

# Effects of Selection Schemes in Genetic Programming for Time Series Prediction

Jung-Jib Kim

Artificial Intelligence Lab (SCAI)  
Dept. of Computer Engineering  
Seoul National University  
Seoul 151-742, Korea  
<http://scai.snu.ac.kr/~jjkim>

Byoung-Tak Zhang

Artificial Intelligence Lab (SCAI)  
Dept. of Computer Engineering  
Seoul National University  
Seoul 151-742, Korea  
<http://scai.snu.ac.kr/~btzhang>

**Abstract-** The problem of time series prediction provides a practical benchmark for testing the performance of evolutionary algorithms. In this paper, we compare various selection methods for genetic programming, an evolutionary computation with variable-size tree representations, with application to time series data. Selection is an important operator that controls the dynamics of evolutionary computation. A number of selection operators have been so far proposed and tested in evolutionary algorithms with fixed-size chromosomes. However, the effect of selection schemes remains relatively unexplored in evolutionary algorithms with variable-size representations. We analyze the evolutionary dynamics of genetic programming by means of the selection to response and the selection differential proposed in the breeder genetic algorithm (BGA). The empirical analysis using the laser time-series data suggests that hard selection is more preferable than soft selection. This seems due to the lack of heritability in genetic programming.

## 1 Introduction

Selection plays a major role in evolutionary algorithms since it determines the direction of search whereas other genetic operators propose new search points in an undirected way. The term selective pressure is usually used to denote the strength of selection to influence the directedness of the search [2]. Hard selection emphasizes exploitation of information gained so far and leads to high convergence speed. Soft selection, on the other hand, emphasizes genotypic diversity and explorative search behavior, leading to good convergence reliability. Characterization of the distinctive features of various selection methods is a prerequisite for effective application of evolutionary algorithms and for the design and analysis of new algorithms.

Historically, different evolutionary computation communities have developed different selection schemes. The evolution strategy community has designed various ranking selection schemes. Tournament selection has been developed in evolutionary programming. Genetic algorithms mainly used proportional selection. Goldberg and Deb [5] provide an anal-

ysis on the convergence time and growth ratio of several selection schemes [9]. Bäck and Hoffmeister [3] studied the connection between takeover time and genotypic diversity.

Mühlenbein and Schlierkamp-Voosen [7] introduce the population genetics terms such as selection differential, response to selection, and selection intensity into the evolutionary computation community. Based on these average fitness concepts they presented predictive models that allow the prediction of behavior of evolutionary algorithms. Blickle and Thiele [4] defined selection variance as the expected variance of the fitness distribution of the population after selection and made a mathematical analysis of tournament selection to predict the fitness values. However, the effects of selection schemes remains relatively unexplored in evolutionary algorithms with variable-length representation such as genetic programming.

The aim of this paper is to shed light on the performance characteristics of various selection methods for genetic programming applied to time series prediction. We compare the solution quality of major selection schemes with varying selective pressures. The repertoire of selection methods compared here includes proportional selection, ranking selection, and tournament selection.

The paper is organized as follows. Section 2 defines the time series problem we address. Section 3 describes the genetic programming approach to this problem. Experimental results are provided in Section 4. Section 5 analyzes the evolutionary dynamics of selection methods by measuring the selection differential and the response to selection. Section 6 draws conclusions.

## 2 Time Series Prediction

In time series prediction, we are given a sequence of past values of a random variable and want to forecast the future values of the variable. The past values are summarized as a vector form

$$X(t) = (x(t), \dots, x(t - \phi)). \quad (1)$$

The future values  $x(t + \tau)$  are estimated as a function of the previous values, i.e.,  $f(X(t))$ . In this paper we consider the

short-term prediction problem of  $\phi = 2$  and  $\tau = 1$ , i.e., we predict the value  $x(t + 1)$  at time  $t + 1$  from the input

$$X(t) = (x(t), x(t - 1), x(t - 2)). \quad (2)$$

The predictive accuracy of models  $f_k$  are evaluated by estimating the normalized mean squared error (NMSE) as follows:

$$E_k = \frac{1}{N \times Var} \times \sum_{t=1}^N |x(t + 1) - f_k(X(t))|^2, \quad (3)$$

$$Var = \frac{1}{N} \times \sum_{t=1}^N \left| x(t + 1) - \frac{1}{N} \times \sum_{t=1}^N x(t + 1) \right|^2. \quad (4)$$

The problem of time series prediction is then reduced to finding the predictor  $f_k(X(t))$  that minimizes its NMSE value  $E_k$ .

Experiments have been performed on the prediction of the intensity of an NH3 far infra-red (FIR) laser [10]. Following Zhang et al. [12], only even data points are used to evolve the predictors. Figures 1 and 2 show the training data and test data used in the experiments.

### 3 Genetic Programming for Time Series Prediction

We use genetic programming to solve the time series prediction problem. Genetic programs are used here as models for the time series data. A set of genetic programs  $A_i$  constitutes the population  $\mathcal{A}$ :

$$A = \{A_1, A_2, \dots, A_M\}. \quad (5)$$

A genetic program consists of primitive functions and terminal symbols. We use  $\mathcal{F} = \{+, -, *, \%, \}$  as the function set and  $\mathcal{T} = \{x(t), x(t - 1), x(t - 2), \mathfrak{R}\}$  as the terminal set.  $\mathfrak{R}$  is an ephemeral random number with a range  $[0, 1]$ . The symbol  $\%$  denotes protected division which returns 1 when divided by zero. Figure 3 shows an example of the genetic program. The function denoted by this program is expressed as

$$f(t) = ((x(t - 2) + 0.48) \times x(t - 2) \times 0.48 \times x(t - 1) \% (x(t - 2) + x(t)) \times x(t)). \quad (6)$$

The basic procedure for genetic programming is summarized in Figure 4. The initial population is created randomly. Genetic operators are used to modify the existing individuals and better ones are selected into the next generation.

For fitness evaluation, the genetic program  $A_k$  is used to compute  $f_k(X(t))$  for the training data

$$D = \{(X(t), x(t + 1))\}_{t=1}^N, \quad (7)$$

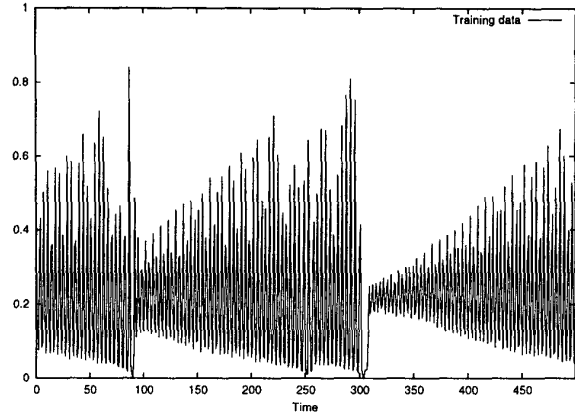


Figure 1: Training data for the laser problem.

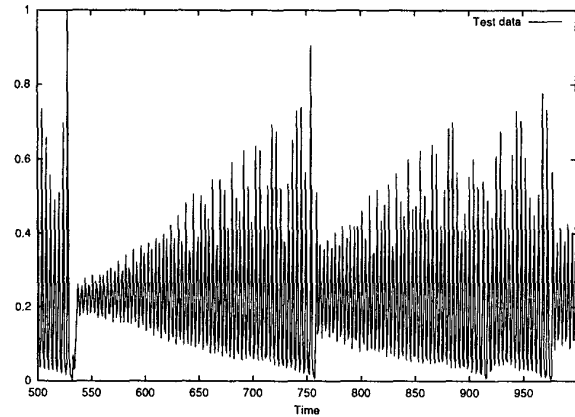


Figure 2: Test data for the laser problem.

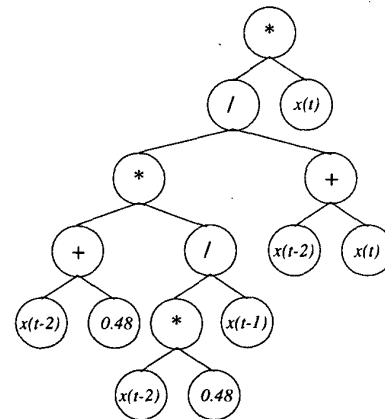


Figure 3: A genetic program for time series prediction.

1. Generate an initial population  $\mathcal{A}(0)$  of random trees using the set of functions  $\mathcal{F}$  and terminals  $\mathcal{T}$ . Set  $g \leftarrow 1$ .
2. Evaluate the fitness of each program in  $\mathcal{A}(g)$  by running it on a set  $D$  of fitness cases.
3. Apply selection, crossover, and mutation to the population  $\mathcal{A}(g)$  to form a new population  $\mathcal{A}(g+1)$ .
4. Terminate if an acceptable solution found. Otherwise, set  $g \leftarrow g+1$  and go to step 2.

Figure 4: Outline of genetic programming.

and the NMSE error  $E_k$  is measured by equation 3. The fitness of an individual  $A_k$  is evaluated as

$$F_k = \frac{1}{1 + E_k + (K \cdot S_k)}, \quad k = 1, 2, \dots, M \quad (8)$$

where  $S_k$  is the tree size measured in terms of the number of nodes, and  $K$  is a small constant for penalizing tree complexity.

After the fitness values of chromosomes are evaluated, fitter individuals are selected into the next generation. The selection methods will be described in the next section.

After the selection process, the genetic trees are modified by genetic operators, i.e., mutation and crossover. Mutation changes the values of tree nodes according to mutation probability. Crossover exchanges subtrees of two parents. Selection of parents depends on the selection method. The process terminates when the maximum number of generations allowed is reached or other termination condition is met.

For all selection methods, we employ the elitist strategy; the best individual is copied into the next generation. We also employ the incremental data inheritance (IDI) technique [11]. IDI accelerates the fitness evaluation process of every selection method, thus does not influence the relative quality of selection methods.

## 4 Experimental Results

We compared the effect of three major selection schemes in GP: proportional, tournament, and ranking selection. In proportional selection, the individuals are selected according to their relative fitness values. The selection probability of  $i$ th individual at generation  $g$  is defined as

$$p_s(A_i^g) = \frac{f(A_i^g)}{\sum_{j=1}^{\lambda} f(A_j^g)}. \quad (9)$$

This is a probabilistic selection method in which every individual with non-zero fitness will have a chance to be reproduced. This selection scheme is adopted by the simple genetic

Parameter	Value
Population size $M$	200
Max generation $G_{max}$	500
Number of runs $R$	20
Probability of crossover $P_c$	0.9
Probability of mutation $P_m$	0.1
Complexity penalty factor $K$	0.00001
Max tree depth $L$	20
Terminals	$x(t), x(t-1), x(t-2), \Re$
Functions	$+, -, *, \%$

Table 1: Parameter values used for experiments.

		Training	Test
PS		$0.197 \pm 0.045$	$0.305 \pm 0.084$
	EXP	$0.182 \pm 0.039$	$0.261 \pm 0.053$
TS	$q = 2$	$0.243 \pm 0.083$	$0.330 \pm 0.087$
	$q = 5$	$0.167 \pm 0.048$	$0.245 \pm 0.064$
	$q = 100$	$0.167 \pm 0.063$	$0.248 \pm 0.074$
RS	$T = 0.50$	$0.280 \pm 0.069$	$0.366 \pm 0.076$
	$T = 0.30$	$0.234 \pm 0.114$	$0.311 \pm 0.105$
	$T = 0.05$	$0.162 \pm 0.065$	$0.224 \pm 0.062$

Table 2: Comparison of NMSE for three selection methods: proportional selection (PS), tournament selection (TS) and ranking selection (RS).

algorithm and believed to be the most similar mechanism that occurs in nature.

Tournament selection [4] is performed by choosing parents randomly and reproducing the best individual from this group. When the number of parents is  $q$ , this is called the  $q$ -tournament selection. This process is repeated  $\lambda$  times to produce the next generation of individuals. The selection probabilities for  $q$ -tournament selection are given by

$$p_s(A_i^g) = \frac{1}{\lambda^q} ((\lambda - i + 1)^q - (\lambda - i)^q). \quad (10)$$

Ranking selection is a selection method which assigns selection probabilities solely on the basis of the rank  $i$  of individuals, ignoring absolute fitness values. In  $(\mu, \lambda)$  uniform ranking [9], the best  $\mu$  individuals are assigned a selection probability of  $\frac{1}{\mu}$ , while the rest are discarded:

$$p_s(A_i^g) = \begin{cases} \frac{1}{\mu}, & 1 \leq i \leq \mu \\ 0, & \mu < i \leq \lambda. \end{cases} \quad (11)$$

Performances of selection methods were compared in terms of solution quality. Table 2 summarizes the average NMSE values over 20 runs for each selection method.

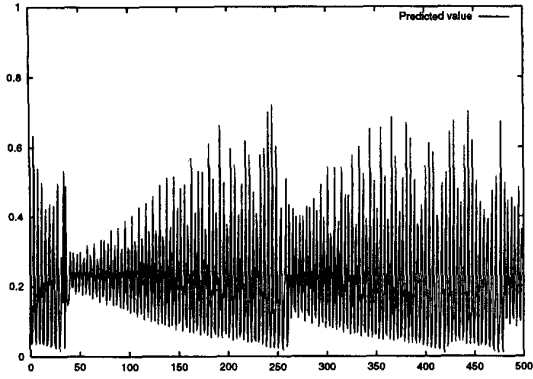


Figure 5: Predicted values.

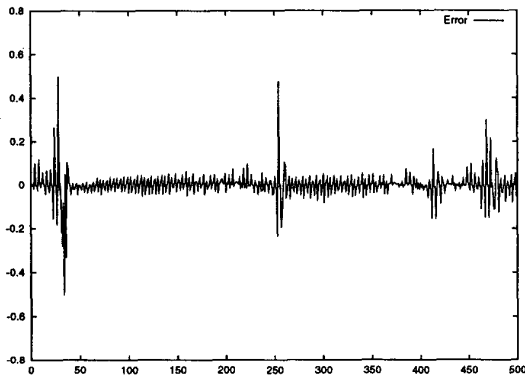


Figure 6: Prediction errors.

Figures 5 and 6 show predicted values and prediction errors for a genetic program whose equation is given as:

$$\begin{aligned}
 f(t) = & (((v0 \% ((((((v1 + (v0 \% ((v2 \% v0) \\
 & \times (v1 \% (v2 - v1)))))) \% (((v1 + (v1 \\
 & + (v0 \% ((v2 \% v0) \times 0.67)))) + v1) \\
 & \times 0.81) + v1)) \% (v1 \% (v1 \times 0.79))) \\
 & + v2) + 0.79) \times v2) + v0) \% (v1 \\
 & \% 0.51)) \times v2 \quad (12)
 \end{aligned}$$

Figures 7-9 show the NMSE errors for the three selection schemes. For each scheme, we varied the algorithm-specific parameters that control the selection pressure. This parameters are fitness scaling, tournament size, and truncation threshold, respectively. The results show that the hardest selection achieved the best performance for every selection method. For example, the tournament size of 100 resulted in better performance than the tournament sizes of 5 and 2 (Figure 8).

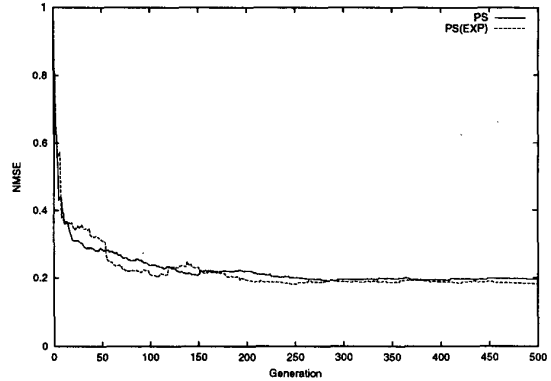


Figure 7: Evolution of NMSE values for proportional selection.

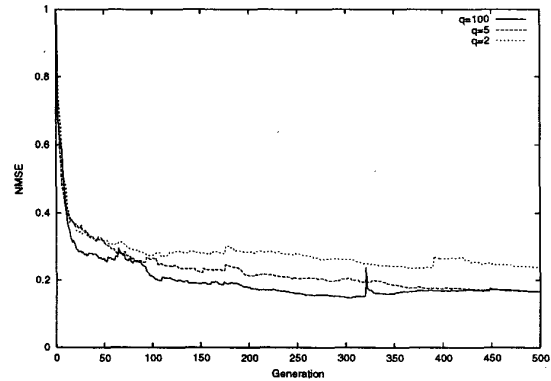


Figure 8: Evolution of NMSE values for tournament selection.

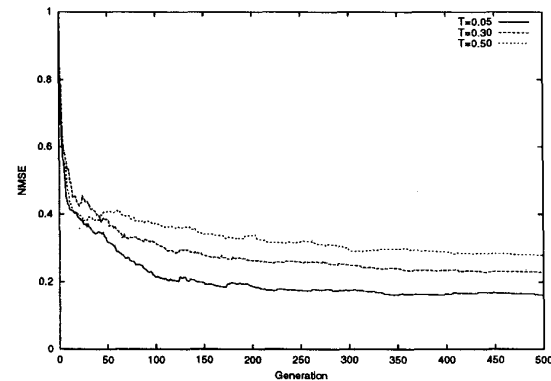


Figure 9: Evolution of NMSE values for ranking selection.

## 5 Analysis

Analysis of selection in evolutionary algorithms can be made at three different levels: the fitness of the best individual [5, 2], the average fitness of the population [7], and the distribution of fitness values in the population [4]. In the following, we analyze the experimental results by comparing them with those of theoretical analysis. This analysis is based on the best fitness. We also examine the behavior of selection schemes in terms of the selection differential and the response to selection which are measures of selective pressure based on average fitness.

### 5.1 Response to Selection

We analyzed the dynamics of population fitness of genetic programming applied to the laser time series data. The concepts of the selection differential and the response to selection has been used as presented by Mühlenbein and Schlierkamp-Voosen [7]. The response to selection is defined as the difference between the population mean fitness of generation  $g + 1$  and the population mean of generation  $g$ .  $R(g)$  measures the expected progress of the population:

$$R(g) = M(g + 1) - M(g) \quad (13)$$

where  $M(g)$  denotes the average fitness of the population at generation  $g$ . Selection differential is defined as the difference between the mean fitness of the selected parents  $M_s(g)$  and the mean fitness of the population:

$$S(g) = M_s(g) - M(g). \quad (14)$$

These measures are useful to make predictions on the behavior of evolutionary algorithms. That is, the designer of the algorithm can try to predict the  $R(g)$  values from the  $S(g)$  values based on the equation:

$$R(g) = b_g S(g), \quad (15)$$

where  $b_g$  is called realized heritability in quantitative genetics. Assuming  $b_g$  is a constant for a certain number of generations, the cumulative response  $R_s$  for  $s$  generations can be predicted by

$$R_s = \sum_{g=1}^s R(g) = b \sum_{g=1}^s S(g). \quad (16)$$

The response to selection is the product of the heritability and the selection differential.

Figures 10-12 depict the average values of  $S(g)$  over 20 runs for each selection method. Comparison of these results with the NMSE results shown in Figures 7-9 confirms that hard selection seems in general better than soft selection in

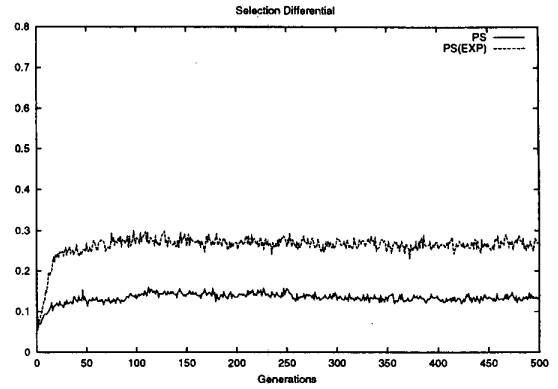


Figure 10: Selection differential curves for proportional selection.

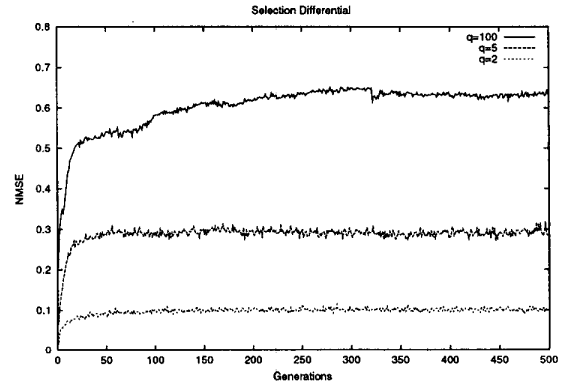


Figure 11: Selection differential curves for tournament selection.

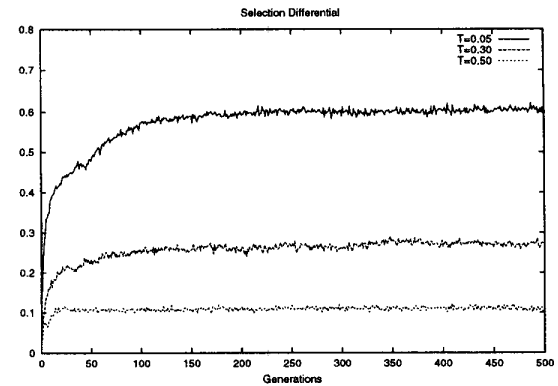


Figure 12: Selection differential curves for ranking selection.

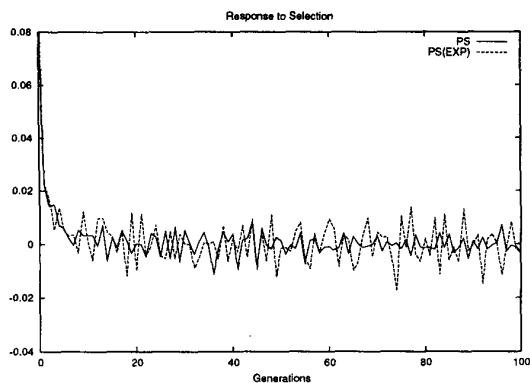


Figure 13: Response to selection curves for proportional selection.

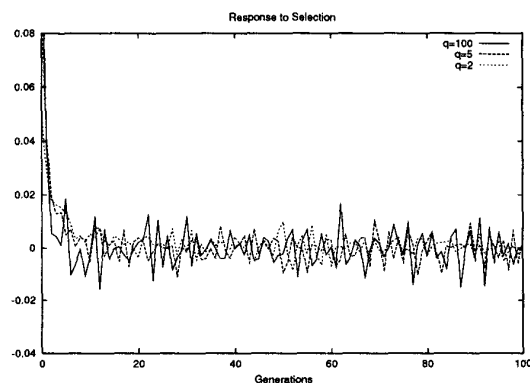


Figure 14: Response to selection curves for tournament selection.

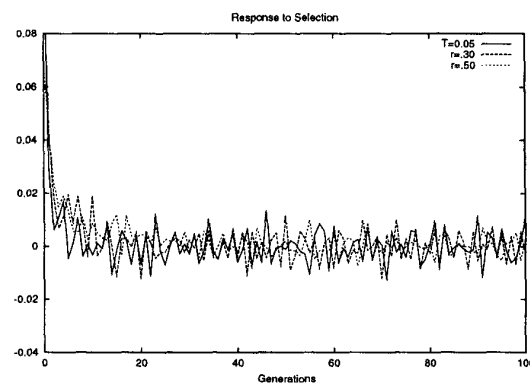


Figure 15: Response to selection curves for ranking selection.

genetic programming. The reason can be seen by observing the low  $R(g)$  values (Figures 13-15). Since the response to selection is the product of the heritability and the selection differential, this result means that genetic programming, at least in its current implementation, has low heritability.

## 6 Conclusions

We have examined three major selection schemes for genetic programming in the context of a time series prediction problem. We compared the effects of selective pressure on solution quality, while the other operators remain the same.

To see the characteristics of the selection methods we analyzed the dynamics of genetic programming with different selection methods by measuring the selection differential and the response to selection. It turned out that the best GP runs used hard selective pressure for each method. The reason is that the current GP has low heritability, as indicated by the low response-to-selection values. A large selection differential combined with a small heritability is an important requirement for effective GP search.

## Acknowledgment

This research was supported in part by the Korea Science and Engineering Foundation (KOSEF) under Grant 981-0920-350-2.

## Bibliography

- [1] Angline, P. J., Evolving predictors for chaotic time series, *Applications and Science of Computational Intelligence: Proceedings of SPIE*, Volume 3390, S. Rogets, D. Fogel, J. Bezdek, and B. Bosacci (Eds.), SPIE, Bellingham, WA, pp. 170-180, 1998.
- [2] Bäck, T., *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, 1996.
- [3] Bäck, T. and Hoffmeister, F., Extended selection mechanisms in genetic algorithms, In Belew, R.K. and Booker, L.B. (eds.), *Proc. 4th Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, pp. 92-99, 1991.
- [4] Bickel, T. and Thiele, L., A mathematical analysis of tournament selection, In *Proc. Sixth Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, pp. 9-16, 1995.
- [5] Goldberg, D. E. and Deb, D., A comparative analysis of selection schemes used in genetic algorithms, In Rawlins, G.J.E. (ed.) *Foundations of Genetic Algorithms*, pp. 69-93, 1991.

- [6] Kim, J.-J. and Zhang, B.-T., Comparison of selection schemes for machine layout design, In *Proc. Asia-Pacific Conf. on Simulated Evolution and Learning (SEAL-98)*, Canberra, Australia, 1998.
- [7] Mühlenbein, H. and Schlierkamp-Voosen, D., Predictive models for the breeder genetic algorithm: Continuous parameter optimization, *Evolutionary Computation*, 1(1):25-49, 1993.
- [8] Mulloy, B.S., Riolo, R.L. and Savit, R.S., Dynamics of genetic programming and chaotic time series prediction, In *Genetic Programming 1996*, pp. 166-174, 1996.
- [9] Schwefel, H.-P., *Evolution and Optimum Seeking*, Wiley, New York, 1995.
- [10] Weigend, A. and Gershenfeld, N., *Time Series Prediction: Forecasting the Future and Understanding the Past*, New York: Addison-Wesley, 1994
- [11] Zhang, B.-T. and Jong, J.-G., Genetic programming with incremental data inheritance, *Genetic and Evolutionary Computation Conference (GECCO-99)*, Orlando, Florida, Morgan Kaufmann, 1999 (in press).
- [12] Zhang, B.-T., Ohm, P., and Mühlenbein, H., Evolutionary induction of sparse neural trees, *Evolutionary Computation*, 5(2):213-236, 1997.