An Approach to Evolutionary Robotics Using a Genetic Algorithm with a Variable Mutation Rate Strategy

Yoshiaki Katada[1], Kazuhiro Ohkura[2], and Kanji Ueda[3]

[1] Graduate School of Science and Technology, Kobe University, Kobe, JAPAN
[2] Faculty of Engineering, Kobe University, Kobe, JAPAN
Phone/Fax: +81-78-803-6135
Email: {katada, ohkura}@rci.scitec.kobe-u.ac.jp
[3] RACE (Research into Artifacts, Center for Engineering),
The University of Tokyo, Meguro, JAPAN
Phone: +81-3-5453-5887 Fax:+81-3-3467-0648
Email: ueda@race.u-tokyo.ac.jp

Abstract. Neutral networks, which occur in fitness landscapes containing neighboring points of equal fitness, have attracted much research interest in recent years. In recent papers [20, 21], we have shown that, in the case of simple test functions, the mutation rate of a genetic algorithm is an important factor for improving the speed at which a population moves along a neutral network. Our results also suggested that the benefits of the variable mutation rate strategy used by the operon-GA [5] increase as the ruggedness of the landscapes increases. In this work, we conducted a series of computer simulations with an evolutionary robotics problem in order to investigate whether our previous results are applicable to this problem domain. Two types of GA were used. One was the standard GA, where the mutation rate is constant, and the other was the operon-GA, whose effective mutation rate at each locus changes independently according to the history of the genetic search. The evolutionary dynamics we observed were consistent with those observed in our previous experiments, confirming that the variable mutation rate strategy is also beneficial to this problem.

1 Introduction

Selective neutrality has been found in many real-world applications of artificial evolution, such as the evolution of neural network controllers in robotics [1, 2] and on-chip electronic circuit evolution [3, 4]. This characteristic is caused by highly redundant mappings from genotype to phenotype or from phenotype to fitness. With these kinds of problems, redundancy is inevitable. Even for problems where redundancy is largely absent, it may be useful to introduce it. A number of researchers have been trying to improve the performance of artificial evolution on more traditional problems by incorporating redundancy in genotype to phenotype mappings[5–8]. Neutrality is also found in natural systems, and has

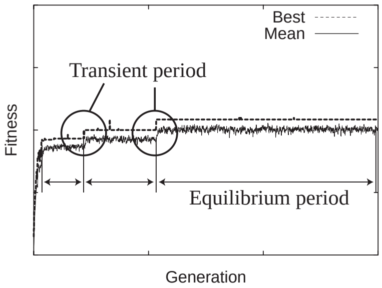


Fig. 1. Typical evolutionary dynamics on a fitness landscape featuring neutral networks, which can be classified into transient periods and equilibrium periods.

been of particular interest to evolutionary theorists [9] and molecular biologists [10, 11].

Landscapes which include neutrality have been conceptualized as containing *neutral networks* [6, 12, 13]. This concept is central to the majority of research in this field. Harvey [12] first introduced the concept of neutral networks into the GA community. His definition is as follows: "A neutral network of a fitness landscape is defined as a set of connected points of equivalent fitness, each representing a separate genotype: here connected means that there exists a path of single (neutral) mutations which can traverse the network between any two points on it without affecting fitness."

Evolutionary dynamics on neutral networks can be classified into transient periods and equilibrium periods (Fig. 1) [14, 15]. During an equilibrium period, the population is clustered in genotype space around the *dominant phenotype*, analogously to *quasi-species* [16], and moves around until it finds a portal to a neutral network of higher fitness. The discovery of a portal leads to a transient period, which is expected to be very short in comparison to an equilibrium period. It has been shown that there is a clear transition in evolutionary dynamics for populations on neutral networks over the mutation rate range. At a very low mutation rate, the population is maintained in a cluster on the neutral network. As the mutation rate increases, the population gradually loses the current network. That is, some individuals fall to lower neutral networks. At a certain critical mutation rate, the whole population will lose the current neutral network. This mutation rate is called the *phenotypic error threshold*[4] [14, 17, 18].

Generally, the error threshold sets the upper limit for a mutation rate that will enable efficient search. This implies that if we adopt a constant mutation rate strategy, we should set a relatively low mutation rate so as to avoid any error threshold effects during the process. From a practical point of view, however, it would be efficient to shorten the equilibrium period which dominates the whole computation (Fig. 1). Additionally, in landscapes which include ruggedness, individuals can easily get trapped on local optima if there is a low mutation rate and high selection pressure. It has been demonstrated in a tunably neutral NK

[4] These concepts originate from molecular evolution [10, 11]

landscape [17,19] that increasing neutrality does not affect the ruggedness, although it does reduce the number of local optima [13,17,19]. This means that the effects of ruggedness must be taken into account even if landscapes include neutral networks. Using a high mutation rate can shorten equilibrium periods and help a population avoid becoming trapped on local optima. However, as noted above, using a high mutation rate can be counterproductive because of the effects of error thresholds. One approach to overcoming these problems would be to adopt *variable mutation rate strategies*, which change the effective mutation rate adaptively during the process of evolution.

Recently, we have investigated the effect of mutation rate and selection pressure on the speed of population movement on very simple neutral networks with different levels of neutrality [20]. We also examined the performance of GAs using terraced NK landscapes with different levels of ruggedness and different selection pressures. [21]. Our results can be summarized as follows:

- For a fixed population size, the speed of a population plotted as a function of the mutation rate yields a concave curve with an optimal mutation rate and an error threshold.
- Increasing the selection pressure will improve the speed at which a population moves on a neutral network.
- The variable mutation rate strategy used by the operon-GA [5] is not only beneficial with simple test functions but also with complex test functions. The benefits increase as the ruggedness of the landscapes increases.

We are interested in whether these observations are consistent with more complex problems. This is because we want to solve complex real-world problems. This paper investigates how well our previous results apply to the evolution of artificial neural networks for robot control by comparing a standard GA [22] and the operon-GA on an evolutionary robotics task using a simulated robot. The paper is organized as follows. The next section describes the neural networks adopted in a robot control problem. Section 3 defines the robot control problem where the evolved neural networks are evaluated. Section 4 gives the results of our computer simulations. Section 5 discusses the relationship between the correlation of the landscape and the overall GA performance. Conclusions are given in the last section.

2 The Neural Controller –Spike Response Model–

The agent's behavior is controlled by a *spike response model* network [23], which is a form of *Pulsed Neural Network* (PNN). A neuron emits a spike when the total amount of excitation due to incoming excitatory and inhibitory spikes exceeds its firing threshold, θ. After firing, the membrane potential of the neuron is set to a low negative voltage, it then gradually returns to its resting potential; during this refractory period, a neuron cannot emit a new spike. The function η_i, accounting for neuronal refractoriness, is given by:

$$\eta_i(r) = -\exp(-\frac{r}{\tau_m})H(r) \tag{1}$$

Here $r = t - t_i^{(f)}$ is the difference between the time t and the time of firing $t^{(f)}$ of neuron i, τ_m is a membrane time constant and $H(r)$ is the Heaviside step function which vanishes for $r < 0$ and gives a value of 1 for $r > 0$.

The function ε_{ij} describes the response to postsynaptic spikes: (2).

$$\varepsilon_{ij}(r) = [\exp(-\frac{r - \Delta^{ax}}{\tau_m})(1 - \exp(-\frac{r - \Delta^{ax}}{\tau_s}))]H(r - \Delta^{ax}) \qquad (2)$$

where τ_s is a synaptic time constant, Δ^{ax} is the axonal transmission delay.

The membrane potential of a neuron i at time t is given by:

$$u_i(t) = \sum_{t_i^{(f)} \in F_i} \eta_i(t - t_i^{(f)}) + \sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in F_j} \omega_{ij}\varepsilon_{ij}(t - t_j^{(f)}) \qquad (3)$$

where F_i is the set of firing times in a neuron i. The neuron i may receive the input from presynaptic neurons $j \in \Gamma_i$. The weight ω_{ij} is the strength of the connection from the j^{th} neuron, and scales the amplitude of the response given in eq.(2).

3 The Task and the Fitness Function

The control task used in this paper was motion pattern discrimination [24], and is based on a task originally implemented by Beer [25]. The agent must discriminate between two types of vertically falling object based on the object's period of horizontal oscillation; it must catch (i.e., move close to) falling objects that have a long period whilst avoiding those with a short period (see Fig. 2). An array of proximity sensors allow the agent to perceive the falling objects. If an object intersects a proximity sensor, the sensor outputs a value inversely proportional to the distance between the object and the agent. The agent can move horizontally along the bottom of the arena. In our experiment, the agent of diameter 30 had 7 proximity sensors of maximum range 220 uniformly distributed over a visual angle of 45 degrees. The horizontal velocity of the agent was proportional to

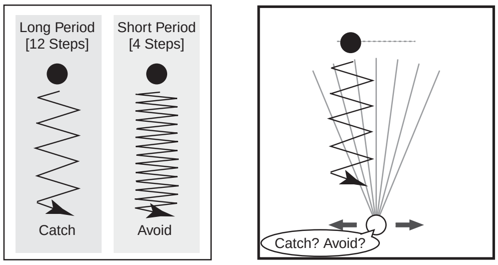


Fig. 2. Experimental setup for the discrimination of the motion patterns. Two kinds of period used in the discrimination experiments (left) and the agent in the arena with its array of the proximity sensors (right).

the sum of the opposing horizontal forces produced by a pair of effectors. It has maximum velocity of 8. Each falling object was circular, with diameter 30, and dropped from the top of the arena with a vertical velocity of 4, a horizontal amplitude of 30 and an initial horizontal offset of ±50. An object's horizontal velocity was ±10 (12 steps in a period) for a long period and ±30 (4 steps in a period) for a short period.

The performance measure to be maximized was as follows:

$$Fitness = 1000 \sum_{i=1}^{NumTrials} \frac{P_i}{NumTrials} \qquad (4)$$

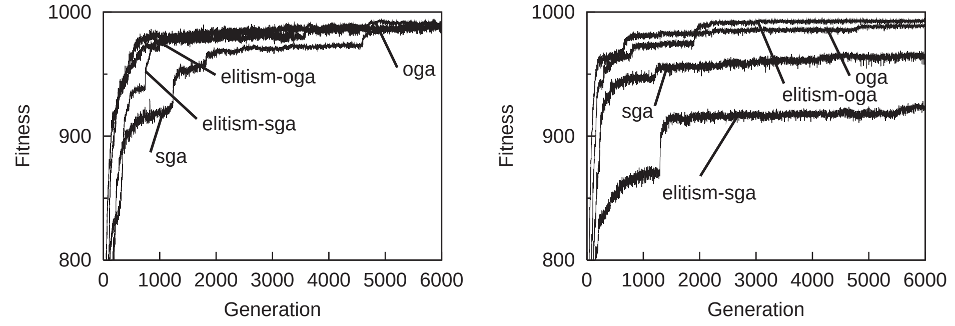
where $P_i = 1 - d_i$ for a long period and $P_i = d_i$ for a short period, $d_i = 1$ when $hd_i > 60$ and $d_i = hd_i/60$ when $hd_i \leq 60$, hd_i is the final horizontal distance between the center of the agent and the object, and $NumTrials$ is the number of trials for an individual (8 trials for each period).

4 Computer Simulations

4.1 Simulation Conditions

For this experiment, an agent's controller was a PNN with 7 sensory neurons, 2 fully interconnected motor neurons and N_h fully interconnected hidden neurons, where $N_h \in \{1, 10\}$. The network's connection weights and the firing threshold for each neuron were genetically encoded and evolved. The total number of parameters was either 33 ($N_h = 1$) or 240 ($N_h = 10$). The parameters were mapped linearly onto the following ranges: connection weights, $\omega \in [-1.0, 1.0]$, and thresholds, $\theta \in [0.0, 3.9]$. The parameters of the neurons and synapses (see section 2) were set as follows: $\tau_m = 4$, $\tau_s = 10$, $\Delta^{ax} = 2$ for all neurons and all synapses in the network following the recommendations given in [26]. Computer simulations were conducted using populations of size 50. Each individual was encoded as a binary string with 10 bits for each parameter. Therefore, the total length of the genotype was either $L_1 = 330$ ($N_h = 1$) or $L_{10} = 2400$ ($N_h = 10$). The standard GA (SGA) and the operon-GA (OGA) were employed to evolve the PNN parameters. The OGA uses standard bit mutation and five additional genetic operators: *connection*, *division*, *duplication*, *deletion* and *inversion*. The probabilities for genetic operations were set at 0.3 for *connection* and *division*, 0.2 for *duplication* and 0.05 for *deletion* and *inversion*, based on our previous results in [21]. The length of the value list in a locus was 6. The genetic operation for the SGA was standard bit mutation. For both GAs, the per-bit mutation rate, q, was set at $1/L$ (0.003 for L_1 and 0.000416 for L_{10}). Crossover was not used for either GA, following Nimwegen's suggestion [14]. Tournament selection was adopted. *Elitism*[5] was optionally applied. The tournament size, s, was set at $\{2, 6\}$ because the SGA prefers low selection pressure while the OGA prefers high selection

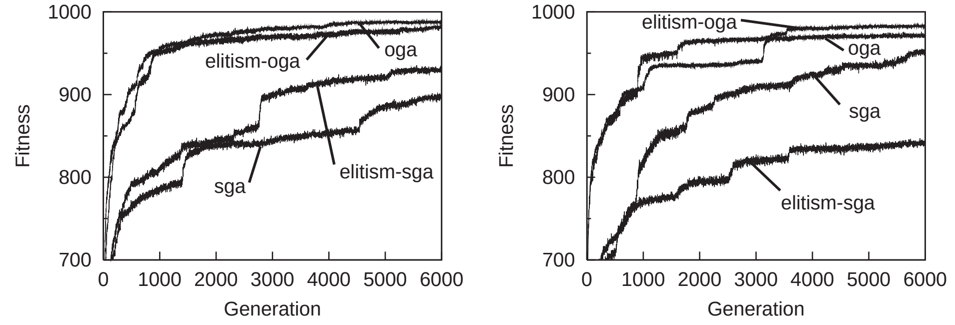
[5] The fittest individual of each generation was passed un-mutated to the next generation (if several individuals had the highest fitness, one was randomly chosen.)



(a) $s = 2$

(b) $s = 6$

Fig. 3. The maximum fitness at each generation for $N_h = 1$



(a) $s = 2$

(b) $s = 6$

Fig. 4. The maximum fitness at each generation for $N_h = 10$

pressure. A generational model was used. Each run lasted 6,000 generations. We conducted 10 independent runs for each of the sixteen conditions. All results were averaged over 10 runs.

4.2 Simulation Results

Fig. 3 shows the maximum fitness at each generation for the SGA and OGA, with and without elitism, for controllers with $N_h = 1$. Fig. 3(a) and 3(b) show the results for the four GA conditions for $s = 2$ and 6 respectively. For $s = 2$, fitness increased faster with the OGA than with the SGA in the early generations. In the final generation, there was no significant difference between the SGA and the OGA. For $s = 6$, the SGA was trapped on local optima, whereas the OGA continued to find better regions of the search space. In addition, the SGA performed better without elitism than with it. These results are consistent with the results obtained using terraced NK landscapes [21]. With respect to final generation fitnesses, there was no significant difference between the SGA with $s = 2$ and the OGA for $s = 6$. However, a closer examination reveals that during the process of evolution the OGA with $s = 6$ performed better than the SGA with $s = 2$ and elitism.

Fig. 4 shows the maximum fitness at each generation for the SGA and OGA, with and without elitism, for $s = 2$ and 6 with $N_h = 10$. With $N_h = 10$, differences between the SGA and the OGA were much more pronounced than with $N_h = 1$. Even for $s = 2$, fitness increased faster for the OGA than for the SGA (Fig. 4(a)). This is consistent with the results obtained using simple neutral networks when the mutation rate is below the optimal mutation rate [20]. As with $N_h = 1$, for $s = 6$, the SGA with elitism was trapped on local optima (Fig. 4(b)), whereas the OGA continued to find better regions; also as before, the SGA performed better without elitism than with it. The OGA for $s = 6$ also outperformed the SGA for $s = 2$.

Under all conditions, the OGA performed better than the SGA on this task, either by achieving higher final fitnesses, or by achieving high fitnesses faster, or both. This shows that the OGA's variable mutation rate strategy was beneficial on this problem.

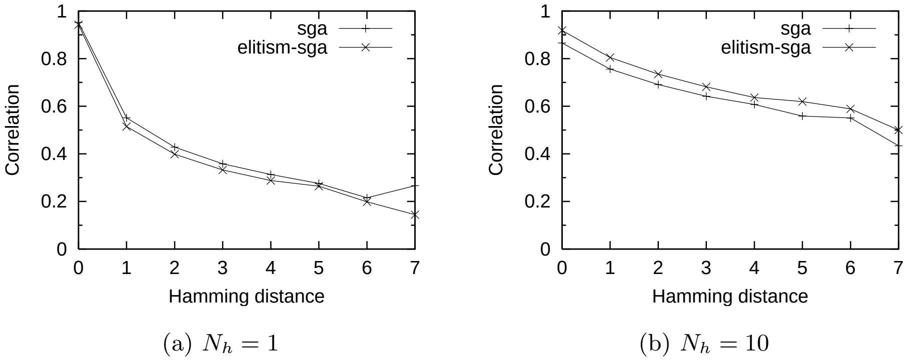
5 Discussion

The evolutionary dynamics observed in these experiments can be explained in the same way as in [21].

The evolutionary dynamics that were observed showed phases of neutral evolution, implying that the fitness landscapes include neutral networks. However, large fluctuations that sometimes cause the best individuals to be lost were not observed under any of the four GA conditions. That is, there was no influence of the error threshold at the mutation rate $q = 1/L$. Therefore, we can assume that the effective mutation rate of $q = 1/L$ was below the error threshold under each condition.

The correlation of the landscapes was analyzed in order to investigate overall GA performance. Fig. 5(a) and 5(b) show the correlation coefficient [27] as a function of the Hamming distance between parents and offspring for the SGA with and without elitism, with $N_h = 1$ and 10 respectively. They suggest high fitness correlation in both landscapes, with the $N_h = 10$ landscape being more highly correlated that the $N_h = 1$ landscape.

As predicted, with $s = 6$, the SGA with and without elitism was trapped on local optima when $N_h = 1$, due to the low mutation rate and high selection pressure. With $s = 6$ and $N_h = 10$, the SGA with elitism was also trapped on local optima. However, the SGA without elitism for $s = 6$ and $N_h = 10$ was not obviously trapped. Based on the analysis of ruggedness shown in Fig. 5(b), it seems likely that fitnesses would continue improving if the runs were extended beyond their final generation. Further computer simulations were therefore conducted



(a) $N_h = 1$

(b) $N_h = 10$

Fig. 5. The correlation coefficient as a function of the Hamming distance between parents and offspring for the SGA

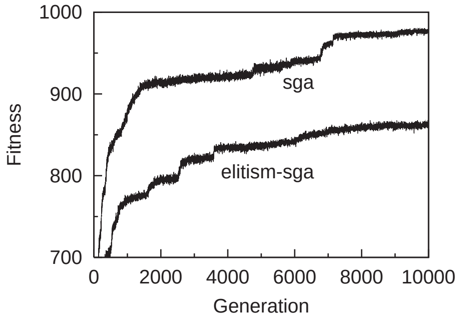


Fig. 6. The maximum fitness over 10,000 generations for the SGA for $s = 6$ and $N_h = 10$

in order to observe the SGA runs over an additional 4,000 generations. Fig. 6 shows the maximum fitness for 10,000 generations of the SGA with $N_h = 10$. The SGA without elitism continued to find better regions of the search space. This indicates that the SGA without elitism can escape from local optima with this level of ruggedness.

When compared on the same landscapes, the OGA continued to find much better regions of search space than the SGA. The continued improvement observed with the OGA was due to the online adaptation of mutation rates during the process of evolution. In addition to this, with the OGA, the effective mutation rate will have been below the error threshold even with low selection pressure (i.e. when $s = 2$). This is why the variable mutation rate strategy of the OGA was a better approach on this problem with both high and low selection pressure.

6 Conclusions

In this work, we applied the standard GA and the operon-GA to evolution of artificial neural networks for robot control, and investigated their performance using different selection pressures. Our results can be summarized as follows:

- This evolutionary robotics problem does show phases of neutral evolution.
- The standard GA with low selection pressure and the operon-GA were able to continually find better regions of the search space.
- The standard GA can easily get trapped on local optima under conditions of high selection pressure and a low mutation rate.
- The benefits of the variable mutation rate strategy used by the operon-GA were more pronounced with a larger genetic search space.

These results are consistent with the results of our previous experiments using simple neutral networks and terraced NK landscapes.

The fitness landscape of this evolutionary robotics problem is relatively smooth. Future work will investigate whether these results are applicable to real-world problems which are expected to have more rugged landscapes.

References

1. Harvey, I.: Artificial Evolution for Real Problems. In Gomi, T. (ed.): Evolutionary Robotics: From Intelligent Robots to Artificial Life (ER'97), AAI Books (1997)
2. Smith, T., Husbands, P., O'Shea, M.: Neutral Networks in an Evolutionary Robotics Search Space. In Proceedings of the 2001 IEEE Congress on Evolutionary Computation (2001) 136–145
3. Thompson, A.: An Evolved Circuit, Intrinsic in Silicon, Entwined with Physics. In Proceedings of the First International Conference on Evolvable Systems: From Biology to Hardware (1996) 390–405
4. Vassilev, V. K., Fogarty, T. C., Miller, J. F.: Information Characteristics and the Structure of Landscapes. Evolutionary Computation, 8(1) (2000) 31–60
5. Ohkura, K.,Ueda, K.: Adaptation in Dynamic Environment by Using GA with Neutral Mutations. International Journal of Smart Engineering System Design, 2 (1999) 17–31
6. Ebner, M., Langguth, P., Albert, J., Shackleton, M., Shipman, R.: On Neutral Networks and Evolvability, In Proceedings of the 2001 IEEE Congress on Evolutionary Computation: CEC2001, IEEE Press (2001) 1–8
7. Knowles, J. D., Watson, R. A.: On the Utility of Redundant Encodings in Mutation-based Evolutionary Search. In Merelo, J.J., Admidis, P., Beyer, H.-G., Fernandes-Villacanas, J.-L., Schwefel, H.-P. (eds.): Proceedings of Parallel Problem Solving from Nature - PPSN VII, Seventh International Conference, LNCS 2439 (2002) 88–98
8. Rothlauf, F., Goldberg, D.: Redundant Representations in Evolutionary Computation. Evolutionary Computation, 11(4) (2003) 381–415
9. Kimura, M.: The Neutral Theory of Molecular Evolution, Cambridge University Press, New York (1983)
10. Huynen, M., Stadler, P., Fontana, W.: Smoothness within Ruggedness: The Role of Neutrality in Adaptation. In Proceedings of the National Academy of Science USA, 93 (1996) 397–401

11. Reidys, C., Stadler, P., Schuster, P.: Generic Properties of Combinatory Maps - Neutral Networks of RNA Secondary Structures. Bulletin of Mathematical Biology, 59 (1997) 339–397
12. Harvey, I., Thompson, A.: Through the Labyrinth Evolution Finds a Way: A Silicon Ridge. In Proceedings of the First International Conference on Evolvable Systems: From Biology to Hardware (1996) 406–422
13. Smith, T. Husbands, P. Layzell, P. O'Shea, M.: Fitness Landscapes and Evolvability. Evolutionary Computation, 10(1) (2002) 1–34
14. Nimwegen, E., Crutchfield, J., Mitchell, M.: Statistical Dynamics of the Royal Road Genetic Algorithm. Theoretical Computer Science, Vol. 229, No. 1 (1999) 41–102
15. Barnett, L.: Netcrawling - Optimal Evolutionary Search with Neutral Networks. In Proceedings of the 2001 IEEE Congress on Evolutionary Computation (2001) 30–37
16. Eigen, M., McCaskill, J., Schuster, P.: The Molecular Quasi-species. Advances in Chemical Physics, 75 (1989) 149–263
17. Barnett, L.: Tangled Webs: Evolutionary Dynamics on Fitness Landscapes with Neutrality. MSc. dissertation, School of Cognitive and Computing Sciences, Sussex University, UK (1997)
18. Nimwegen, E., Ctrutchfield, J.: Optimizing Epochal Evolutionary Search: Population-size Dependent Theory. SFI Working Paper 9810-090, Santa Fe Institute (1998)
19. Newman, M. Engelhardt, R.: Effect of Neutral Selection on the Evolution of Molecular Species. In Proceedings of the Royal Society of London B, Morgan Kaufmann, 256 (1998) 1333–1338
20. Katada, Y., Ohkura, K., Ueda, K.: Tuning Genetic Algorithms for Problems Including Neutral Networks -The Simplest Case: The Balance Beam Function-. In Proceedings of the 7th Joint Conference on Information Sciences (2003) 1657–1660
21. Katada, Y., Ohkura, K., Ueda, K.: Tuning Genetic Algorithms for Problems Including Neutral Networks -A More Complex Case: The Terraced NK Problem-. In Proceedings of the 7th Joint Conference on Information Sciences (2003) 1661–1664
22. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
23. Maass, W., Bishop, C.M.: Pulsed Neural Networks, MIT press (1998)
24. Katada, Y., Ohkura, K., Ueda, K.: Artificial Evolution of Pulsed Neural Networks on the Motion Pattern Classification System. In Proceedings of 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation(CIRA) (2003) 318–323
25. Beer, R.: Toward the Evolution of Dynamical Neural Networks for Minimally Cognitive Behavior. In Maes, P., Mataric, M., Meyer, J., Pollack, J., Wilson, S. (eds.): Proceedings of From Animals to Animats 4, MIT press (1996) 421–429
26. Floreano, D., Mattiussi, C.: Evolution of Spiking Neural Controllers. In Gomi, T. (ed.): Evolutionary Robotics: From Intelligent Robots to Artificial Life (ER'01), AAI Books, Springer-Verlag (2001) 38–61
27. Manderick, B., Weger, M., Spiessens, P.: The Genetic Algorithm and the Structure of the Fitness Landscape. In Belew, R., Booker, B. (eds): Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann (1991) 143–150