This page is for indexing purposes and will not be printed in the conference proceedings book.

The text area is 29 × 45 picas (~12.3 cm × 19 cm) to fit in the printed proceedings book without reduction. When printed on letter paper there will be wide margins.

Author(s)

First Name	Middle Name	Surname	Role
Salomão	S.	Madeiro	First Author

Affiliation

Organization	URL	Email
Department of Computing Systems Polytechnic School of Engineering - Pernambuco State University, Rua Benfica, 455 - Bloco "C" - 3o. Andar CEP: 50.720-001 – Recife/Pernambuco, Brazil	www.dsc.upe.br	salomao_sampaio@yahoo.com.br

Author(s)

First Name	Middle Name	Surname	Role
Flávio	R. da S.	Oliveira	Second Author

Affiliation

Organization	URL	Email
Department of Computing Systems Polytechnic School of Engineering - Pernambuco State University, Rua Benfica, 455 - Bloco "C" - 3o. Andar CEP: 50.720-001 – Recife/Pernambuco, Brazil	www.dsc.upe.br	flaros@hotmail.com

Author(s)

First Name	Middle Name	Surname	Role
Frederico	B. A.	Alexandre	Third Author

Affiliation

Organization	URL	Email
Department of Computing Systems Polytechnic School of Engineering - Pernambuco State University, Rua Benfica, 455 - Bloco "C" - 3o. Andar CEP: 50.720-001 – Recife/Pernambuco, Brazil	www.dsc.upe.br	fred.bralvalex@gmail.com

Author(s)

First Name	Middle Name	Surname	Role
Fernando	B. de	Lima Neto	Corresponding Author

Affiliation

Organization	URL	Email
Department of Computing Systems Polytechnic School of Engineering - Pernambuco State University, Rua Benfica, 455 - Bloco "C" - 3o. Andar CEP: 50.720-001 – Recife/Pernambuco, Brazil	www.fbln.pro.br	fbln@dsc.upe.br

Publication Information

Pub ID	Pub Name	Pub Date
ASABE will complete	*Computers in Agriculture 2006* (24-26 July, Lake Buena Vista, Florida, USA).	Pub. Date July 2006

Intelligent Modeling of Sugar-Cane Maturation

Salomão Sampaio Madeiro[1], Flávio Rosendo da Silva Oliveira[1], Frederico Bruno Alves Alexandre[1], Fernando Buarque de Lima Neto[1,2]

1) Department of Computing Systems – Polytechnic School of Engineering (*Escola Politécnica de Pernambuco*)
Pernambuco State University. Rua Benfica, 455 - Bloco "C" – 3o. Andar * CEP: 50.720-001. Recife / Pernambuco, Brazil.
2) Corresponding Author: fbln@dsc.upe.br

Abstract. Previous use of Artificial Intelligence (AI) in agriculture for forecasting productivity indicators, especially Artificial Neural Networks (ANN), has shown that it is possible to approximate sugarcane maturation curves. However, ANNs are widely known to offer some difficulties to be parameterized; normally, some heuristics and devotion by the users are necessary to provide a suitable parametrical selection. In this work we propose a tool that searches ANN parameters automatically (*i.e.* without direct interference of users in this task). For this high goal, we utilized another artificial intelligent technique: Genetic Algorithms (GA). The paper concludes showing results of ANNs which parameters have been provided by (the contributed) automated approach; comparisons to manually setup ANNs are included.

Keywords. Sugarcane, Harvest, Maturation, Artificial intelligence, Artificial neural networks, Genetic algorithms.

Introduction

Sugar cane is a genus of up to 37 species (not to mention the hybrids) of tall grasses that grow extensively on warm temperate to tropical regions of the globe. They are a chief source of sugar and fuel for millions of people world wide.

Greater sugar levels can be extracted if computer programs could model the maturation curve of each sugar cane variety. This is possible given the known productivity factors and conditions along the farming period [Lima Neto and Ludermir, 1997] [Lima Neto, 1998] [Trigo *et al*, 2005] [Pacheco *et al*, 2005].

The problem at hand has two parts: first, how to create a working model (*i.e.* including productivity factors) and, second, which parameters to ANNs to consider. It is worth mentioning that all these factors impact non-monotonically on the modeling process of different sugarcane varieties maturation.

In this work approach, the available historical data of the Management Information Systems (MIS) existing in most sugarcane mills are used to train an intelligent computer system based on artificial neural networks and genetic algorithms [Lima Neto, 1998]. After ANNs are conceived, Genetic Algorithms are used to automatically select a fair parametrical set-up for them [Russell, 2003].

Simulation results obtained from the trained model are confronted with real data. The reader can then evaluate how good can be the maturation curve approximation of sugarcane produced by both manual and automated approaches. These results are very useful to economically sound harvesting decisions.

In the following sections of this paper, some background aspects relating to Artificial Neural Networks and Genetic Algorithms are subsumed. Next, the contributed tool for parameterize ANNs is detailed. Finally, all experiments carried on are described and commented upon.

Background

Artificial Neural Network

Artificial Neural Network is a technique based on the functioning of human neural system. They present high level of parallelism, generalization ability and capability to adapt itself – learning by its own mistakes. ANNs are used with great efficiency in classification, function approximation and prediction problems.

A typical ANN is composed by small linear processing units called artificial neurons which are highly connected among them. One such unit computes values presented to its input and evokes a threshold value on its output, according to an internal activation function it also computes [Haykin, 1994].

When assembled in a network of neurons, such as the Multi-Layer Perceptron – MLP (Rumelhart, 1986), ANNs acquire non-linear capabilities. MLP topology was used in this work, because of its flexibility, generalization ability and computing power. They are able to solve any mathematical function [Cybenko, 1989], thus, suited to forecast productivity indicators [Lima Neto, 1998]; see that in Figure 1.

The learning capacity of ANN arises from weights adaptation of every artificial neuron (*i.e.* synapses). These weighs are adjusted during training phase, which is composed of two steps: propagation of data through input layer in forward direction and back propagation of errors from output layer.

Simulations included here were carried out by the Error Backpropagation Algorithm (Rumelhart, 1986).

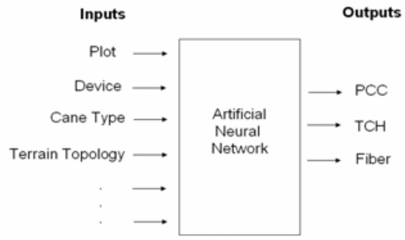

Figure 1. Schematic view of the ANN simulated in this work. Note sugarcane productivity factors and indicators – ANN Input/Output.

Genetic Algorithm

Genetic algorithm is another AI technique that is used to solve problems of search and optimization, based on principles of genetics and natural selection. It allows a population composed of many individuals to evolve, following specific rules for selection, until it finds a state which maximizes the fitness of each individual [Haupt, 2004].

GA is an effective global search method and is capable to surpass problems dealt by traditional techniques such as *hill climbing*. It is capable of simultaneously explore a wide area of the search surface and can avoid early convergence [Henderson, 2000]. Derivative information is not required to guide the search, but it requires a function capable of evaluating the merit of each candidate solution. GA has proven to be effective in finding a high quality solution even within a wide and complex search surface.

In vanilla GA, there are at least three basic components: chromosomes, fitness function and operators. It is common to use a bit string as chromosome, which is a representation of a possible solution to the problem. The combination of features of a chromosome is called a genotype. Phenotype is when a genotype is decoded into real world representation.

When an ANN parametrical set is encoded to be optimized through GA, some considerations must be made; according to [Balakrishnam, 1995] they are:

- Completeness: it must be possible to represent all phenotypes within search space
- Closure: all genotypes must be decoded to valid phenotypes
- Compactness: space efficiency in genotype, when encoding a phenotype
- Scalability: relates to how much a genotype must grow to encode a larger phenotype
- Complexity: relates to effects caused in complexity of GA, according to the choice of encoding

The fitness function is how GA measures the merit of a particular chromosome. The probability of a given chromosome survive is proportional to its fitness value. Operators are rules imposed by the environment the population to evolve. There are operators for selection, crossover and mutation [Tzung-Pei, 2002]. The selection is used to choose which elements will be combined via evolutionary processes. Crossover is the method by which two chromosomes will combine their genes in order to produce better offspring. Mutation occurs at the end of an evolutionary process, creating a random change in few chromosomes; that is, how genetic variability is introduced in the population and how premature convergence is avoided.

The processing cycle of GA is composed of the following steps: initial population generation, fitness function (of chromosomes) evaluation, selection of the fittest, crossover (among parents), mutation of features, and algorithm convergence check. This cycle is referred to as (new population) generation and can be performed many times until the GA reaches its objective.

Modeling Sugarcane Maturation via ANN

Modeling real world problem with Artificial Neural Network can be achieved in five steps: (i) dataset acquisition, (ii) data pre-processing, (iii) adequate model selection for the problem, (iv) ANN parametrical setup, and (v) ANN training & validation [Russell and Norvig, 2003].

The first step referred above, aims at gathering data about the problem to be modeled in amounts that allow the ANN training phase to acquire the most prominent features of it. Next, mathematical operations (*e.g.* normalizations, change of variable codification, dimensionality reduction etc) are carried out on the dataset in order to avoid bias among input variables. The model selection includes choice of neural architecture (i.e. type of ANN) that better incorporates features of the real problem; this choice is based on heuristics, demands previous experience on AI theory and a reasonable knowledge about the problem domain. However, there are some aspects of model selection that can only be defined thru experimentation, namely, number of hidden layers, number of neurons in hidden layers, learning rate, activation function, and other special parameters of particular algorithms. As the definition of all these parameters can take a long time, the contribution of this work can be very helpful to shorten this task because it automates the parametrical selection.

Prior to the implementation of the tool that helps on finding a suitable ANN parameterization; these parameters had to be chosen in order to be mapped by genes. As an example, in this work we chose: number of neurons in the first layer, number of neurons in the second layer, learning rate and activation function. Each gene is then represented as a set of bits. According to this implementation, a chromosome encapsulates the four parameters above representing them as chain of bits. The sequences of bits are dealt with as chains of characters, due to some facilities offered by the programming language utilized (*i.e.* Java language). Five bits were used to represent the number of neurons in the first and second layers, one bit to determine the activation function – *logistic or hyperbolic tangent*, and seven bits to code the learning rate (represented as the negative exponent of a power of two). Altogether, these choices lead possible configurations up to 2^{18}.

Given the codification above, the parametrical search process is carried upon the initial population of individuals, which was randomly generated. The total number of individual is an information provided by the user directly to the tool. Every individual of the population is equivalent to one distinct ANN's – complete – parametrical setup. That is, for each ANN configuration, there is an individual in the population and an ANN had to be evaluated. For ANN evaluations, they were (i) parameterized by the parameters coded in the individual's chromosome and (ii) submitted to a custom made neural engine developed by the authors. This neural engine was the same used for training, validation and testing of all neural networks.

As the fitness function, to guide the GA search for suitable solutions (*i.e.* an optimal neural network parametrical setup), we used the inverse MSE value obtained after training every neural network evaluated. After the evaluation process, individuals (*i.e.* their codified parametrical setup) were sorted by the quicksort algorithm and each individual is repositioned in accordance to its performance. Following fitness selection and convergence check, the process of creation of the next generation was started; see example in Figure 2.

A1	A2
B1	B2
C1	C2
D1	D2
E1	E2
F1	F2

Crossover →

A1	A2
A1	B2
B1	A2
B1	C2
C1	B2
C1	D2

Figure 2. ***An**, **Bn**, **Cn**, **Dn**, **En** and **Fn*** are gene values for each setup before (left) and after (right) crossover.

The first individual of each generation, after fitness evaluation, was automatically passed on to the next generation, thus preserving the best solution throughout generations. The overall search finishes after **k** evolutionary cycles, where **k** is also informed to the tool. Figure 3 illustrates all steps of the proposed tool.

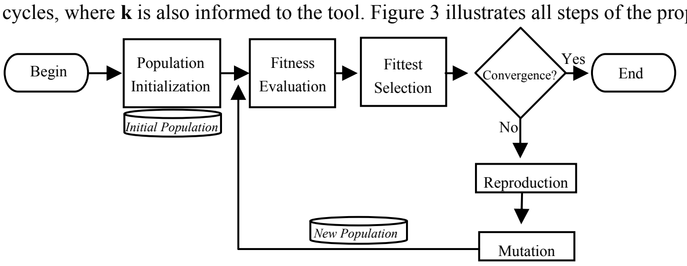

Figure 3. Flowchart of Genetic Algorithm steps comprising the proposed tool.

For modeling sugar cane maturation one could profit by having a function approximation tool such as ANN. As these architectures are hard to configure, we came across with the idea of using GA to search for a suitable parametrical setup for these ANN. Figure 4 illustrates the overview of the complete solution. In the figure, the ascending path (upward arrows) represent how data come from the sugarcane mill MIS towards decision support systems (DSS), passing thru a neural processing phase – now automatically parameterized by the presently contributed tool. The descending path of information (downward arrows) represents how decisions based on DSS can be feedback to: (i) neural processing, (ii) ANN setup tools and (iii) production.

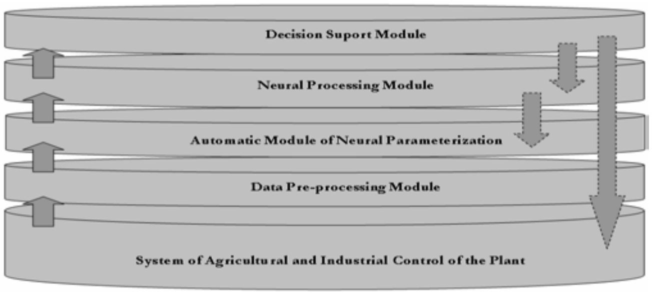

Figure 4. Overview of all modules comprising the complete solution for modeling sugarcane maturation.

Experiments

To validate the ideas and tool put forward here, some experiments were carried on in order to automatically find values for two important ANN attributes: neurons per (hidden) layer and learning rate. As external parameters for the GA we choose to vary: number of individuals in the population (which is the number of distinct ANN architectures the tools is supposed to investigate at each time) and the number of generations (which means how many evolutionary cycles will be carried on within each experiment).

Table 1. Settings for the best individual on Experiments

Experiments	Generation Number	Number of Individuals	Neurons per layer	Learning Rate
1	10	10	[06]	0.091
2	10	15	[12][06]	0.125
3	10	20	[10][01]	0.125
4	10	25	[03][03]	0.091
5	15	10	[05][01]	0.091
6	15	15	[06]	0.091
7	15	20	[04][01]	0.125
8	15	25	[04]	0.091
9	20	10	[06][08]	0.125
10	20	15	[08][04]	0.2
11	20	20	[05][01]	0.091
12	20	25	[08][10]	0.091

We carried on 12 experiments, in which we varied maximal number of generations as 10, 15 or 20; the number of individuals for each generation as 10, 15, 20 or 25; see Table1. These experiments yield good parameterizations to the ANNs. The evolved ANN settings were automatically attributed to real nets that were evaluated by the fitness function explained above (based on forecasts of each ANN). An important remark is that same ANN architectures with distinct parameterizations can yield different forecasts.

The best result obtained from the 12 experiments generated parameters utilized to setup one ANN that was able to forecast the productivity indicators displayed in Figure 5, Figure 6 and Figure 7. All the best individuals of each experiment had the *logistic function* as their activation function and the *learning rate* within [0.091, 0.2]. It was also found that these two parameters were more important than any other, tested.

The dataset used to train all networks here was the same used in previous work [Buarque, 1998], [Pacheco *et al*, 2005] and [Trigo *et al*, 2005]. Productivity indicators forecast were PCC[1], TCH[2] and Fiber[3]. The forecast precision rate of these indicators, obtained from the fittest individual of each experiment, can be seen in the charts of Figure 5, Figure 6 and Figure 7, respectively.

[1] Percentage sucrose of sugarcane (of a plot);
[2] Tons of biomass (sugarcane) per hectare (of a plot);
[3] Total of (combustible) dry matter from biomass after milling (of a plot);

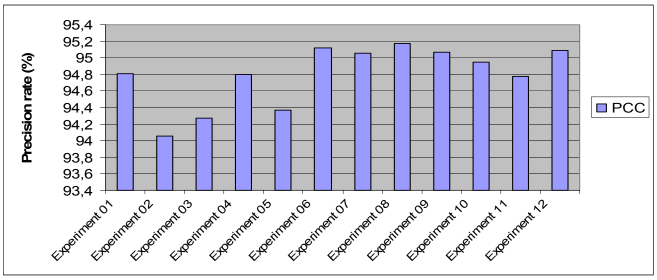

Figure 5. Forecast precision of *PCC* for the best individuals of each experiment

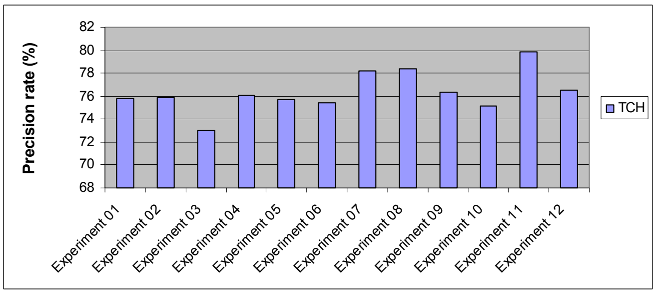

Figure 6. Forecast precision of *TCH* for the best individuals of each experiment

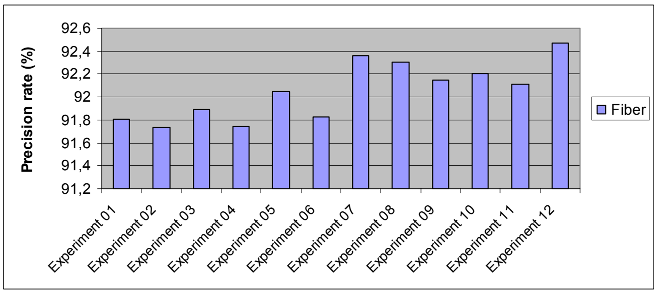

Figure 7. Forecast precision of *Fiber* for the best individuals of each experiment

We compared forecast results produced by the best ANN found by this work approach to previous works in the same experimental conditions [Buarque, 1998], [Pacheco *et al*, 2005] and [Trigo *et al*, 2005]. It is important to notice that the best ANN (i.e. experiment number 11), had its parameters found automatically by the tool introduced here. Table 2 details the precision of all productivity indicators considered among different works. In the table, one can notice that the ANNs simulated in this work experiments, despite of being automatically parameterized, have produced results that are quite compatible to previous ones.

Table 2. Comparison of Forecast precisions among different work

Indicators	**[Buarque, 1998] (%)**	**[Pacheco *et al*, 2005] (%)**	**[Trigo *et al*, 2005] (%)**	**This work–Exp # 11 (%)**
PCC	95.330	95.620	95.401	94.777
TCH	49.200	78.070	79.475	79.910
Fiber	89.680	92.520	92.736	92.110
Average	**78.070**	**88.737**	**89.204**	**88.932**

In addition to save time of users during parametrical selection of ANNs, the current approach has produced neural networks that converged faster than any other architecture of related works. For instance: 10000 cycles in Pacheco's work (2005), exactly 2795 cycles in Trigo's work (2005), while we used here a maximal value of 2000 training cycles. It is worth mentioning that training cycles of ANN can be increased for better predictions, should it be necessary.

Conclusion

The use of GA in automatic search for some ANN parameters is the main contribution of this work. Here those neural networks are used as models of sugarcane maturation. In which context, the information produced by the tool put forward here is highly useful for harvesting decisions.

The performances of forecasts obtained are compatible to similar ones of recent works, in which ANN parameters were selected manually [Trigo *et al*, 2005]. Therefore, the approach and the computer tool put forward here not only help decision makers during harvest, but it also speeds-up the computation time.

As future work, we intended to improve some GA aspects of the algorithm, *e.g.* to use a more aggressive strategy of evolution, namely, the method of the roulette. Other ANN parameters are also intended to be automatically sought in order to increase tool automation; probable candidates are: maximum number of cycles of training (and other stopping criteria), training algorithm type and momentum term. With all this, we expect to increase the efficiency of the evolutionary process and, thus, to improve further end results.

References

C. E. Henderson and Walter D. Potter, *Predicting Aflatoxin Contamination in Peanuts: A Genetic Algorithm/Neural Network Approach*, in Applied Intelligence 12, 2000, pp. 183–192

Diogo Pacheco, Filipe Regueira and Fernando Buarque de Lima Neto. Using Artificial Neural Networks on Sugarcane Harvest to Predict Sucrose, Fiber and Productivity (in Portuguese), in *ALCOOLbrás*. São Paulo, Brazil, v. 90, pp. 60-63, 2005.

Fernando Buarque de Lima Neto and Teresa Bernarda Ludermir. Decision Support on Sugarcane Harvest (in Portuguese), in *IV Brazilian Symposium on Neural Networks – SBRN*. Goiania, Brazil, 1997.

Fernando Buarque de Lima Neto. *Managerial Decision Support, based on Artificial Neural Networks* (in Portuguese), Master Dissertation presented to Department of Informatics, Federal University of Pernambuco, Recife, Brazil, 1998.

George Cybenko. *Approximation by superposition of a sigmoid function*. Mathematics of Control, Signal and Systems. 1989.

James McClelland, David Rumelhart, and Geoffrey Hinton. The Appeal of Parallel Distributed Processing, in D. Rumelhart, J. McClelland and PDP Research Group (Editors). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986.

Karthik Balakrishnan and Vasant Honavar, *Properties of genetic representations of neural architectures*, in Proceedings of the World Congress on Neural Networks, 1995, pp. 807–813.

Simon Haykin. *Neural Networks – A Comprehensive Foundation*. Prentice-Hall International Editions. New Jersey, USA, 1994.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.*2nd Edition. Prentice-Hall International Editions. New Jersey, USA, 2003.

Randy L. Haupt and Sue E. Haupt. *Practical Genetic Algorithms*. 2nd ed. Wiley-Interscience, 2004.

Tiago Trigo, Paulo Batista Jr. and Fernando Buarque de Lima Neto. Artificial Neural Networks on Sugarcane Harvest (in Portuguese), in *Proceeding of V Brazilian Congress of Agroinformatics*. Londrina, Brazil, 2005.

Tzung-Pei Hong, Ke-Yuan Huang, Wen-Yang Lin: Applying genetic algorithms to game search trees. Soft Comput. v. 6(3-4) pp. 277-283, 2002.