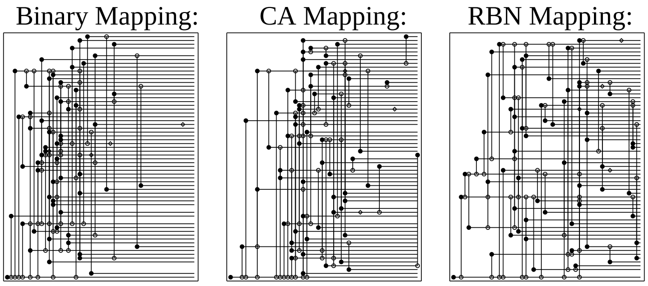


Figure 8: Results of experiment 2(a). Only adaptive speciation events are allowed. It can be seen how the requirement that the new species must have a fitness which is higher than or equal to the fitness of the existing species delays speciation events. Speciation is shown for 50 time steps.

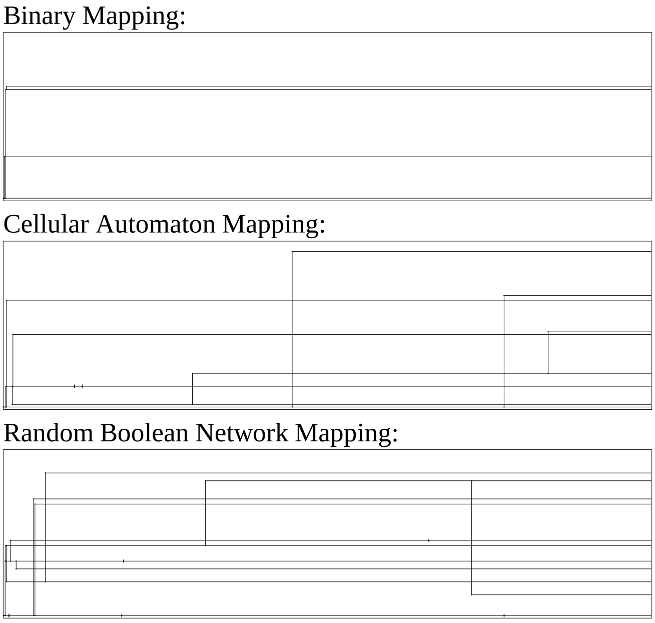


Figure 9: Results of experiment 2(b) if speciation events are dependent on the maximum fitness found so far. The graphs show results for the binary mapping, cellular automaton mapping, and the random boolean network mapping. This graph clearly shows how neutral networks increase the reachability of new species as opposed to prolonging the search. Speciation events are shown for 1000 time steps.

stasis are mostly due to the scarcity of better fitness values as the species adapts to its environment.

An evolutionary algorithm usually uses a population of individuals. Thus, for our third experiment we start with all species moving through genotype space right from the start. For each species we select a single genotype which is a member of the species and has the lowest possible fitness. Since all species exist right from the start, no new species are available to be discovered. The species may only improve their fitness as they move through genotype space. Thus only moves to a neutral neighbor or to a neighbor with a higher fitness value are allowed.

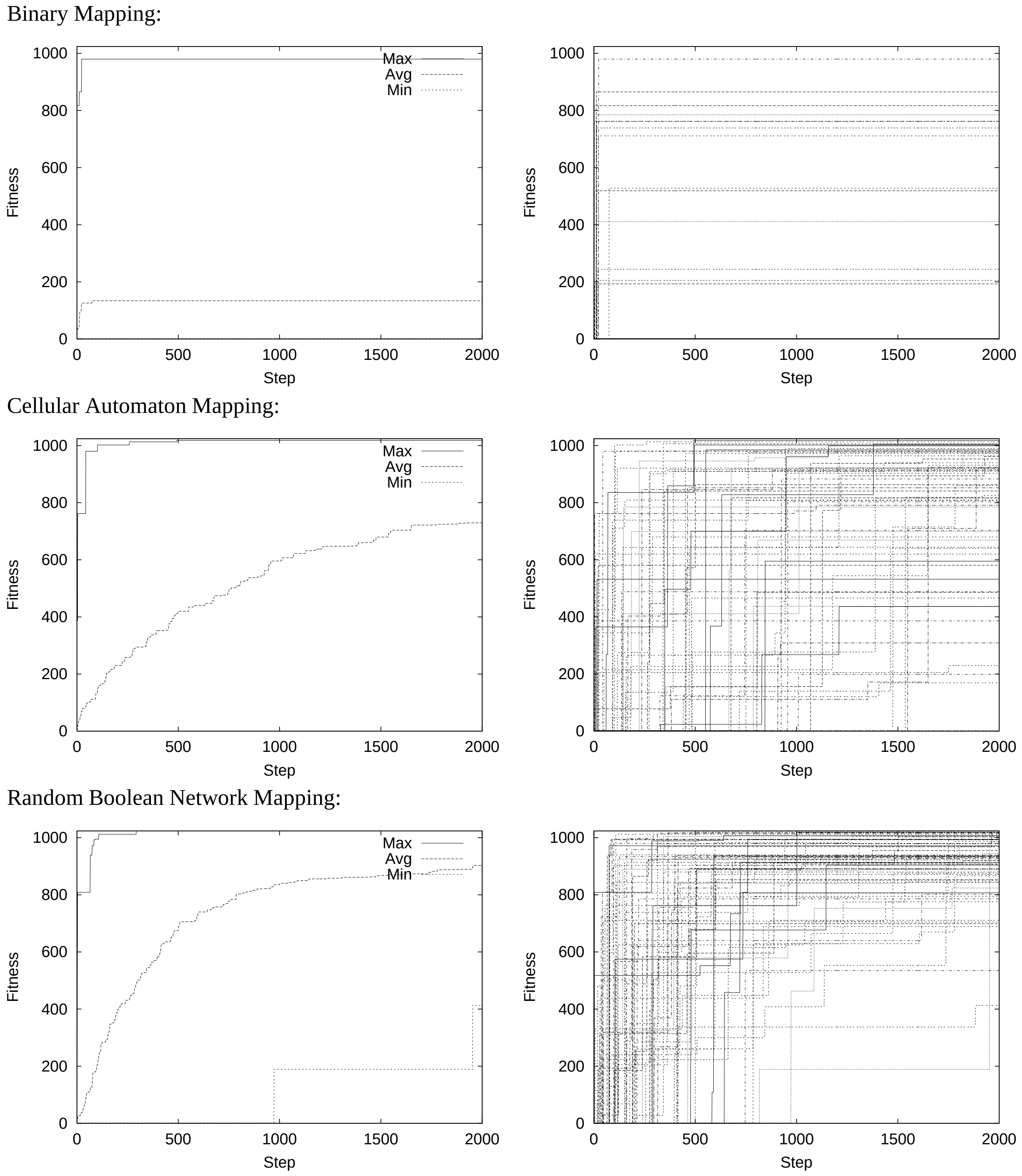
For the first two experiments we always started with a single species (or search point) in genotype space. From this point new species (or new solutions) were discovered. It might seem that one should be at a serious disadvantage if one only starts with a single point in the search space as compared to multiple points all randomly distributed over the search space. It could be that one becomes stuck in a local optimum if the connectivity of neutral networks is low. That is, it might be that it is not possible to reach some points of the search space if we start in a particular region of the search space. However, the results of experiment 3 show that this is not the case. The results are shown in Figure 10. The average fitness value reached for the random boolean network mapping at the end of the run is comparable to the average fitness value reached in the first experiment which started from a single point in the search space.

4 Conclusion

We proposed to use a genotype-phenotype mapping which shows some characteristics as nature's search space for an artificial evolutionary algorithm. Three different genotype-phenotype mappings were analyzed in the context of a population based search. We have shown that evolvability, defined as the ability of random variations to sometimes produce improvement, is influenced by the existence of neutral networks created by the genotype-phenotype mapping. In order to investigate the mappings we have developed a model of speciation and adaptation. We start with a single point in the search spaces which moves through genotype space. Over time, new search points are created by spawning new trajectories which move through genotype space in parallel.

Depending on the genotype-phenotype mapping used, we are either able to continue to find better points of the search space or become stuck in a local optimum. The extent of the neutral networks clearly affects the interconnectivity of the search space as was observed in the case of adaptive evolution described above. Highest average fitness values were reached for the random boolean network mapping, followed by the cellular automaton mapping. In case of a binary mapping the individuals did not evolve at all. The search points quickly became stuck in a local optimum.

The fitness of the species increased in jumps with intermediate periods of stasis. The same behavior was observed for speciation events if the speciation events were made dependent on the fitness of the current species or the maximum fitness found so far. It was suspected that the periods of stasis were caused by species moving *inside* neutral networks. However, the current results suggest that the periods of stasis are mostly due to the scarcity of better fitness values as the species adapts to its environment. It appears thus, that the neutral networks are intertwined with a high degree of connectivity such that the species may move *along* the neutral networks to discover new adaptive mutations. This is an im-



Binary Mapping:

Cellular Automaton Mapping:

Random Boolean Network Mapping:

Figure 10: Results of experiment 3 for the binary mapping, cellular automaton mapping, and the random boolean network mapping. Experiment 3 starts a parallel search with all species initialized to the lowest possible fitness. The first graph shows the minimum, average and maximum fitness values of the different species for each generation. The second graph shows the fitness values of the individual species for each generation.

portant property which distinguishes the mappings described in this paper from other types of redundancy. Instead of prolonging the search, the neutral networks present in the mappings which were investigated in this paper, increase evolvability.

Acknowledgment

We thank Susan Ptak from Stanford University for helpful discussions and comments.

References

[1] W. Banzhaf. Genotype-phenotype-mapping and neutral variation – a case study in genetic programming. In Y. Davidor, H.-P. Schwefel, and R. Männer, eds., *Parallel Problem Solving from Nature – PPSN III. Proc. of the Int. Conf. on Evolutionary Computation*, pp. 322–332, Berlin, 1994. Springer-Verlag.

[2] R. Dawkins. *The Blind Watchmaker*. W. W. Norton & Company, New York, 1996.

[3] M. Ebner. On the search space of genetic programming and its relation to nature's search space. In *Proc. of the 1999 Congress on Evolutionary Computation, Vol. 2, Washington, D.C., July 6-9*, pp. 1357–1361. IEEE Press, 1999.

[4] N. Eldredge and S. J. Gould. Punctuated equilibria: An alternative to phyletic gradualism. In T. J. M. Schopf, ed., *Models in Paleobiology*, pp. 82–115, San Francisco, 1972. Freeman, Cooper & Company.

[5] S. F. Elena, V. S. Cooper, and R. E. Lenski. Punctuated evolution caused by selection of rare beneficial mutations. *Science*, 272:1802–1804, 1996.

[6] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Reading, MA, 1989.

[7] J. H. Holland. *Adaptation in natural and artifical systems: an introductory analysis with applications to biology, control, and artificial intelligence*. The MIT Press, Cambridge, MA, 1992.

[8] M. A. Huynen. Exploring phenotype space through neutral evolution. *Journal of Molecular Evolution*, 43:165–169, 1996.

[9] M. A. Huynen, P. F. Stadler, and W. Fontana. Smoothness within ruggedness: The role of neutrality in adaptation. *Proc. Natl. Acad. Sci. USA*, 93:397–401, 1996.

[10] B. A. Julstrom. Redundant genetic encodings may not be harmful. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, eds., *GECCO-99: Proc. of the Genetic and Evolutionary Computation Conference, Vol. 1, Orlando, FL*, p. 791, San Francisco, CA, 1999. Morgan Kaufman Publishers.

[11] H. Kargupta. The genetic code and the genome representation. In *In Workshop "Gene Expression: The Missing Link in Evolutionary Computation" at the Genetic and Evolutionary Computation Conference (GECCO-2000), Las Vegas, NV*, 2000.

[12] S. A. Kauffman. *The Origins of Order. Self-Organization and Selection in Evolution*. Oxford University Press, Oxford, 1993.

[13] R. E. Keller and W. Banzhaf. The evolution of genetic code in genetic programming. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, eds., *GECCO-99: Proc. of the Genetic and Evolutionary Computation Conference, Vol. 1, Orlando, FL*, pp. 1077–1082, San Francisco, CA, 1999. Morgan Kaufman Publishers.

[14] M. Kimura. *Population Genetics, Molecular Evolution, and the Neutral Theory: Selected Papers*. The University of Chicago Press, Chicago, 1994.

[15] J. R. Koza. *Genetic Programming, On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge, MA, 1992.

[16] J. R. Levenick. Swappers: Introns promote flexibility, diversity and invention. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, eds., *GECCO-99: Proc. of the Genetic and Evolutionary Computation Conference, Vol. 1, Orlando, FL*, pp. 361–368, San Francisco, CA, 1999. Morgan Kaufman Publishers.

[17] P. Schuster. Extended molecular evolutionary biology: Artificial life bridging the gap between chemistry and biology. In C. G. Langton, ed., *Artificial Life: An Overview*, pp. 39–60, Cambridge, MA, 1995. The MIT Press.

[18] M. Shackleton, R. Shipman, and M. Ebner. An investigation of redundant genotype-phenotype mappings and their role in evolutionary search. In *Proc. of the 2000 Congress on Evolutionary Computation, San Diego Marriott Hotel, La Jolla, CA, Vol. 1*, pp. 493–500. IEEE Press, 2000.

[19] R. Shipman. Genetic redundancy: Desirable or problematic for evolutionary adaptation? In A. Dobnikar, N. C. Steele, D. W. Pearson, and R. F. Albrecht, eds., *4th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA '99)*, pp. 337–344, New York, 1999. Springer-Verlag.

[20] R. Shipman, M. Shackleton, M. Ebner, and R. Watson. Neutral search spaces for artificial evolution: A lesson from life. In M. A. Bedau, S. Rasmussen, J. S. McCaskill, and N. H. Packard, eds., *Artificial Life: Proc. of the 7th Int. Conf. on Artificial Life*. MIT Press, 2000.

[21] M. Sipper. *Evolution of Parallel Cellular Machines: The Cellular Programming Approach*. Springer-Verlag, Berlin, 1997.

[22] E. van Nimwegen, J. P. Crutchfield, and M. Huynen. Neutral evolution of mutational robustness. *Proc. Natl. Acad. Sci. USA*, 96:9716–9720, 1999.

[23] G. P. Wagner and L. Altenberg. Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976, 1996.

[24] S. Wolfram. Statistical mechanics of cellular automata. *Rev. of Modern Physics*, 55(3):601–644, 1983.

[25] S. Wolfram. Cellular automata as models of complexity. *Nature*, 311(4):419–424, 1984.