Using Factorial Experiments to Evaluate the Effect of Genetic Programming Parameters

Robert Feldt and Peter Nordin

Dept. of Computer Engineering & Dept. of Physical Resource theory
Chalmers University of Technology
Gothenburg, Sweden
feldt@ce.chalmers.se, nordin@phy.chalmers.se

Abstract. Statistical techniques for designing and analysing experiments are used to evaluate the individual and combined effects of genetic programming parameters. Three binary classification problems are investigated in a total of seven experiments consisting of 1108 runs of a machine code genetic programming system. The parameters having the largest effect in these experiments are the population size and the number of generations. A large number of parameters have negligible effects. The experiments indicate that the investigated genetic programming system is robust to parameter variations, with the exception of a few important parameters.

1 Introduction

The Genetic Programming (GP) method might be the first instance of real *automatic programming* (Koza et al 1999). In an even more general sense, GP could be the first technique to tell the computer *what* to do without having to specify *how* to do it. However, in order for that to be true the user must be able to run the GP system using only a minimal set of natural parameters. In an ideal case there should be no parameters or only parameters that make immediate sense to the user's requirements such as *maximal search time* etc. This is far from true with present genetic programming systems. A modern GP system with additions such as Automatically Defined Functions (ADFs), Demes, and Dynamic Subset Selection have a very large number of parameters and settings creating a combinatorial explosion for the complete parameter space. This enormous parameter search space makes the search for an optimal or near optimal parameter setting difficult for the user.

What is even more severe is the theoretical implication of numerous parameters and settings. Each time we set a parameter we supply information to the search algorithm. If we set too many specific parameters, we might *"point out"* the solution with the parameters and we will not get *more out of the system than we put in*. We are supplying more information than the system is giving us back or in other words we are spending more effort and intelligence on the search for the right combination of parameters than the system does for the right solution.

The standard defence against this argument is that GP is very robust and accepts a wide range setting with little degradation in performance. This is usually only a hunch from GP researchers since there has been no large, systematic investigation of parameter effect using genetic programming. Such an investigation would have the additional benefit of enhancing experiments by providing close to optimal parameter settings. The only broad directions in the

literature are experience-based, *rule-of-thumb-type* parameter recommendations (Koza 1992), (Banzhaf et al 1998).

In this works we describe the first series of experiments that address parameter influence in a broad and systematic way.

The questions that we are addressing are:

- Is GP robust toward different parameter settings or do settings have an effect on performance?
- If there is an effect on fitness, which parameters have the largest effect?
- Is the parameter effect dependent on single parameter settings or are combinations of parameters important?
- Can some parameters be ignored and can general guidelines be devised for the most important ones?

This paper address these questions using statistically sound experimental methods for parameter screening based on *fractional factorial designs* (Box et al 1978). Thes methods reduce the number of runs needed and increases the amount of knowledge that can be gained.

2 Method

To overcome the combinatorial explosion in the number of parameter combinations that need to be considered we use experimental design methods studied in mathematical statistics.

2.1 Experimental design

Statistical Design of Experiments (DoE) provides a framework to design and analyze comparative experiments, ie. experiments with the purpose to determine the quantitative effects of inputs on some output (Kleijnen 1998) (Box et al 1978). In this context the inputs are called *factors* and the output is called the *response*. The major advantage of using DoE designs is that experimentation becomes more efficient: both the effects of individual factors and their interaction can be investigated with limited experimental effort. This is achieved by changing more than one factor at a time.

A basic DoE experimental design is the factorial design where each factor has a discrete number of levels. An example of a two-level factor in GP is whether a certain function should be included in the function set or not. Continuos factors, such as for example the population size, can be used in factorial experiments if two discrete levels are chosen from their valid range. In a full two-level factorial design all combination of factor levels are included, resulting in 2^k different parameter settings, where k is the number of factors. Even for relatively small k:s the number of combinations needed is impractical. To overcome this fractional factorials are used. They utilize the fact that higher-order interactions between factors, ie that two or more factors have a combined effect different from each one of them in isolation, often have negligible effects. By letting lower-order effects, such as the main effects of the parameters and their two-factor interactions, be confounded with each other only a fraction of the full factorial design needs to be run.

The amount of confounding between effects in a design is determined by the design resolution. Design resolution refers to the amount of detail, separate identification of factor effects and interactions, that a design supports. For example, in a design of resolution five the main

effects are confounded with four-factor interactions while two-factor interactions are confounded with three-factor interactions. The confounding pattern can be calculated from the design generators that define how the design is to be constructed. For more information on factorial designs see (Box et al 1978).

A typical strategy for experimentation using DoE is to make sequential use of designs with increasing resolution (Box et al 1978). In the first experiment a large number of factors are included since we do not yet know which of them may have large effects on the response. A heavily fractionalized design with low resolution is often used to *screen out* a majority of the factors. The remaining factors are studied in more detail in later experiments. Later experiments typically have higher resolution to permit separation of main and two-factor effects.

Tradtional DoE have been developed for physical and medical sciences and its development has been biased by the typical applications in these fields. For example, when an experiment is conducted in the real world it is often impractical to control more than 15 factors. (Kleijnen 1998) points out that a number of things are different when conducting experiments on a computer simulation: there are often more factors to be studied, we can practically control many more factors, and we do not need to randomize the run order of the experiments to get results that are robust to uncontrolled, and possibly even unknown, factors. These issues apply, in a similar way to genetic programming experiments.

3 Experiments

A total of 1108 GP runs were performed in seven different experiments on three different problems. In these runs, a total of about 2.5 billion individuals have been evaluated. Below we describe the problems, GP system, factors and response variable used in the experiments. We also describe the design of the experiments.

3.1 Problems

We believe in the importance of evaluating machine learning algorithms over *several* problems. In this work we have used three different binary classification problems. However, we plan to expand the number and types of evaluated problems significantly in future work, see section 6. The problems used are all standard machine learning problems: Ionosphere, Gaussian, and Pima Indians Diabetes Database.

Ionosphere Problem

This real-world radar echo classification problem has been donated by Vincent Sigillito of the Space Physics Group at John Hopkins University in the US. It is taken from the UCI Machine Learning repository (UCI ML Repository 1999). There are 200 instances in the training set and 151 instances in the validation set. The problem has thirty-four attributes and a binary-valued response indicating whether the echoes have detected any structure in the ionosphere.

Gaussian Problem

The gaussian classification problem is an artificial problem for heavily overlapping distributions with non-linear separability. The class 0 is represented by a multivariate normal distribution with zero mean and standard deviation equal to 1 in all dimensions, and the class 1 by a normal distribution with zero mean and standard deviation equal to 2 in all dimensions.

There are 1000 patterns, 500 in each class. We have used a variant of the standard eight-dimensional version, where there are 16 additional false (random) inputs in addition to the eight true inputs. Theoretical maximal classification for the pure 8-D problem is 91%. The problem is probably *not* easier with the false inputs added.

Pima Indians Diabetes Problem

This real-world medical classification problem has been donated by National Institute of Diabetes, Digestive and Kidney Diseases in the US. It is taken from the UCI Machine Learning repository (UCI ML repository 1999). The diagnostic, binary-valued response variable indicates whether the patient shows signs of diabetes according to World Health Organisation criteria (i.e., if the 2 hour post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care). The population lives near Phoenix, Arizona, USA. There are 576 instances in the training set and 192 in the validation set. The problem has eight attributes and a binary-valued response value.

3.2 Genetic programming system, its parameters and their values

For our experiments we used the Discipulus™ system, a commercial implementation of machine code GP (RML 1999). Discipulus™ is based on the AIM-GP approach, a very efficient method for genetic programming formerly knows as CGPS (Nordin 1997). The system uses a linear representation of individuals and a substring-exchanging crossover. In this survey we have used most of the parameters in Discipulus™. These parameters are used as factors in the experiments described below. Their factor identifier (A to Q) and their value at the low and high level used for the experiments are given in table 1. The levels of the continous parameters were chosen to represent qualitatively distinct levels based on our previous experience with the GP system in use. The parameters are briefly described below:

A. PopSize: The number of individuals in the population. At the low level the population size is 50 and at the high level it is 2000.
B. Generations: The system uses steady-state tournament selection so the generation parameter is the number of *generation equivalents* computed from *number of tournaments*. At the low level 50 generations are used and at the high level 250 are used.
C. MutationsFreq: Mutation frequency is the probability that an offspring will be subject to mutation. At the low level the mutation frequency is 10% and at the high level it is 90%.
D. CrossoverFreq: Crossover frequency is the probability that an offspring will be subject to crossover. At the low level the crossover frequency is 10% and at the high level it is 90%.
E. Demes: Determines whether the population is subdivided into subpopulations. In each experiment with demes we used 5 subpopulations, a crossover rate between demes of 3% and a migration rate of 3%. At the low level demes are not used and at the high level they are used.
F. ErrorMeasurement: The error measurement determines whether fitness is the sum of *absolute values* of errors (parameter at low level) *or* the sum of *squared* errors (parameter at high level).
G. DynamicSubsetSelection: Dynamic Subset Selection (DSS) is a method that only uses a subset of all the fitness cases in each evaluation. The selection of fitness cases was based on their individual difficulty (40%), the time since they were last used in fitness calculation (40%) and randomly (20%) (Gathercole 1994). At the low level DSS is not used and at the high level it is used.

H. MissClassificationPenalty: The classification problems are mapped to symbolic regression problems; each class is given a unique number. The fitness value is either the absolute distance or the squared distance between the actual and desired value. This parameter governs the amount of extra penalty that is added to the fitness for incorrect (miss) classifications. At the low level it is 0.0 and at the high level it is 0.25.
I. FunDiv: Determines whether division instructions are in (high level) or not in (low level) the function set.
J. FunCondit: Determines whether conditional instructions such as comparison, conditional loads and jumps are in (high level) or not in (low level) the function set.
K. FunTrig: Determines whether trigonometric functions are in (high level) or not in (low level) the function set.
L. FunMisc: Determines whether other non- trigonometric, non-arithmetic and non-conditional functions are in (high level) or not in (low level) the function set.
M. InitSize: The maximal initial size of the individuals, measured in number of instructions. At the low level it is 50 and at the high level it is 100.
N. MaxSize: The maximal allowed size of an individual, in number of instructions. At the low level it is 128 and at the high level it is 1024.
O. Constants: Determines the number of constants used in each individual. At the low level it is 1 and at the high level it is 10.
P. MutationDistr: When an instruction block is mutated it can be done on several different levels; the block level, instruction level or sub-instruction level (RML 1998). At the low level the distribution between them is 80%, 10%, and 10% respectively, and at the high level it is 10%, 10% and 80%.
Q. HomologousCrossover: Determines the percentage of crossovers that are performed as homologous crossover (Nordin et al 1999). At the low level it is 5% and at the high level it is 95%.

3.3 Response variable

We have used the maximum validation hit rate as the response variable for all problems and runs. This value was obtained by extracting the best individual on *training data* and running it on the validation set. It is reported as the percentage of correctly classified instances in the validation set.

Our choice of response variable defines the unit for the effects from the analysis of the experimental data. If, for example, an effect is calculated to be 5 this means that the average effect that can be expected when changing the factor from its low to its high level will be 5 percentage units (*not* 5%). Thus if the average response is 65% we would expect 70% on average with the factor at its high level.

3.4 Experimental designs

We have used three different experimental designs in a sequential fashion, each one based on the results from the previous one. The first two designs have been used on all three problems with the settings of factor levels described above. The third design uses different levels for the factors and has only been used on the gaussian problem. The purpose of the first ex-

periment is to screen the large number of factors down to a more manageable set. Later experiments study the effects of the remaining factors in more detail.

To reduce the number of runs in the screening experiment we have employed a saturated design first described by Ehlich (Ehlich 1964) (Statlib 1999). This design allows the estimation of the main effects of seventeen factors in eighteen runs. The confounding patterns for this design is very complicated; main effects are confounded with several two- and higher-order effects.

The factors that had the largest effect in the screening experiments are varied in the second round of experiments. The rest of the factors are held constant at intermediate levels (N = 256, P = (40, 40, 20), Q = 50) or at the level indicated by the sign of its effect from the screening experiment (I, K, L, M, O at their low level and J at its high). We employ a fractional factorial experiment of resolution four. In this design the main effects are confounded with three-factor interactions which are assumed to be negligible. This allows the estimation of all main effects. Two-factor effects can be estimated but are confounded with each other. The actual design used is a 2^{8-4} fractional factorial with 16 runs (Box et al 1978). The generators for this design are D=ABC, E=BCH, F=ACH and G=ABH where a low level is represented by ñ1 and a high level by 1.

In order to estimate all two-factor interactions individually we need a design of resolution five. This is illustrated for the gaussian problem in the third experiment, which uses a 2^{5-1} fractional factorial with 16 runs (Box et al 1978). The generator for this design is H = ABCD.

In this third experiment we study the five factors that had the largest effect in experiment 2 on the gaussian problem. We alter the levels of these factors to gain more knowledge of their effect. The population size and number of generations had a significant effect and by increasing them (A to (500, 5000) for low and high level respectively and B to (100, 500)) we want to investigate if this effect holds also for higher levels. By increasing the low level of the mutation and crossover probabilities to 50% and keeping the high level at 95%, we can investigate if the level of 95% was extreme. By altering the values of both the low (to 0.05) and high levels (to 0.5) of the miss-classification penalty we can investigate if it is only important to have this penalty regardless of level or if the level in itself is important.

4 Results

Below we document the results for the seven experiments conducted. All values reported for the effect of factors and for confidence intervals is in the same unit as the response variable, see section 3.3. We have conducted a sensitivity analysis to evaluate how sensitive our results are to the number of replicates used for each parameters setting. This analysis is briefly described below.

4.1 Results of the screening experiment on IONOSPHERE

For each of the eighteen factor settings ten (10) replicates were run on the ionosphere problem. The standard error calculated from these 180 runs was 2.04 giving a 95% confidence interval of 4.63. The effects that were statistically significant at this confidence level are (in order of decreasing effect): A, B, G, C, H, D, E, F. The effect of A was about 45% larger than the effect of F.

4.2 Results of the screening experiment on Gaussian

For each of the eighteen factor settings eight (8) replicates were run on the gaussian problem. The standard error calculated from these 144 runs was 0.74 giving a 95% confidence interval of 1.94. The effects that were statistically significant at this confidence level are (in order of decreasing effect): A, B, C, H, E, D, G, F, J*, O*, P*, L*. However, note that the four factors marked with an asterisk had much smaller effect than the previous eight. For example the effect of F is more than four times higher than the effect of J.

4.3 Results of the screening experiment on PIMA-diabetes

For each of the eighteen factor settings, eight (8) replicates were run on the pima-diabetes problem. The standard error calculated from these 144 runs was 0.33 giving a 95% confidence interval of 0.96. The effects that were statistically significant at this confidence level are (in order of decreasing effect): A, C, G, B, F, E, H, D, L*, N*, P*, M*. However, note that the four factors marked with an asterisk had much smaller effect than the previous eight. For example the effect of D is more than eight times the effect of L.

4.4 Result of Second experiment on ionosphere

For each of the sixteen factor settings ten (10) replicates were run on the ionosphere problem. The standard error calculated from these 160 runs was 0.66 giving a 95% confidence interval of 1.50. The effects that are statistically significant at this confidence level are shown in table 4.

Table 4: Factors and their levels for experiment 2 on the ionosphere problem

CONTRAST	EFFECT	95% CONF. INTERVALL
A	4.19	+/- 1.50
B	2.23	+/- 1.50
AD + BC + EH+ FG	2.16	+/- 1.50
AG + BH + CE + DF	1.89	+/- 1.50
AH + BG + CF + DE	1.75	+/- 1.50

The population size (A) has the largest effect while the number of generations (B) and three different two-factor-interaction combinations have similar effects. The values reported in the table should be interpreted in the following way: if we change the level of factor A from its low to its high level we can expect an average increase in the validation hit rate by 4.19 units with a 95% confidence interval from 2.69 to 5.69 units. The same type of interpretation can be made for all effects reported in this paper.

The average validation hit rate was 92.1%, with a maximum of 98.7% and a minimum of 66.9%. The maximum average for a particular setting of the factors was 98.2% and the minimum 85.8%. These results can be compared with the maximum reported result from the UCI

database describing the `ionosphere` problem: an average of 96% obtained by a backprop NN and 96.7% obtained with the IB3 algorithm (UCI ML repository 1999). However, we measure generalisation in a slightly different way: In the GP community it is common to look for the best generalizer in the population at reporting intervals in contrast to noting generalization capabilities among the best performing solution candidate on the training set. This difference applies for all experiments in this paper.

4.5 Result of second experiment on gaussian

For each of the sixteen factor settings ten (10) replicates were run on the gaussian problem. The standard error calculated from these 160 runs was 0.84 giving a 95% confidence interval of 1.94. The effects that are statistically significant at this confidence level are shown in table 5.

Table 5: Factors and their levels for experiment 2 on the gaussian problem

CONTRAST	EFFECT	95% CONF. INTERVAL
A	11.51	+/- 1.94
C	5.21	+/- 1.94
B	5.14	+/- 1.94
AD + BC + EH+ FG	3.39	+/- 1.94
D	2.82	+/- 1.94
AH + BG + CF + DE	2.76	+/- 1.94
AF + BE + CH + DG	2.35	+/- 1.94

The population size (A) clearly has the largest effect with the mutation probability (C) and number of generations (B) having about half the effect of A. Three different two-factor-interaction combinations and the crossover probability (D) have smaller effects.

The average validation hit rate was 63.8%, with a maximum of 88.9% and a minimum of 48.6%. The maximum average for a particular factor setting was 83.7% and the minimum 52.3%. This can be compared to the theoretical limit for this problem with a dimensionality of eight: 91%. However, note that this limit does not take the eight false inputs into account.

4.6 Result of second experiment on pima-diabetes

For each of the sixteen factor settings ten (10) replicates were run on the pima-diabetes problem. The standard error calculated from these 160 runs was 0.72 giving a 95% confidence interval of 1.63. The effects that are statistically significant at this confidence level are shown in table 6.

Table 6: Factors and their levels for experiment 2 on the pima-diabetes problem

CONTRAST	EFFECT	95% CONF. INTERVAL

A	5.72	+/- 1.63
B	2.12	+/- 1.63
G	2.02	+/- 1.63

The population size (A) have the largest effect while the number of generations (B) and the dynamic subset selection (G) have smaller effects.

The average validation hit rate was 65.46%, with a maximum of 77.6% and a minimum of 61.5%. The maximum average for a particular setting of the factors was 72.8% and the minimum 61.5%. These results can be compared with the maximum reported result from the UCI database describing the pima-diabetes problem: 76% using the ADAP learning algorithm (UCI ML repository 1999).

4.7 Result of third experiment on gaussian

For each of the sixteen factor settings ten (10) replicates were run on the gaussian problem. The standard error calculated from these 160 runs was 0.89 giving a 95% confidence interval of 2.02. The effects that are statistically significant at this confidence level are shown in table 7. Note that the levels used for the factors in this experiment are not the same as for the previous experiments. Hence, the actual effects are not comparable between experiments 2a and 3.

Table 7: Factors and their levels for experiment 3

CONTRAST	EFFECT	95% CONF. INTERVAL
B	9.39	+/- 2.02
A	7.53	+/- 2.02
H	3.71	+/- 2.02
AD	-2.30	+/- 2.02

The number of generations (B) and the population size (A) have the largest effects. The positive effect of the increased miss-classification is smaller but still significant. The same is true for the interaction between the population size (A) and the crossover probability (D). Note that since this design has resolution five this two-factor interaction is not confounded with any other two-factor interaction, as was the case in previous experiments. The somewhat surprising negative effect of this interaction means that some caution is called for when using large population sizes; increasing the crossover probability might have a detrimental effect.

The average validation hit rate was 76.4%, with a maximum of 88.9% and a minimum of 61.6%. The maximum average for a particular factor setting was 85.8% and the minimum 65.1%.

5. Discussion

We have presented our first results in a larger project attempting to investigate the effect of GP parameters. Even though these results stem from a limited number of problems and experimental designs we believe that some interesting conclusions can be drawn. However, we are far from settling the questions raised in the introduction, but we can identify interesting patterns.

In all three screening experiments the same eight parameters had the largest effects with the remaining nine factors having small or statistically insignificant[1] effects. Among these nine factors that were consistently screened *out*, we can find the factors determining the function set, the initial and maximal size of the individuals, the number of constants, the distribution of different mutation operators and the amount of crossovers that are homologous. It will be interesting to see if this result is valid for other problems and in other ranges of the continuos parameters.

Consistently, on all three problems, the population size and the number of generations are the most significant parameters. The population size comes out on top in the second experiments on all three problems with the number of generations a close second or third. However, note that the effect of the population size is numerically much larger than the other effects; this indicates that having a large population is important to get good results with GP. *Effort* has not been individually targeted in this survey, but it is interesting to note that choosing a large population size sometimes is more important than a large number of generations. In other words: a large population size running for very small number of generations could be better than a small population size running for a ìnormalì number of generations. More investigation is needed on this.

It is interesting to note that the mutation and crossover probabilities have rather large effects on the gaussian problem. This somewhat contradicts the notion that mutation probability should be low. However, these factors did not have a statistically significant effect on the two real-world problems.

Dynamic subset selection can have a positive effect on the performance (Gathercole 1994). The fact that it, in addition, decreases the execution time of a run considerably would further speak for a more widespread use.

On both the gaussian and the ionosphere problem there are significant two-factor interactions. Since the design for experiment number two had a resolution of four we cannot separate the effect of different two-factor interactions. If we would like to do so we could add further runs to the existing designs or use a design of resolution five. Note that it can often be wise to use a design with lower resolution first and then add runs to separate between two-factor interactions of interest. In general, this will reduce the total number of runs needed. For example, to separate the four two-factor interactions having a combined effect of 3.39 on the gaussian problem in table 5 would require 3 extra experiments. Using a design of resolution five would require 64 runs; 48 more runs than for the design used herein.

The third experiment on the gaussian problem was included to show an example of a design of resolution five. Furthermore, the levels of the factors studied were changed to see their effect in other ranges of values. It is notable that the population size and number of generations are still the dominant factors. Note, however that the population size is no longer dominating; this could indicate that there is a limit to what can be gained from increasing the population size. The positive effect of the miss-classification penalty factor indicates that not only is it good to have such a penalty, but a relatively large penalty is better than a smaller one.

Our results partly support the notion that GP systems are robust to different parameter settings, as long as we choose the right values for the most important ones: population size and number of generations. On some problems the crossover and mutation probability can give good results with large levels. However, the negative interaction between population size and crossover probability in experiment 3 indicates that some caution must be taken.

---

[1] If an effect is not statistically significant it can not be separated from the natural variation in the response, ie noise.

The methodology used in this work can be used to optimize the results from a GP system. For example, note that the average response on the third experiment on gaussian is higher than the average response on the second experiment on the same problem. This is because the levels used in the third experiment were chosen based on the results from the second experiment. Thus, in addition to giving researchers a way to map out the effect of different parameters, DoE techniques may be used to optimize the response on a particular problem.

A drawback with the kind of DoE techniques used in this work is that they assume that higher-order interactions between factors are negligible. The empirical evidence for making this assumption are abundant; experimental investigations frequently show that the effect can be explained by a few important factors (Kleijnen 1998). However, we can never be fully sure and it will probably be wise to conduct full factorial experiments on some problems to validate this assumption. We have also noted that the responses in our experiments are often not normally distributed but grouped into clusters. In theory this makes statistical analysis of effects difficult since it violates the assumption of normally distributed responses. In practice, most statistical techniques have shown to be robust against deviations from normality (Box et al 1978).

It is worth noting that the GP system consistently performed very well compared to the previously reported best results on the test problems but with the caveat that generalization is measured differently. In future work we plan to change generalisation measurements to comply with the methods used in the UCI-database.

7. Conclusions

The Design of Experiments (DoE) techniques, from mathematical statistics, have been introduced as a solid methodology for evaluating the effect of genetic programming parameters. These techniques can also be used to increase the performance of a GP system, by guiding the user in choosing ëgoodí parameter combinations.

Our experiments show that, on three binary classification problems, the most important parameter was the population size followed by the number of generations. On one problem, large mutation and crossover probabilities had a positive effect. Furthermore, on all three problems, the same and large number of factors could be screened out because their effect could not be distinguished from noise. The result supports the notion that GP systems are robust against parameter settings but highlights the fact that there are a few parameters that are crucial.

This work reports the first results from a larger project attempting to investigate the effect of GP parameters. Much more work, involving more detailed designs as well as more varied test problems, is needed before we can address the questions as to the role and effect of GP parameters. We believe that such findings can be of great importance to the applicability of genetic programming in both industry and academia.

Acknowledgements

The authors wish to acknowledge Martin Hiller and Bill Langdon, whose comments increased the quality of this paper. Peter Nordin gratefully acknowledges support from the Swedish Research Council for Engineering Sciences.

References

Banzhaf, W., Nordin, P. Keller, R. E., and Francone, F. D. (1998). *Genetic Programming —An Introduction. On the automatic evolution of computer programs and its applications*. Morgan Kaufmann, Germany

Box, G. E., Hunter, W. G., Hunter, J. S. (1978). *Statistics for Experimenters ñ an Introduction to Design, Data Analysis and Model Building.* Wiley & Sons, New York, USA.

Ehlich, H. (1964). Determinantenabschatzungen fur binare Matrizen. *Math. Z.* 83, 123-132.

Gathercole C. and Ross P. (1994) Dynamic Training Subset Selection for Supervised Learning in Genetic Programming, Chris. In proceedings of the 3rd conference on Parallel Problem Solving from Nature (PPSN III), Springer-Verlag, Berlin, Germany.

Kleijnen, J. P. C. (1998). Experimental Design for Sensitivity Analysis, Optimization, and Validation of Simulation Models. In *Handbook of Simulation*, (ed.) Banks, Wiley & Sons, New York, USA.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA., USA.

Koza J.R, Andre D., Bennett F.H., Keane M. A. (1999) *Genetic Programming III: Darwinian Invention and Problem Solving*. Academic Press/Morgan Kaufmann.

Nordin, J.P. (1997), *Evolutionary Program Induction of Binary Machine Code and its Application*. Krehl Verlag, Muenster, Germany.

Nordin J. P., Banzhaf W., and Francone F. (1999) Efficient Evolution of Machine Code for CISC Architectures using Blocks and Homologous Crossover. To appear in *Advances in Genetic Programming III*, (eds.) Langdon, OíReilly, Angeline, Spector, MIT-Press, USA

RML (1999) Register Machine Learning Incorporated. http://www.aimlearning.com

StatLib (1999), Online Statistical resources library at the Department of Statistics, Carneige Mellon University, USA, http://lib.stat.cmu.edu/.

UCI ML repository (1999). Files for the Pima-diabetes and Ionosphere problems from the Machine Learning repository at University of California, Irvine describing the Ionosphere problem. http:// www.ics.uci.edu/~mlearn.