Mining Water Quality Time Series for Predictive Rules of Algal Blooms by Genetic Algorithms

Jason Bobbin and Friedrich Recknagel
Department of Soil and Water, The University of Adelaide
Glen Osmond, South Australia 5064

Abstract This paper discusses the application of genetic algorithms to the construction of rule-based models. Water quality time series will be explored to extract predictive rules for algal blooms in freshwater lakes. The hypertrophic Japanese lake Kasumigaura is used to demonstrate the technique.

1 Introduction

Pattern recognition in water quality time series by machine learning algorithms such as artificial neural networks and genetic algorithms has been identified as a promising approach to predict future and understand past behaviour of freshwater ecosystems (Recknagel [1997]; Recknagel and Wilson [1999]; Whigham and Recknagel [1999]). Water quality data have been systematically monitored and stored for many freshwater lakes worldwide since the 1980's. Current machine learning techniques are supported by high speed and large memory computers creating new opportunities to explore this data for predictive modeling of freshwater quality.

Harmful blooms of toxic blue-green algae can cause serious management problem in lakes and drinking water reservoirs. Predictive models of algal blooms can provide decision support on preventative and operational control of these events.

In the context of this paper genetic algorithms are used to develop a predictive rule set from water quality time series for the prediction and explanation of algal blooms in the Japanese lake Kasumigaura. Lake Kasumigaura is situated 60 kms north east of Tokyo and is Japan's second largest lake. The lake is shallow, with a maximum depth of 7 meters and a mean depth of 4 meters. Because of the lakes shallowness and strong mixing of the lake water by winds, persistent stratification of the water body does not occur. The lake has been monitored for more than 20 years at different sampling sites providing a long record of water quality data. Data from a central sampling site which is known for recurrent harmful algal blooms have been chosen for modelling in this paper.

Genetic algorithms are robust optimisation procedures which have been widely utilised in areas such as engineering optimisation and design, forecasting, image recognition and functional optimisation. Genetic algorithms are modeled off natural evolution, and require no information about the nature of the search space. Evolutionary techniques are also applied to a wide variety of problem representations, recently including rule sets for knowledge discovery (Bobbin and Yao [1999]).

Genetic algorithms allow the creation of relational models for ecological applications. They can be applied for modeling with a priori hypothesised relational rules or ad hoc constructed rules. In this way a structure which has desirable properties can be chosen for evolution.

By combining function optimisation and rule set generation a hybrid genetic algorithm approach has been applied for modeling of algal bloom events in Lake Kasumigaura. Mining the available water quality data for useful patterns and regularities offers the chance for a deeper understanding of the problem domain and the possibility of useful new relationships being discovered inductively. The evolutionary approach proposed in this paper addresses the problem of elucidation and prediction from an inductive machine learning method in an ecological problem domain.

2 Evolving Models of Algal Abundance

Evolutionary methods in machine learning use a formalised statement of evolution to evolve solutions to a problem of interest. The term evolutionary computation encompasses several different computing paradigms including self-adaptive optimisation approaches (evolutionary strategies, Schwefel [1994], and evolutionary programming, Fogel [1995]). These approaches concentrate on the phenotypic aspects of evolution in contrast with genetic algorithms, Holland [1975], and genetic programming, Koza [1992], which use a more genetic based analogy. This paper utilizes a hybrid algorithm based on the evolutionary strategies and evolutionary programming paradigms to optimize parameters associated with coevolved rule sets.

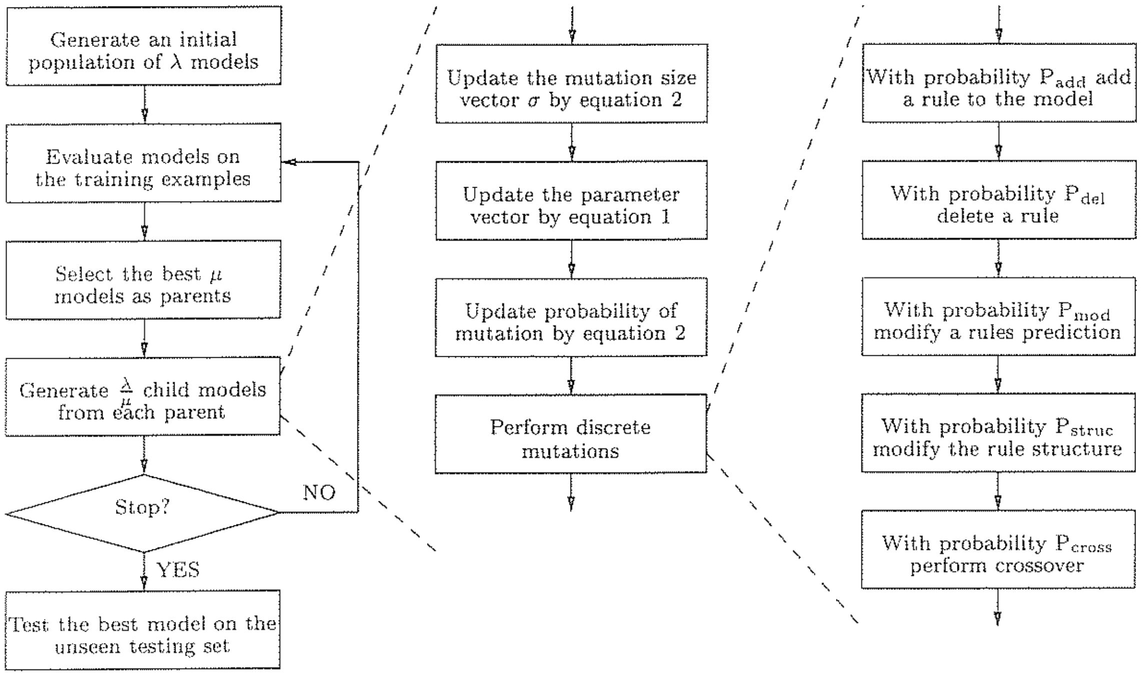
Generate an initial population of λ models
Evaluate models on the training examples
Select the best μ models as parents
Generate $\frac{\lambda}{\mu}$ child models from each parent
Stop?
NO
YES
Test the best model on the unseen testing set
Update the mutation size vector σ by equation 2
Update the parameter vector by equation 1
Update probability of mutation by equation 2
Perform discrete mutations
With probability P_{add} add a rule to the model
With probability P_{del} delete a rule
With probability P_{mod} modify a rules prediction
With probability P_{struc} modify the rule structure
With probability P_{cross} perform crossover

Figure 1: The main components of the evolutionary process for model creation

Finding an optimal model for prediction from a data set is a difficult problem. Most studies concentrate on predictive accuracy of models and the training and preprocessing required to achieve good results. This approach neglects the possibility of acquiring knowledge from the learning algorithm. In order to predict new data the model must have learned some knowledge of the patterns contained in the data. The representation of this knowledge is not generally available for examination. Evolutionary methods can be applied to a range of problem representations, allowing the evolution of models with more transparent knowledge representations. This allows model predictions and model behaviour to be understood. It may also be of use in further understanding the patterns, regularities and relationships which exist and drive a certain phenomenon, such as algal abundance in Kasumigaura.

2.1 Structure of the Evolutionary Process

The structure of the evolutionary algorithm is shown in figure 1. For the experiments presented in this paper the population size was set to 500 (λ=500) and the ratio of parent to children was set to $\frac{1}{5}$ (μ = 100). Each model is constructed from a rule set and a parameter vector. The parameter vector is evolved according to the paradigms of evolutionary strategies, Schwefel [1994], and evolutionary programming, Fogel [1995].

A genetic algorithm is used to automatically find a rule structure associated with the parameters for the prediction of the algal time series data. The approach concentrates on using mutation and selection as the principal search operators. Although present, the crossover operation plays a secondary role in the search.

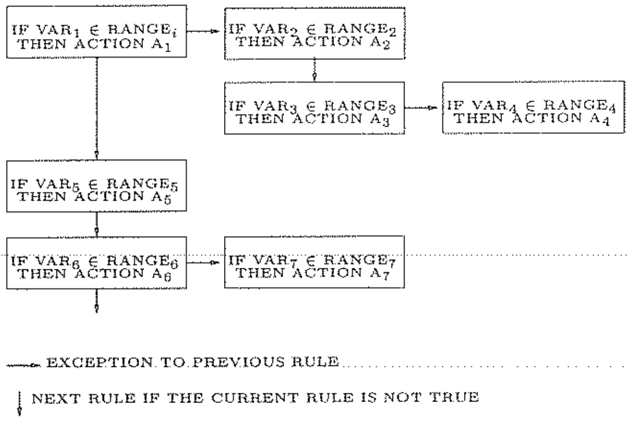
IF VAR$_1$ ∈ RANGE$_i$ THEN ACTION A$_1$
IF VAR$_2$ ∈ RANGE$_2$ THEN ACTION A$_2$
IF VAR$_3$ ∈ RANGE$_3$ THEN ACTION A$_3$
IF VAR$_4$ ∈ RANGE$_4$ THEN ACTION A$_4$
IF VAR$_5$ ∈ RANGE$_5$ THEN ACTION A$_5$
IF VAR$_6$ ∈ RANGE$_6$ THEN ACTION A$_6$
IF VAR$_7$ ∈ RANGE$_7$ THEN ACTION A$_7$
EXCEPTION TO PREVIOUS RULE
NEXT RULE IF THE CURRENT RULE IS NOT TRUE

Figure 2: Structure of an evolved rule set

The selection procedure is a truncation method referred to as (μ,λ) selection (Schwefel [1994]). From a population of λ individuals the best μ are selected as the parents to the next generation. Each parent produces $\frac{\lambda}{\mu}$ children which form the next generation. This technique has no elitism, the best individual may be lost to the population. This method has empirically been found to promote successful self-adaptation.

A rule set is structured as a series of if-then rules.

Each rule can have an exception, allowing a further test(s) and the possibility of having a selected rules prediction modified. The rule structure is shown in figure 2. The value of all the subscripted variables are linked to the numbers in the parameter array. The possible values of VAR$_i$ depend upon the attributes in the input set being used. The A$_i$ variable is arbitrarily limited to 6 possible values, as this was sufficient for the algal prediction problem. Each VAR$_i$ has its range (RANGE$_i$) divided into 3 regions for comparisons. A rule in figure 2 might have VAR$_i$ set to water temperature, RANGE$_i$ set to range 1, and A$_i$ set to prediction 2. This symbolic knowledge is realised by values held in the parameter vector. The parameter vector holds the lower and upper bounds for RANGE$_i$ and the predicted chlorophylla measurement for A$_i$.

The three ranges allowed for each variable is reduced to two parameters by considering the three ranges to be (MIN, p_1), (p_1, p_2) and (p_2, MAX), where MIN and MAX are smaller and larger than any possible value respectively (ie 0 and ∞). The parameter vector $(p_{11}, p_{12}, p_{21}, \ldots, A_1, A_2, \ldots)$ consists of the values

$$\begin{array}{lll} p_{jk} & \text{where} & j \in \{1, \ldots, \text{Inputs}\} \\ & \text{and} & k \in \{1, 2\} \end{array}$$

and the values A_l where $l \in \{1, \ldots, 6\}$.

A rule is a direct comparison between an input variable and a range of values. If the input variable's (VAR$_i$) value falls within the range (RANGE$_i$) coded on the rule then the action (A$_i$) coded on the rule is performed. If a rule is true then its exception list is checked. If it is not then its 'otherwise' rule is checked. In this way the rule set is recursively parsed, with the last true rule having its prediction used as the output of the model on the given input example.

The evolutionary process is self-adaptive. There is a mutation vector σ which determines the size of the mutation step performed on a model, and which is co-evolved with the model. The parameter vector x is updated according to

$$\vec{x}' = \vec{x} \cdot \vec{N}(\vec{0}, \vec{\sigma}) \qquad (1)$$

The strategy, or mutation vector σ is updated according to

$$\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \qquad (2)$$

where $\tau \propto (\sqrt{2\sqrt{n}})^{-1}$ and $\tau \propto (\sqrt{2n})^{-1}$. $N(0,1)$ in equation 2 is a normally distributed random number with standard deviation one and expectation zero. $N_i(0,1)$ means the random variable is sampled anew for each value of i. σ_i denotes the i th component of $\vec{\sigma}$. The constant of proportionality for τ and τ' can be interpreted as a learning rate ([Bäck, 1996,page 72]), and although it is often set to 1, for the current experiments a constant of 0.5 was found most useful. Recombination and covariance methods were not used in the self-adaptation scheme.

The rule set is evolved by a set of mutation operators acting on the discrete rule tree structure. With probabilities determined by a probability vector updated by equation 2 one or more of the mutation operators listed in figure 1 can occur. The addition and deletion mutations add or delete a random rule from the tree. The modification mutation modifies the chlorophylla prediction that a rule codes. The structural mutation changes the order that rules are parsed in the rule tree, and the crossover operator performs a crossover operation between members of the population. Details of operator functions and performance are discussed in Bobbin and Yao [1999].

2.2 Model Evaluation

A root mean square (RMS) error formula is used to evaluate the models produced by the evolutionary process. The fitness associated with the models is calculated from a modified RMS error method to concentrate the evolutionary search on models which correctly predict higher levels of chlorophylla. The fitness function used was

$$f = \begin{cases} \sqrt{\frac{(x_{\text{pred}} - x_{\text{actual}})}{\text{Examples}}} & \text{If } x_{\text{pred}} > x_{\text{actual}} \\ \sqrt{\frac{4 \cdot (x_{\text{pred}} - x_{\text{actual}})}{\text{Examples}}} & \text{If } x_{\text{pred}} < x_{\text{actual}} \end{cases}$$

where Examples is the number of input vectors in the input set. This fitness function biases the fitness in favour of over prediction. All numbers reported in this paper are standard RMS evaluations of the models.

3 EXPERIMENTAL STUDIES

The evolutionary algorithm is used to extract models using a variety of different characteristics about the state of the lake. The different input sets are summarized in table 1.

The evolutionary process made 20 independent runs with each of the input sets in table 1 with a population size of 500 and for 150 generations. The quartile and median values are listed in table 2. Results from input set 1, 2, 4 and 6 can be considered equivalent in the current experimental set up. Secchi depth, an indicator for water transparency, is common to all runs which produce results with a median error of less than 40. This is most probably due to the indication of a clear water stage by high secchi depths. This is indicative of low algal abundance (chlorophylla levels) in winter or after high grazing pressure from an abundant zooplankton population.

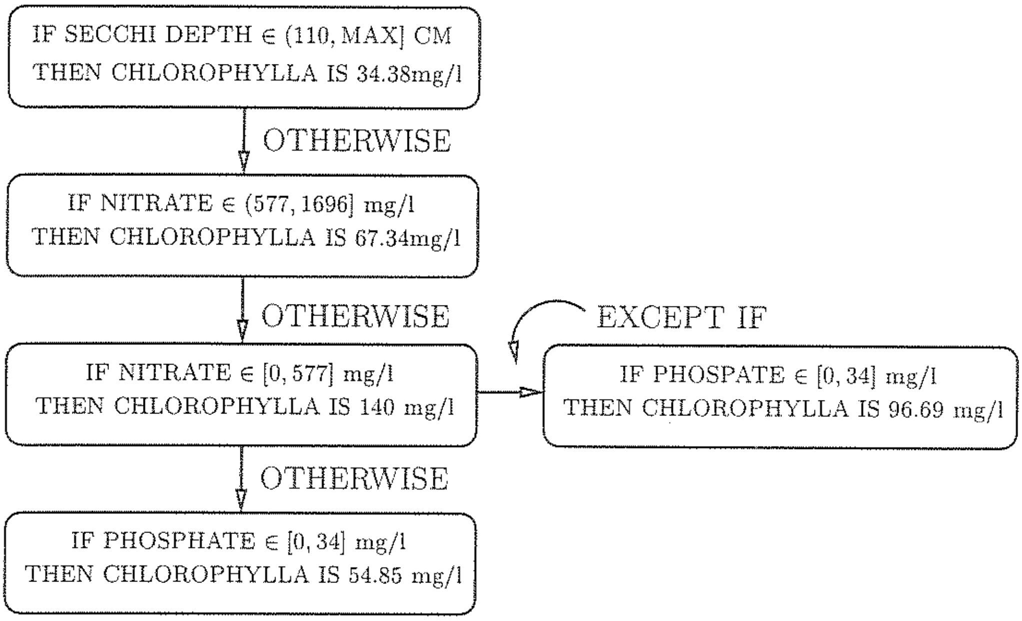
IF SECCHI DEPTH ∈ (110, MAX] CM
THEN CHLOROPHYLLA IS 34.38mg/l
OTHERWISE
IF NITRATE ∈ (577, 1696] mg/l
THEN CHLOROPHYLLA IS 67.34mg/l
OTHERWISE
EXCEPT IF
IF NITRATE ∈ [0, 577] mg/l
THEN CHLOROPHYLLA IS 140 mg/l
IF PHOSPATE ∈ [0, 34] mg/l
THEN CHLOROPHYLLA IS 96.69 mg/l
OTHERWISE
IF PHOSPHATE ∈ [0, 34] mg/l
THEN CHLOROPHYLLA IS 54.85 mg/l

Figure 3: Rule Set

Input Set	Lake Data
1	Water Temperature, Secchi Depth, PO_4, NO_3, NO_3 : PO_4 ratio, Dissolved Oxygen, ph, Solar Radiation
2	Water Temperature, Secchi Depth, PO_4 and NO_3 concentration
3	Water Temperature, Secchi Depth, NO_3 : PO_4 ratio
4	Water Temperature, Secchi Depth, PO_4 and NO_3 concentration, NO_3 : PO_4 ratio
5	Water Temperature, Solar Radiation, PO_4 and NO_3 concentration
6	Water Temperature, Secchi Depth, Solar Radiation, PO_4 and NO_3 concentration

Table 1: Table of input parameters for the lake model

Input Set	1st Quartile	Median	3rd Quartile
1	36.9123	37.8305	40.6746
2	37.1431	37.6297	39.1522
3	42.4198	43.7651	45.5794
4	37.0226	37.9379	38.7717
5	41.1755	42.2246	42.9999
6	37.3972	37.8433	39.4419

Table 2: Summary of results for different input sets

3.1 A model from input set 2

For input set 2 consisting of Temperature, Secchi Depth, Phosphate and Nitrate concentrations a typical run of the genetic algorithm discovered the model in figure 4. The mean square error on the testing set of this model is 37.80, and the training set RMS is 28.08. This model used the rule set shown in figure 3.

The model firstly uses the secchi depth to classify low algal abundance. This rule is mostly true over the low algal winter months. The model uses nitrate and phosphate concentrations to indicate algal consumption of available nutrients. It is interesting that the level of nitrate is used to indicate the most severe blooms. For Kasumigaura, the model discovered that large algal abundance is well correlated with low nitrate levels over the summer months. This may be explained by the fact that blue-green algae, which are dominating in summer, are able to fix nitrogen from the atmosphere for photosynthesis and do not depend on dissolved nitrogen in the water. If phosphate is also low then the algal cell count is high (96 mg/l). This indicates that the largest blooms (in the order of 140 mg/l) occur when algae have consumed most of the available free nitrogen (nitrate levels are low), but phosphate levels are not near to exhaustion.

This model does not, however, succeed in predicting all of the peaks in algal abundance. The high algal abundance in early 1986 and also in the training set are not predicted by this model. The average prediction of 20 independent runs using input set 2 shows that the average model also misses these algal abundance peaks. This suggests that either the driving forces for these peak abundances are not present in the input set, or that the concepts required to pre-

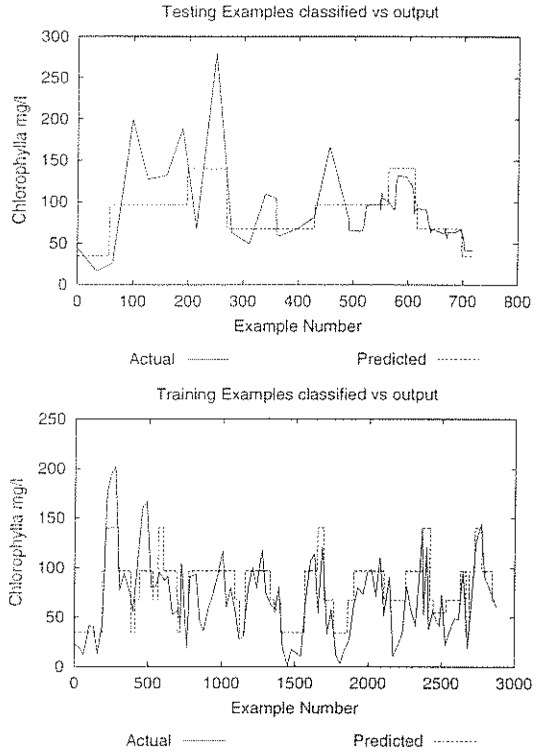

Figure 4: Training and testing output for input set 2

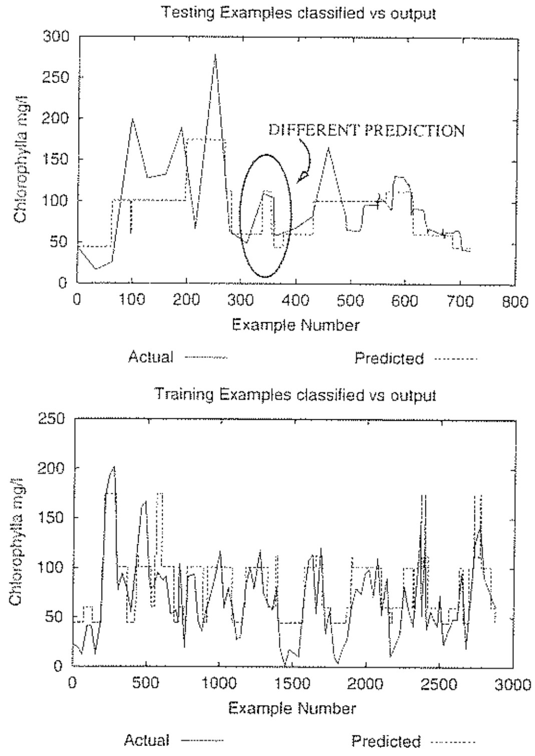

Figure 6: Training and testing output for input set 1

dict the algal abundance at this time cannot be represented or discovered by the learning algorithm.

3.2 Using all available inputs

Input set 1 contained all the available input parameters. Results from using this set on the unseen data were equal to the best results achieved, which demonstrates the evolution's ability to sort through redundant information and provide equivalent models. Since more information is available to the algorithm there is an increased chance of over-training. This did not occur, and the use of input set 1 allows us to compare different outputs from differently structured models which nevertheless achieve equivalent root mean square errors.

The different model structures produce qualitatively different predictions. In particular, the timing of high algal abundance are picked up by different model structures differently. The model detailed in figure 5 produced the prediction on the unseen testing years of 1986 and 1993 shown in figure 6. This model had a very good root mean square error of 35.76, and was able to pick up the algal abundance indicated in figure 6. This particular bloom was not predicted successfully by most other evolved models. The model in figure 5 used rule [5] to categorize this peak. That is, it was a time of low nutrient levels and the pH level was less than 9.16. Had the lake been in a more alkaline state the prediction would have been for an extremely high level of chlorophylla (174mg/l). This rule tells us that the peak chlorophylla level occurred at a time when the nutrient levels were low and pH was less than 9.16. It could be concluded that higher chlorophylla at approximately 174 mg/l would occur at pH levels above 9.16. Nevertheless, the validity of this conclusion is unconfirmed.

4 Conclusion

In situ time series hold unique information about ecosystem processes and behaviour. Inductive modeling techniques can be used to explore this information. Machine learning techniques offer a new quality of inductive modeling by extracting not only seasonal and annual patterns, but related connectivity between key variables as well.

Preliminary results in this paper show that in particular genetic algorithms can be used to extract and develop rules from water quality time series, that can be used for prediction and elucidation of timing and magnitudes of algal bloom events. Even though test results look already promising, the validity of some rules at this stage is yet to be confirmed.

Further efforts will be made to elucidate and predict chlorophylla by means of generated rules as an indication of algal blooms, before the abundance and succession of algal species will become the subject of rule-based modeling by genetic algorithms.

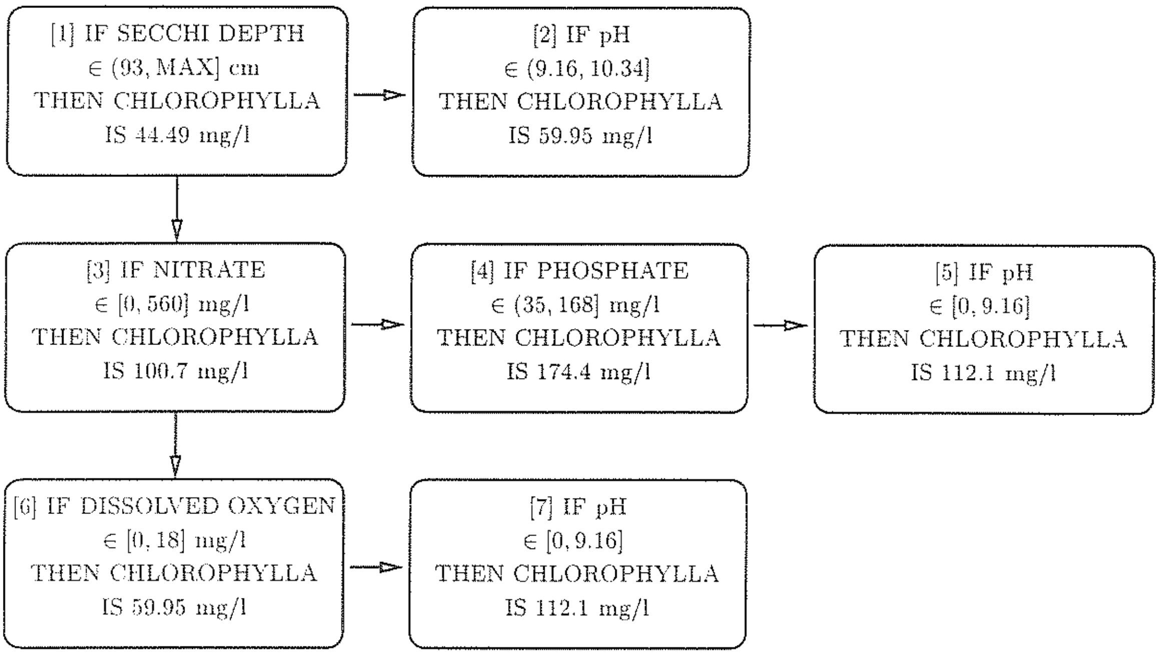

Figure 5: Rule Set 2

References

Bäck, T., *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, 198 Madison Avenue, New York, New York 10016, 1996.

Bobbin, J. and X. Yao, Evolving rules for nonlinear control, in M. Mohammadian, ed., *New Frontier in Computational Intelligence and its Applications*, IOS Press, Amsterdam, 1999, in press.

Fogel, D., *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ, 1995.

Holland, J. H., *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press, 1975.

Koza, J. R., *Genetic Programming*, MIT, Stanford University, Cambridge, MA, 1992.

Recknagel, F., Anna - artificial neural network model for predicting species abundance and succession of blue-green algae, *Hydrobiologia*, 349, 47–57, 1997.

Recknagel, F. and H. Wilson, Elucidation and prediction of aquatic ecosystems by artificial neural networks, *Ecological Modelling*, 1999, in press.

Schwefel, H.-P., *Evolution and Optimum Seeking*, John Wiley and Sons, 605 Third Avenue, New York, NY 10158-0012, United States of America, 1994.

Whigham, P. A. and F. Recknagel, Predictive modelling of plankton dynamics in fresh water lakes using genetic programming, *Submitted to MODSIM 1999*, 1999.