A Multimodal GA and a Visualization Technique Applied to the Hopfield Model of Associative Memory

Akira Imada

Graduate School of Information Science
Nara Institute of Science and Technology
1-11-13-604, Abeno, Hannan, Osaka 545-0021 Japan

Abstract. We apply evolutionary computations to the fully-connected neural network model of associative memory. In the model, a number of patterns can be stored in the network as attractors if synaptic weights are determined appropriately. In this paper, a multimodal genetic algorithm, called the Deterministic Crowding Genetic Algorithm, explores pattern space searching for the attractors, and the search is visualized by the Principal Component Analysis.

1 Introduction

We study the fully-connected neural network model of associative memory using evolutionary computations. In the study, two approaches have been considered: one is to explore *pattern space* searching for attractors under a specific weight configuration, and the other is to explore *weight space* searching for an appropriate weight configuration that stores a given set of patterns. The exploration of weight space enables us to have an information about the number and distribution of weight configurations that give a network a function of associative memory, which is still an open issue despite of a decade of discussions. To address this issue, we have applied a variety of evolutionary algorithms to the weight space, and have found many such solutions in the space [**?**, **?**, **?**, **?**]. The question is whether these solutions are representative of the whole solutions. Usually, evolutionary algorithms converge only on one of these solutions at a time. So, with the goal being obtaining representative samples from the whole solution space, we have tested some multimodal evolutionary techniques which search for multiple solutions simultaneously. However, we have found it very difficult to explore the weight space by the multimodal technique in general, probably due to the high dimensionality of the search space and real-valued variables of the domain. A network with N neurons constructs the N^2-dimensional weight space and the weights are real-valued. Therefore, as one of preliminary steps for the exploration of weight space, we explore pattern space in this paper. The dimensionality is N instead of N^2 and the variables are binary instead of continuous. The Deterministic Crowding Genetic Algorithm proposed by Mahford [**?**] is employed here, and the Principal Component Analysis visualizes the search by mapping points in the multi-dimensional search space into the one-dimensional space.

The remainder of this paper is organized as follows. Since we argue solutions in terms of *fitness landscape*, some remarks on the concept of fitness landscape is briefly given in Section 2. Section 3 gives some description about the fully-connected neural network model of associative memory and how the fitness is evaluated there. Section 4 gives a description of the Deterministic Crowding Genetic Algorithm. After experimental results and discussion are given in Section 5, we present concluding remarks.

2 Fitness Landscape

The concept of the fitness landscape was first introduced by Wright [**?**] in 1932 to study biological evolutionary processes. The definition was such as *"a collection of genotypes arranged in an abstract metric space, with each genotype next to those other genotypes which can be reached by a single mutation, as well as a value assigned to each genotype"* [**?**]. Since then, this concept has been used not only in evolutionary biology but also in chemistry, physics, computer science and so on.

In chemistry, for example, a molecule can be represented as a string of N letters with each letter being chosen from an alphabet of size k [**?**]. Twenty amino acids ($k = 20$) for proteins or four nucleotides ($k = 4$) for nucleic acids can be considered as examples of the alphabet. The k^N possible combinations of the letters construct a configuration space of the string. Then, for example, the free energy of RNA folding into secondary structures [**?**], or the ability of peptides to bind to a particular substrate to catalyze a specific reaction [**?**] is assigned as a fitness value to each configuration. In physics, the Hamiltonian energy of Ising spins defines a fitness landscape on the configuration space of N spins, where each spin takes the value either 1 or -1 ($k = 2$). Bray and Moore [**?**] argued about the number and distribution of meta-stable states (local optima) of the Hamiltonian energies.

To explore these fitness landscapes, we need a rule by which a point in the space moves to one of its neighbors. Then, consecutive movements of a point to the neighbors form a *walk* on the landscape. Macken et al. [**?**] used *random point mutation* that changes a single letter in the string to specify neighbors of the string. Then, by sampling points along an *evolutionary walk* in which point moves to the *firstly found fitter* neighbor, they studied the statistical properties of the landscape defined by the chemical affinity of antibody for antigen in immune response. Weinberger [**?**] used two different walks: *"gradient walk"* in which the walker steps to the *best* of its neighbors and *"random adaptive walk"* in which the next step is chosen *at random* from the set of better neighbors, to investigate the Kauffman's NK landscape which is a model formulated in more general form [**?**].

In this paper, fitness landscape is defined on all the possible configurations of bipolar patterns, sequences of $+1$ and -1, according to how each configuration is stored as associative memory on a network.

3 Neural Network Model of Associative Memory

Associative memory is a dynamical system which has a number of stable states with a domain of attraction around them [?]. If the system starts at any state in the domain, it will converge to the stable state. Hopfield [?] proposed a fully connected neural network model of associative memory in which information is stored by being distributed among neurons, and we can retrieve the information from dynamically relaxed neurons' states. In the model, some of the appropriate configurations of synaptic weights give the network a function of associative memory. Namely, the network stores a given set of bipolar patterns as attractors.

The Hopfield model consists of N neurons and N^2 synapses. Each neuron state is either active $(+1)$ or quiescent (-1). When an arbitrary N-bit *bipolar pattern* is given to the network as an initial state, the dynamical behavior of neurons' states afterwards are characterized by the strengths of the N^2 synapses. The synaptic strengths are called *weights*, and the weight from neuron j to neuron i is denoted as w_{ij}. Provided the synaptic weights are determined appropriately, network can store some number of patterns as attractors. Hopfield employed the so-called Hebbian rule [?] to prescribe the weights. That is, to store p bipolar patterns ξ^ν:

$$\xi^\nu = (\xi_1^\nu, \cdots, \xi_N^\nu), \qquad \nu = 1, \cdots, p,$$

the weight values are determined as:

$$w_{ij} = \frac{1}{N} \sum_{\nu=1}^{p} \xi_i^\nu \xi_j^\nu \quad (i \neq j), \quad w_{ii} = 0.$$

An instantaneous state of a neuron is updated asynchronously (one neuron at a time) as:

$$s_i\,(t+1) = sgn\left(\sum_{j \neq i}^{N} w_{ij} s_j(t)\right),$$

where $s_i(t)$ is a state of the i-th neuron at time t. If an initial state converges to one of the stored patterns ξ^ν as an equilibrium state, then the pattern is said to be *recalled.* Furthermore, if an initial state chosen from the stored patterns remains unchanged from the beginning, then the pattern is said to be stored as a *fixed point.* When a pattern is stored as a fixed point, a noisy and/or partial input of the pattern is expected to result in the relaxation into the perfect form of the pattern.

As stated earlier, there have been basically two different approaches in the analysis of the Hopfield model: exploration of the pattern space and the weight space. To be more specific, the former is an analysis of the Hamiltonian energy as a function of all the possible configurations of bipolar pattern to be given to the network, where the synaptic weights are pre-specified using a learning algorithm, usually the Hebb's rule, so that the network stores a set of p given patterns. In this context, the model for $p = 1$ corresponds to the Mattis model of spin-glass [?], in which the Hamiltonian energy has two minima, while the model for infinitely large p corresponds to the Sherrington-Kirkpatrick model

[?], in which the synaptic weights become Gaussian random variables. Analyses of the former type have been made in between these two extreme cases (see Amit [?]). The latter analysis, on the other hand, was addressed by Gardner [?]. She discussed the optimal weight configurations for a *fixed* number of given patterns in terms of the volume of the solutions in weight space, suggesting that the volume shrinks to vanish when p approaches to $2N$. In short, the former approach searches for the optimal pattern configurations which minimize the Hamiltonian energy *in pattern space with the weights being fixed*, while the latter searches for the weight configurations *in weight space* that optimally store a set of given *fixed patterns*.

So far, we have studied the model with the latter approach. We have explored fitness landscapes of the model *defined on weight space*, and have found many solutions that store more patterns or store them with larger basin of attractions than, e.g., the Hebbian synaptic weights [?, ?, ?, ?]. In this paper we use the model with the latter approach to use it as a test function of evolutionary algorithms.

Fitness Evaluation. Since we explore the fitness landscape defined on the pattern space in this paper, the fitness function of an arbitrary configuration of N-bit bipolar pattern should be defined. Here we evaluate the fitness of a pattern according to how the instantaneous neurons' states $s_i(t)$ after the pattern is given to the network, are similar to either of the stored patterns. The similarity as a function of time is defined as:

$$m^{\nu}(t) = \frac{1}{N}\sum_{i=1}^{N} \xi_i^{\nu} s_i^{\nu}(t).$$

The temporal average of $m^{\mu}(t)$ is calculated for each stored pattern, and the maximum value among them is assigned to the input pattern as fitness. Thus, the fitness value f is evaluated as follows:

$$f = \max\{ \frac{1}{t_0}\sum_{t=1}^{t_0} m^{\nu}(t) \ \mid \ \mu = 1, 2, \cdots, p\}.$$

In this paper, t_0 is set to $2N$, twice the number of neurons. Note that the fitness 1 implies the pattern is a fixed point attractor, while fitness less than 1 includes many other cases.

4 Niching Methods

Niching genetic algorithm is a genetic algorithm (GA) [?] that was devised to locate multiple solutions simultaneously. To do this, *sharing* [?], for example, derates the fitness value of each individual using *sharing function* which reflects the number of similar individuals in the population. Or *crowding* [?] reduces the number of similar individuals in the population by replacing some of the individuals with new individuals. Here, we employ the *Deterministic Crowding GA* proposed by Mahfoud [?] because of its niching capability [?] as well as the simplicity for implementation.

In each generation, as neatly summarized by Mahfoud [?], the current population is reproduced as follows.

(1) Choose two parents, p_1 and p_2, at random, with no parent being chosen more than once.
(2) Produce two children, c_1 and c_2, with uniform crossover [**?**].
(3) Mutate the children by flipping bit chosen at random with probability p_m, yielding c_1' and c_2'.
(4) Replace parent with child as follows:
   - IF $\quad d(p_1, c_1') + d(p_2, c_2') > d(p_1, c_2') + d(p_2, c_1')$
     * IF $\quad f(c_1') > f(p_1)$ THEN $\quad$ replace p_1 with c_1'
     * IF $\quad f(c_2') > f(p_2)$ THEN $\quad$ replace p_2 with c_2'
   - ELSE
     * IF $\quad f(c_2') > f(p_1)$ THEN $\quad$ replace p_1 with c_2'
     * IF $\quad f(c_1') > f(p_2)$ THEN $\quad$ replace p_2 with c_1'

where $d(\zeta_1, \zeta_2)$ is the Hamming distance between two points (ζ_1, ζ_2) in pattern configuration space. The process of producing child is repeated until all the population have taken part in the process. Then the cycle of reconstructing a new population and restarting the search is repeated until all the global optima are found or a set maximum number of generation has been reached.

5 Results and Discussion

Experiments were carried out on networks with 49 neurons. For a given set of p patterns, a configuration of synaptic weights in a network is calculated by the Hebb's rule. Thus, the network with 49 neurons now stores these p patterns as fixed point attractors, so far as p dose not exceed the storage capacity. Namely, the fitness landscape defined on pattern space has p global peaks exactly at the location of the patterns. Then the Deterministic Crowding GA searches for these attractors. The parameters of the GA employed in this paper are as follows. The population size is 200; mutation probability p_m is set to 0.05; and the number of iterations is limited to a maximum of 12,000 generations.

Deterministic Crowding GA. Thus, we apply the Deterministic Crowding GA to this problem. We investigated how many of the embedded patterns can be located. For a specific number of patterns p, starting with $p = 2$, we studied 30 runs with varying random number seed at the start of a run. The experiments are iterated with p being incremented if any of the 30 runs locates *all* the embedded solutions. As results, we observed that when $p = 2$, 19 out of 30 runs locate all the embedded patterns; then as p increases, the number of successful runs decreases, i.e., when $p = 3, 4, 5, 6$ the number of successful trials out of 30 runs was 14, 8, 3, 0, respectively. The results for $p = 4, 5$, and 6 of the numbers of individuals that converged on each of the given patterns when a GA run terminated are shown in Table 1.

We can see that all the individuals eventually find some different niches for $p < 6$. Then, how the number of individuals in each of these niches changes as an evolution proceeds? An example is shown in Figure **??**. The figure shows that all the individuals in the Deterministic Crowding GA converge to any of the solutions.

Table 1. Number of individuals converged on each attractor under an evolution by the Deterministic Crowding GA for $p = 4, 5$, and 6.

	number of located attractors						
p	#1	#2	#3	#4	#5	#6	total
4	40	96	27	59	-	-	200
4	68	26	24	82	-	-	200
4	109	7	20	64	-	-	200
4	23	52	76	49	-	-	200
4	23	43	78	56	-	-	200
4	18	79	11	92	-	-	200
4	54	8	7	131	-	-	200
4	31	28	38	103	-	-	200
5	85	71	5	34	5	-	200
5	42	54	68	11	25	-	200
5	15	102	26	52	5	-	200
6	33	17	58	50	0	42	200

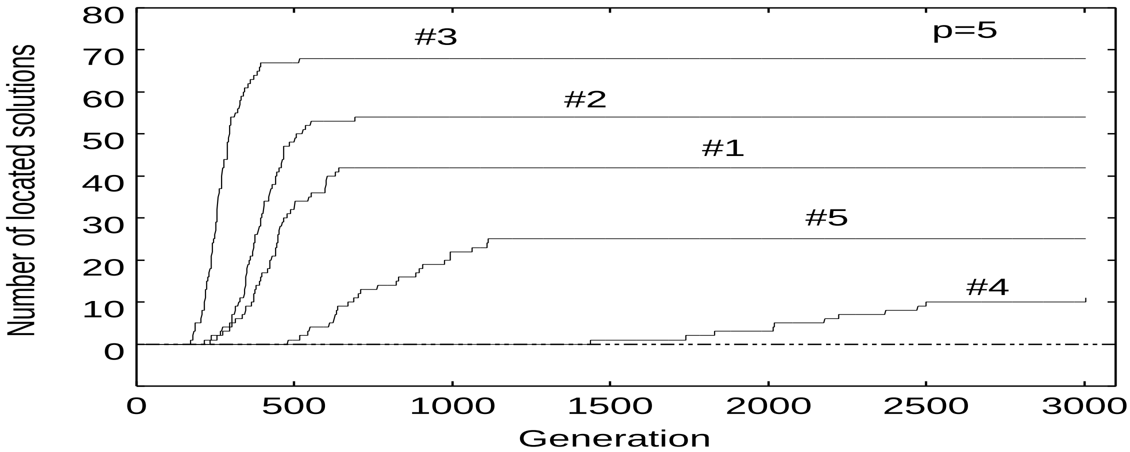

Figure 1. Number of individuals converged on each attractor under an evolution by the Deterministic Crowding GA for $p = 5$.

A Visualization of the Search. The Principal Component Analysis (PCA) is a dimension reduction technique where points in multi-dimensional space are projected on fewer mutually orthonormal axes selected so that the variance of the projected points to each axis is maximal.

Here, we apply the PCA to the population at some stages of the evolution by the Deterministic Crowding GA. The points representing the individuals in the population sampled from the evolution shown in Figure **??** are projected on a line of the maximal variance. In Figure **??**, points sampled at generation 100, 500, 1000, 1500, 2000, 2500, 2800 and 3100 are shown. The number of individuals at each of these eight generations is

Table 2. Number of individuals converged on each attractor during an evolution by the Deterministic Crowding GA for $p = 5$.

generation	number of located attractors #1	#2	#3	#4	#5	total
100	0	0	0	0	0	0
500	32	48	67	0	1	148
1000	42	54	68	0	22	186
1500	42	54	68	1	25	190
2000	42	54	68	3	25	192
2500	42	54	68	9	25	198
2800	42	54	68	10	25	199
3100	42	54	68	10	25	200

summarized in Table **??**. As we can see in Figure **??**, the individuals in the population are distributed almost at random at generation 100. Then as generation proceeds, individuals move to approach any of the five attractors. However, at generation 2800 where 198 out of 200 individuals reached any of the five attractors, we can only see 4 points on the line. Furthermore, at generation 3100 where all the 200 individuals converged on any of the five attractors, we see only 3 points. This is because two attractors at generation 2800 and three attractors at generation 3100 happen to be on the axis perpendicular to the line on which the points are projected. In Figure **??**, the PCA projection of the five attractors on the 2-dimensional plane. This is the deficient aspect of the visualization by the PCA.

We also report elsewhere another visualization method [**?**]

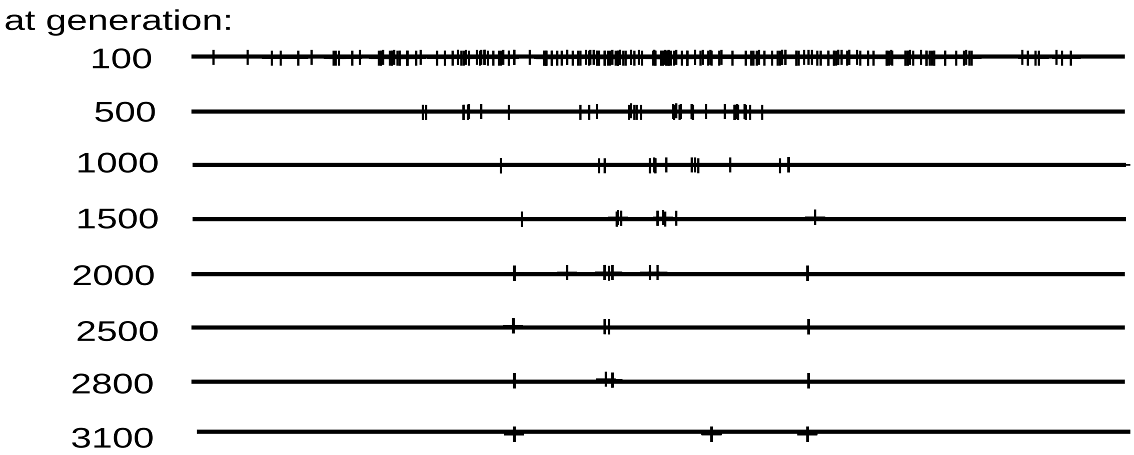

Figure 2. One-dimensional points mapped from 49D search space along an evolution.

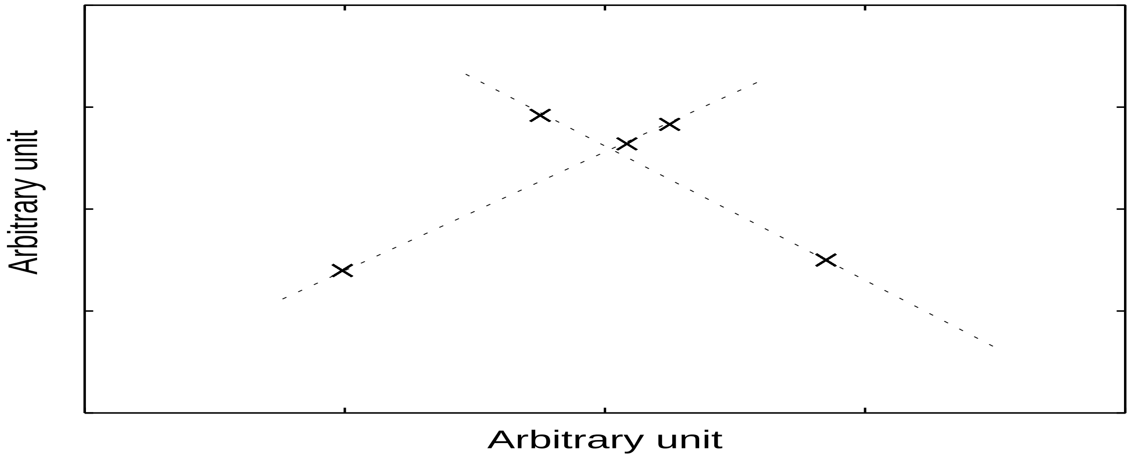

Figure 3. Two-dimensional points mapped from 49D solutions at generation 3100.

6 Conclusion

We have described an application of the Deterministic Crowding GA, one of multi-modal function optimization techniques, to the fully-connected neural network model of associative memory. What we are concerned with is to obtain multiple solutions of weight configuration in weight space simultaneously, and we think of the Deterministic Crowding GA as one of the candidate tools for the purpose. However, because of the difficulty in searching for the solutions in continuous weight space of high-dimensionality, the fitness landscape defined on pattern space was explored in this paper as a preliminary stage for that. As expected, the Deterministic Crowding GA converges to multiple solutions simultaneously on the condition that the number of solutions is small enough. In our experiments with networks with 49 neurons the search space to be explored is vertices of the 49-dimensional hyper cube and the Hebbian learning algorithm creates up to 8 solutions in the space. We found that the Deterministic Crowding GA can locate all the solutions embedded in the landscape unless the number of stored patterns is larger than five.

The Principal Component Analysis was also studied to visualize the search space. Here, the candidate solutions in the 49-dimensional pattern space are mapped on to the one-dimensional line. Though we cannot see all the original points in the high-dimensional space on the projected line, we somehow can overview the convergence behavior.

The approach we have taken in this paper, that is, the exploration in pattern space to search for solutions, can be extended in the problem of searching for the solutions in weight space. However, the multimodal technique employed in our current experiments does not seem to have the enough ability to locate multiple solutions in weight space. We leave the improvement of the multi-modal technique as well as visualization tool to our future research.

Although the analysis of the fully-connected network model of associative memory is rather classical, many issues are still unknown. Our goal of applying multimodal GA to the model is to address one of these issues, that is, the distribution of solutions in weight space. We believe these studies using evolutionary algorithms shed new light on the analysis of the model.

Acknowledgments

We thank Peter Davis at Advanced Telecommunication Research Institute for providing us great insights into the dynamics of the Hopfield neural networks. Thanks also goes to Shigeaki Tagashira at Nara Institute of Science and Technology with whom we made a stimulating discussion about how multi-dimensional space is visualized by PCA.

References

1. A. Imada and K. Araki, Genetic Algorithm Enlarges the Capacity of Associative Memory, In *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 413–420. (1995)
2. A. Imada and K. Araki, Evolution of Hopfield Model of Associative Memory by the Breeder Genetic Algorithm, In *Proceedings of the 7th International Conference on Genetic Algorithms*, pp. 784–791. (1997)
3. A. Imada and K. Araki, Application of an Evolution Strategy to the Hopfield Model of Associative Memory, In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 679–683. (1997)
4. A. Imada and K. Araki, Basin of Attraction of Associative Memory as it Evolves from Random Weights, In *Proceedings of the 1st Asia-Pacific Conference on Simulated Evolution and Learning*, pp. 271-278. (1996)
5. S. W. Mahfoud, A Comparison of Parallel and Sequential Niching Methods, In *Proceedings of the 2nd Parallel Problem Solving from Nature*, pp. 27–36. (1992)
6. S. Wright, The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution, In *Proceedings of the 6th International Congress of Genetics*, pp. 356–366. (1932)
7. P. F. Stadler and G. P. Wagner Algebraic Theory of Recombination Spaces, *Evolutionary Computation* 5(3), pp. 241–275
8. C. Macken, P. Hagan and A. Perelson Evolutionary Walks on Rugged Landscapes, *SIAM Journal of Applied Mathematics* 51, pp. 799–827. (1991)
9. W. Fontana and P. Schuster A Computer Model of Evolutionary Optimization, *Biophysical Chemistry* 26, pp. 123–147. (1987)
10. J. Maynard Smith Natural Selection and the Concept of a Protein Space, *Nature* 225, pp. 563–564. (1970)
11. A. Bray and M. Moore Metastable States in Spin Glasses, *Journal of Physics* C: Solid State Physics 13, pp. L469–L476. (1980)
12. E. D. Weinberger Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference, *Biological Cybernetics* 63, pp. 325–336. (1990)
13. S. A. Kauffman and S. Levin Towards a General Theory of Adaptive Walks on Rugged Landscapes, *Journal of Theoretical Biology* 128, pp. 11–45. (1987)

14. J. Komlós and R. Paturi, Convergence Results in an Associative Memory Model, In *Neural Networks* 1, pp. 239–250. (1988)

15. J. J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, In *Proceedings of the National Academy of Sciences (USA)* 79, pp. 2554-2558. (1982)

16. D. O. Hebb, *The Organization of Behavior. Wiley.* (1949)

17. D. C. Mattis, Solvable Spin Systems with Random Interactions, In *Physics Letters* 56A, pp. 421–422. (1976)

18. D. Sherrington, and S.Kirkpatrick, Solvable Model of a Spin Glass, In *Physical Review Letters* 35, pp. 1792–1796. (1975)

19. D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks*, Cambridge University Press. (1989)

20. E. Gardner, The Phase Space of Interactions in Neural Network Models, In *Journal of Physics* 21A, pp. 257–270. (1988)

21. J. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press. (1975)

22. D. E. Goldberg and J. Richardson, Genetic Algorithms with Sharing for Multimodal function Optimization, In *Proceedings of 2nd International Conference on Genetic Algorithms*, pp. 41-49. (1987)

23. K. A. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems, Ph.D. theses University of Michigan. (1975)

24. S. W. Mahfoud, A Comparison of Parallel and Sequential Niching Methods, In *Proceedings of the 2nd Parallel Problem Solving from Nature*, pp. 27–36. (1992)

25. S. W. Mahfoud, A Comparison of Parallel and Sequential Niching Methods, In *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 136–143. 1995)

26. G. Syswerda, Uniform Crossover in Genetic Algorithms, In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp2–9. (1989)

27. A. Imada Neural Network Model of Associative Memory: To Visualize Solutions in Weight Space. (to submit) 1999.