UNIVERSITÄT
KOBLENZ · LANDAU
Institut für Wirtschafts- und Verwaltungsinformatik

FB 4
Informatik

A universal simulator based on artificial neural networks for computer clusters

Oleg V. Kryuchin
Alexander A. Arzamastsev
Klaus G. Troitzsch

Nr. 2/2011

Arbeitsberichte aus dem Fachbereich Informatik

Universität Koblenz-Landau • Campus Koblenz • Universitätsstraße 1 • 56070 Koblenz
Telefon +49 261 287-0 • http://www.uni-koblenz.de

Arbeitsberichte des Fachbereichs Informatik

ISSN (Print): 1864-0346
ISSN (Online): 1864-0850

Herausgeber / Edited by:

Der Dekan:
Prof. Dr. Zöbel

Kontaktdaten der Verfasser

Klaus G. Troitzsch, Oleg Kryuchin, Alexander Arzamastsev
Institut für Wirtschafts- und Verwaltungsinformatik
Fachbereich Informatik
Universität Koblenz-Landau
Universitätsstraße 1
D-56070 Koblenz
EMail: kgt@uni-koblenz.de

A universal simulator based on artificial neural networks for computer clusters

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch

Abstract

This paper describes parallel algorithms for training artificial neural networks. Possible levels of parallelity are presented. Experiments for checking the efficiency of algorithms are discussed.

Keywords: artificial neural networks, computer clusters, parallel algorithms.

1 Introduction

1.1 Background and aims

Nowadays artificial neural networks (ANN) are used in different branches of science [8]. Especially they occur in such sciences as psychology, sociology and economics. The reason why they are used in these sciences is the peculiarity of the objects of these sciences as these objects cannot be easily defined by simple rules. The usage of artificial neural networks can cost a lot of time because neural network models have a complex structure and many parameters. One possible solution of this problem is using parallel algorithms and computer clusters.

The aim of this paper is to improve the efficiency of the deveopment of ANN-models by means of parallel algorithms. To achieve this aim, several tasks have to be solved. These tasks are the analysis of software products appropriate for using ANN, the development of parallel algorithms of training ANNs, the development of an information system with parallel algorithms of training ANNs and the development of numerical experiments to check the efficiency of these algorithms.

1.2 Current products

Nowadays there are no neural networks simulators for computer clusters. There are simulators for single PCs (JavaNNS, NeuroShell, NNC, Mathlab neural network tools etc.) but they cannot be executed on computer clusters. There are tools for computer clusters (T-System etc.) but they do not consider the specific requirements of neural networks simulation. And there are hardware solutions (Nimfa and etc.) which do not use computer clusters and cannot be configured for concrete task. So there are not products which bring together the technologies of artificial neural networks and of parallel algorithms.

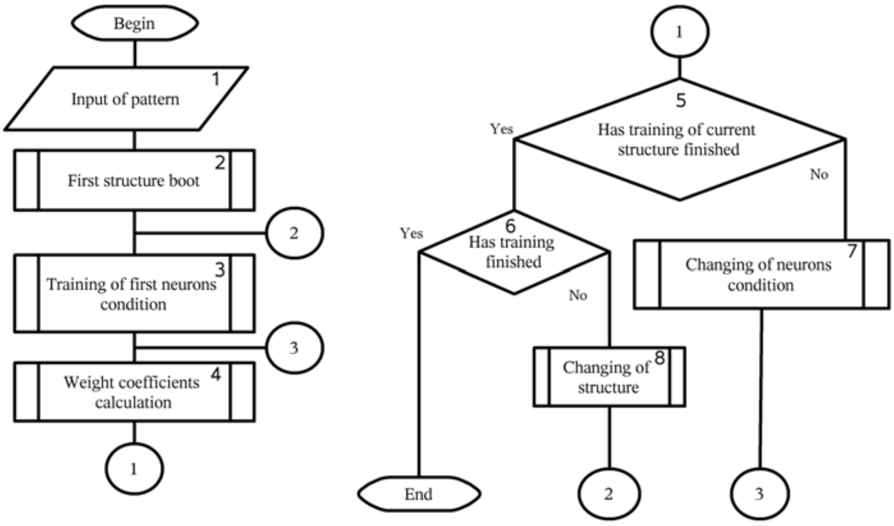
2 Training neural networks

The main task in trining a neural network is the minimization of a target function (1).

$$\varepsilon = \sum_{i=0}^{N-1} \sum_{j=0}^{P-1} (d_{i,j} - F(\vec{x}_i, \vec{w}, \vec{\mu})_j)^2 \tag{1}$$

where $\vec{x}$ is input pattern, $\vec{d}$ is output pattern, F is ANN output value calculation function, $\vec{w}$ are weight coefficients, $\vec{\mu}$ are activation functions, N is pattern row count and P is ANN output neurons count.

This main task consist of three levels: the level of changing the structure, the level of changing the activation functions of the neurons and the level of changing the weight coefficients. Figure 1 shows the general algorithm of training artificial neural networks.


Begin
1
Input of pattern
2
First structure boot
2
3
Training of first neurons condition
3
4
Weight coefficients calculation
1
1
5
Has training of current structure finished
Yes
No
6
Has training finished
Yes
No
7
Changing of neurons condition
8
Changing of structure
End
2
3

Pic. 1: The scheme of training of artificial neural network.

It is possible to develop parallel algorithms of training on each level:

- on the level of calculaing the value of the target function,
- on the level of updating the weight coefficients,
- on the level of searching the neuron activation function,
- on the level of training the structure.

Each of these will be discussed in the next section.

3 Parallel calculation on different levels

3.1 Parallel calculation of the value of the target function

If there are n processors and the pattern consists of N rows then each processor can calculate the inaccuracy for part of the pattern (equation 3) for processor number k and equation 2 for the zero (lead) processor. Then the zero processor sums up (formula 4). So the lead processor sends M rows to each processor, calculates the inaccuracy by $\hat{M}$ rows, receives the inaccuracies from the other processors and calculates the result. Values of M and $\hat{M}$ are calculated by formulas 5–6 [3].

$$\varepsilon_0 = \sum_{i=0}^{\hat{M}-1} \sum_{j=0}^{P-1} (d_{i,j} - F(\vec{x}_i, \vec{w}, \vec{\mu})_j)^2 \qquad (2)$$

$$\varepsilon_k = \sum_{i=0}^{M} \sum_{j=0}^{P-1} (d_{i+(k*M)+\hat{M},j} - F(\vec{x}_{i+(k*M)+\hat{M}}, \vec{w}, \vec{\mu})_j)^2, \quad k > 0 \qquad (3)$$

$$\varepsilon = \frac{1}{N} \sum_{k=0}^{n-1} \varepsilon_k \qquad (4)$$

$$M = \begin{cases} \left[\frac{N}{n}\right], & N \; mod \; n = 0; \\ \left[\frac{N}{n-1}\right], & N \; mod \; n \neq 0; \end{cases} \qquad (5)$$

$$\hat{M} = \begin{cases} M, & N \; mod \; n = 0; \\ N - M(n-1), & N \; mod \; n \neq 0; \end{cases} \qquad (6)$$

3.2 Parallel training of weight coefficients

There are different methods for the level of updating weight coefficients training. These are the method of full scanning, the Monte-Carlo method and several gradient methods. Each method uses unique parallel training.

3.2.1 Method of full scanning

This method searches all variants of weight coefficients values. So the value of the i-th weight coefficient in the I-th iteration is calculated by formula (7). The method iterates I_F times (the value of I_F is calculated by formula (9)).

$$w_i^{(I)} = l_0^{down} + s_i \times \left(\left[\frac{I}{\hat{s}(i)} \right] \; mod \; \left(\left[\frac{l_i^{up} - l_i^{down}}{s_i} \right] + 1 \right) \right) \qquad (7)$$

$$\hat{s}(i) = \begin{cases} 1, & i = l_w - 1; \\ \prod_{j=i+1}^{l_w-1} \left(\left[\frac{l_j^{up} - l_j^{down}}{s_j} \right] + 1 \right), & i < l_w - 1; \end{cases} \qquad (8)$$

$$I_F = \prod_{i=0}^{l_w-1} \left(\left[\frac{l_j^{up} - l_j^{down}}{s_j} \right] + 1 \right) \qquad (9)$$

Here l_i^{up}, l_i^{down} are the upper and lower limits of the i-th weight coefficient, s_i is the step of the i-th scanning of the weight coefficient, and l_w is the weights count.

If there are n processors then each nonzero processor iterates J_F times (formula (10)) ,and the lead processor iterates $\hat{J}_F$ times (formula (11)).

$$J_F = \begin{cases} \left[\frac{I_F}{n}\right], & I_F \; mod \; n = 0; \\ \left[\frac{I_F}{n-1}\right], & I_F \; mod \; n \neq 0; \end{cases} \qquad (10)$$

$$\hat{J}_F = \begin{cases} J_F, & I_F \; mod \; n = 0; \\ I_F - J_F(n-1), & I_F \; mod \; n \neq 0; \end{cases} \tag{11}$$

3.2.2 Monte-Carlo method

In this method, the weight coefficients are initialised with random values $\vec{r} = (r_0, r_1, \ldots, r_{l_w - 1})$. In each iteration each processor generates k points $\vec{w}_k^{(I)}$ of weight coefficients in the neighbourhood of $\vec{w}^{(I-1)}$ and calculates the inaccuracy $\varepsilon_k^{(I)}$ for each of these points. Then the zero processor chooses the minimum inaccuracy ($\varepsilon^{(I)} = \min\limits_k(\varepsilon_k^{(I)})$). If this inaccuracy is less than the current inaccuracy then the new coefficients will be set to network [5].

3.2.3 Gradient methods

Gradient methods are based on the calculation of a gradient vector and change the values of the weight coefficients in the opposite direction:

$$\vec{w}^{(I)} = \vec{w}^{(I-1)} - \vec{s}^{(I)} \nabla \varepsilon^{(I)} \tag{12}$$

Here $\nabla \varepsilon = (\frac{\partial \varepsilon}{\partial w_0}, \frac{\partial \varepsilon}{\partial w_1}, \ldots, \frac{\partial \varepsilon}{\partial w_{l_w - 1}})$ is the gradient.

This is the general idea but different gradient methods (for example QuickProp and RPROP) can use other similar formulas.

In these algorithms the vector of weight coefficients is divided into n parts and each part is located on a different processor. The lead processor calculates $\hat{N}_w$ weight coefficients, and the other processors calculate $\hat{l}_w$. The values of $\hat{l}_w$ and $\hat{N}_w$ are calculated by formulas (13)-(14) [3].

$$\hat{l}_w = \begin{cases} \left[\frac{l_w}{n}\right], & l_w \; mod \; n = 0; \\ \left[\frac{l_w}{n-1}\right], & l_w \; mod \; n \neq 0; \end{cases} \tag{13}$$

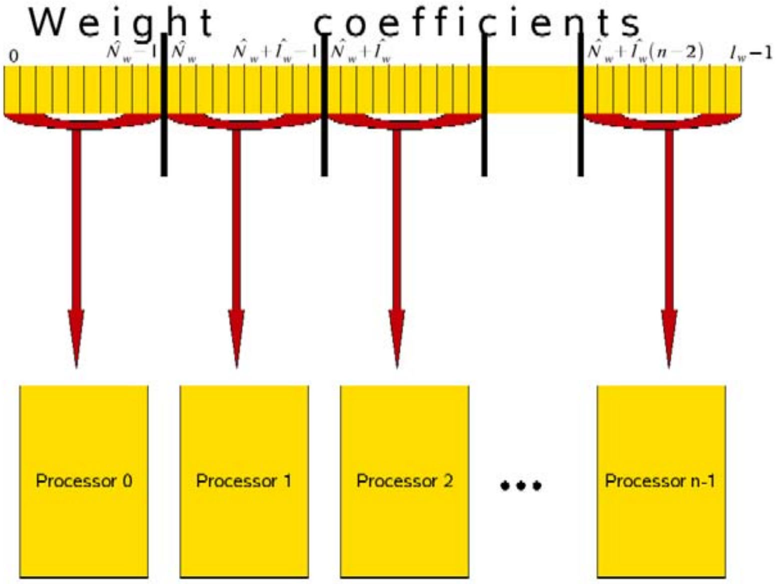
$$\hat{N}_w = \begin{cases} \hat{l}_w, & l_w \; mod \; n = 0; \\ l_w - \hat{l}_w(n-1), & l_w \; mod \; n \neq 0; \end{cases} \tag{14}$$

3.3 Parallel search of the neuron activation function

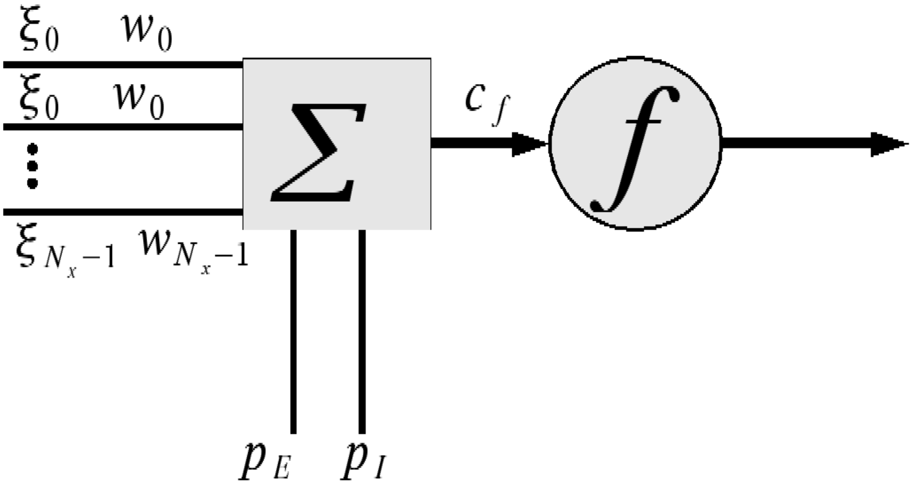
A universal mathematics neuron consists of two parts — an adder and an activation function (pic. 3). An adder calculates the sum of products of input signal values and weight coefficients. The output neuron value is calculated by formula (15). So a neuron has several search parameters — two impulses (internal and external), the activation function coefficient(s) and the type of activation function. Several types of functions (for example threshold functions) need some other parameters.

$$y = f\left(c_f\left(\sum_{i=0}^{N_x - 1}(\xi_i w_i) + p_I + p_E\right)\right) \tag{15}$$

Here c_f is the coefficient of activation function, p_I, p_E are the internal and the external impulse (appendant weights), $\vec{\xi}$ contains the input signals, N_x is the count of input signals, and f is a function of some type.



Pic. 2: The location of weight coefficients on processors.



Pic. 3: Universal mathematics neuron.

Training the activation function consists of searching all these parameters. It needs a full scanning of all variants. It symbolizes all possible values of the i-th activation neuron function as $\vec{\nu}_i$ (first function value is $\vec{\nu}_{i,0}$) and size of this vector as $N(\vec{\nu}_i)$. So there will be I_μ iterations (this is calculated by formula (16)), and the activation function value of the i-th neuron in the I-th iteration is calculated by formula (17).

$$I_\mu = \prod_{i=0}^{l_\mu-1} N(\vec{\nu}_i) \tag{16}$$

$$\mu_i^{(I)} = \nu_{\iota,i}, \quad \iota = \left[\frac{I}{\hat{s}(i)}\right] \; mod \; N(\vec{\nu}_i) \tag{17}$$

$$\hat{s}(i) = \begin{cases} 1, & i = l_\mu - 1; \\ \prod_{j=i+1}^{l_\mu - 1} N(\vec{\nu}_j), & i < l_\mu - 1; \end{cases} \tag{18}$$

Here l_μ is the number of neurons.

If there are n processors then each nonzero processors iterates J_μ times (formula (19)), and the lead processor iterates $\hat{J}_\mu$ times (formula (20)).

$$J_\mu = \begin{cases} \left[\frac{I_\mu}{n}\right], & I_\mu \ mod\ n = 0; \\ \left[\frac{I_\mu}{n-1}\right], & I_\mu \ mod\ n \neq 0; \end{cases} \tag{19}$$

$$\hat{J}_\mu = \begin{cases} J_\mu, & I_\mu \ mod\ n = 0; \\ I_\mu - J_\mu(n-1), & I_\mu \ mod\ n \neq 0; \end{cases} \tag{20}$$

3.4 The parallel training of the neural network for its structure

Usually in the beginning the structure is minimal, and then new neurons and layers are appended and inserted. Input neurons count is equal input pattern size and output neurons count depends from output pattern size. In the beginning the structure has one hidden layer witch consists of one neuron (if it is possible for using structure type). After training for structure (selection of activation functions and weight coefficients) the inaccuracy value is checked. If this value is large then a new neuron is added in the hidden layer and the network is again trained for structure. Such an operation is repeated until adding neurons leads to a depreciation of the inaccuracy . After this a new hidden layer with one neuron is added. The training finishes after the inaccuracy is below a threshold value (or when some other stopping criterion is met) [1].

The lead processor builds the structure and sends it to some free processor, and this processor begins to search activation functions and weight coefficients. After the end of the training this processor sends the structure back to the zero processor and receives a new structure. This process continues until a stop constraint is executed.

4 Estimating the performance of parallel algorithms for ANN training

If a serial training algorithm spends t hours, then the time expense of an ideal parallel algorithm τ_I is calculated by formula (21). But ideal time expense is impossible because such effect is possible only in particular cases. So the real time expense of parallel training is calculated by formula (22).

$$\tau_I = \frac{t}{n} \tag{21}$$

Here n is the number of processors.

$$\tau_R = \frac{t}{n} + \Phi(n) \tag{22}$$

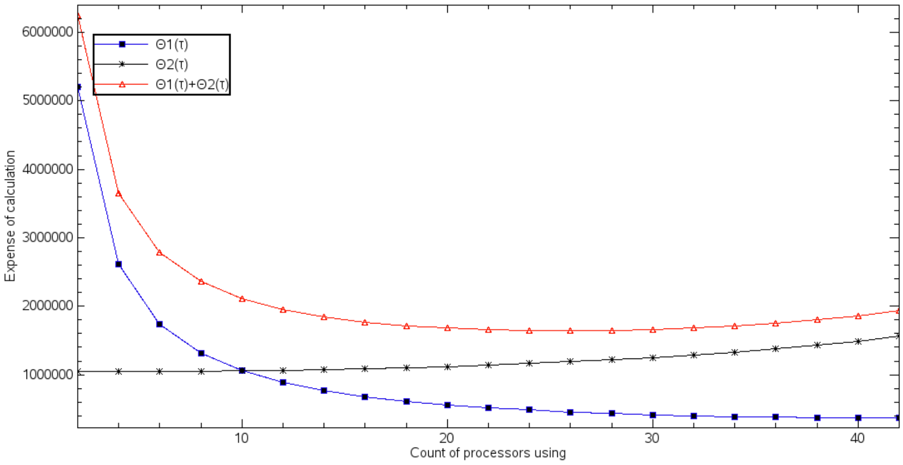
Here $\Phi(n)$ is the time expense for preparing and sending data.

So the efficiency coefficient may be calculated by formula (23).

$$\alpha = \frac{t}{\tau_R n} \tag{23}$$

where t, τ_R is the time expense of serial and parallel algorithms, and n is the numbver of processors.

The increase of the number of processors involves a speedup of training. But after several moments the time expenses begin to rise. The reason for this is the large time expense for transferring data from one processors to some other. So there is an optimal number of processors for each task. This count depends on the expense of calculation waiting ($\Theta_1(\tau)$ in pic. 4) and the expense of computer cluster using ($\Theta_2(\tau)$ in pic 4). The value $\Theta_1(\tau) + \Theta_2(\tau)$ in pic 4 shows the total expense.



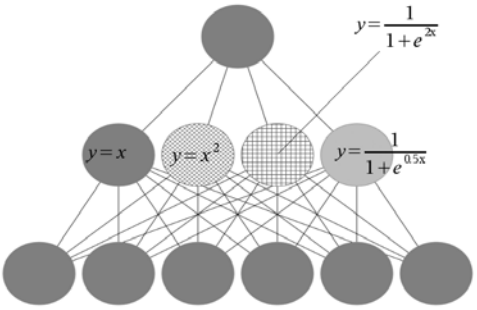
Pic. 4: Dependence of expense from processors count.

5 Experiments

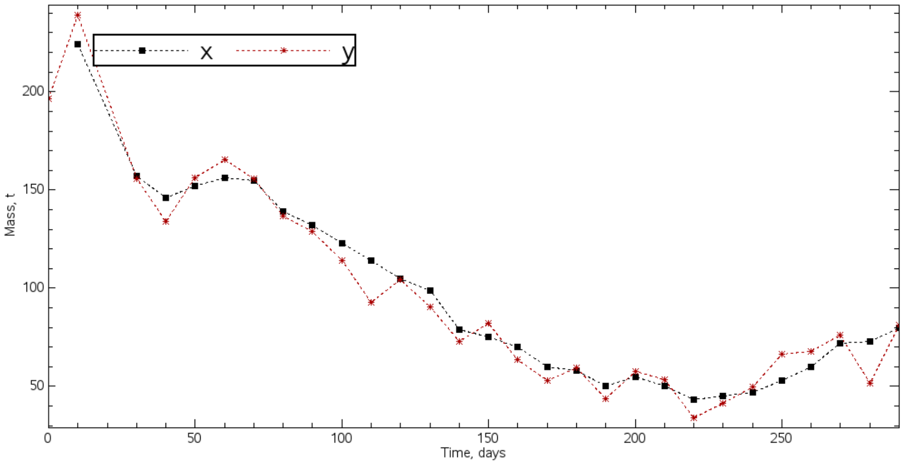
For checking the efficiency of parallel algorithms several experiments were done. All experiments were done on the computer cluster of Tambov State University, on the computer cluster of Tambov State Technical University and on the computer cluster of Moscow Calculation Center. ANN models were built by serial and parallel algorithms (using 4, 6 and 8 processors) and coefficient of efficiency was calculated. There were gradients training algorithms and there were two levels – the level of target function value calculation and the level of weight coefficients calculation.

5.1 Caught shrimp mass forecasting

The first experiment is forecasting the mass of caught shrimp. This experiment uses a multilayer perceptron [6]. Pic. 5 shows the obtained ANN structure and pic 6 shows the empirical and predicted values of shrimp mass. The values of the efficiency coefficient can be found in table 3.



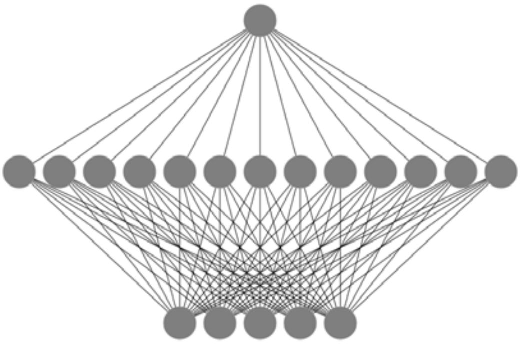
Pic. 5: ANN structure in the experiment on forecasting caught shrimp mass.



Pic. 6: Empirical (x) and forecast (y) values of the mass of caught shrimp.

5.2 Air temperature forecasting

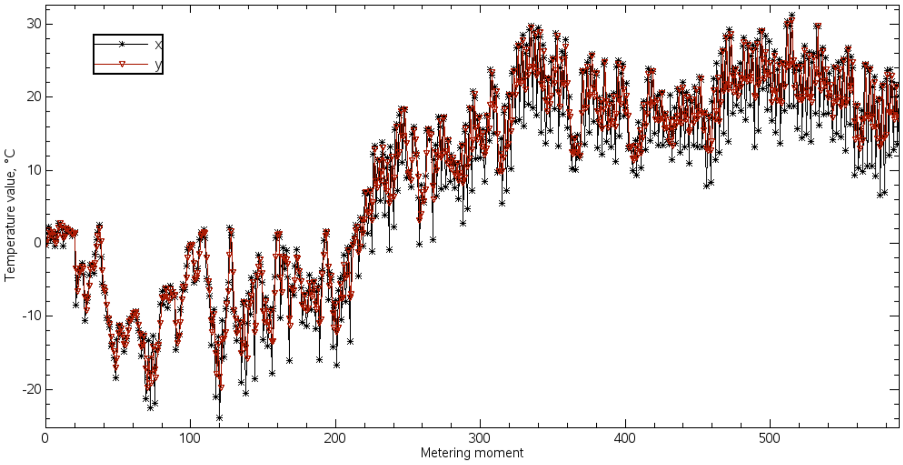
The second experiment is the forecast of the air temperature in Tambov (Russia)[7]. The values are shown in pic. 8, the structure is shown in pic. 7, and the coefficients of efficiency are shown in table 2.



Pic. 7: ANN structure in forecasting of air temperature.

Tab. 1: Coefficient of efficiency values for the experiment on forecasting caught shrimp mass

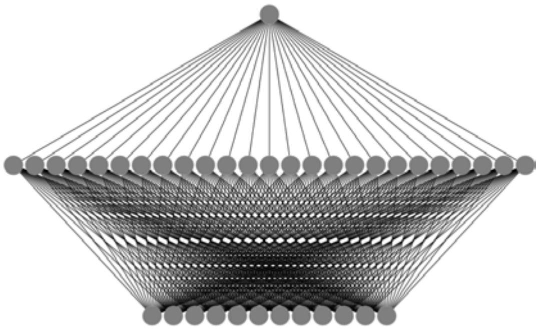
	Cluster TSU	Cluster TSTU	Cluster MCC
α_{wG} (4 proc)	0.9235	0.9199	0.9297
α_{wG} (6 proc)	0.9221	0.9157	0.9291
α_{wG} (8 proc)	0.9204	0.9105	0.9283
α_{ε} (4 proc)	0.8921	0.8822	0.9091
α_{ε} (6 proc)	0.8906	0.8793	0.9079
α_{ε} (8 proc)	0.8891	0.8769	0.9067



Pic. 8: Empirical (x) and forecast (y) values of air temperature.

5.3 Forecast of currency exchange rates

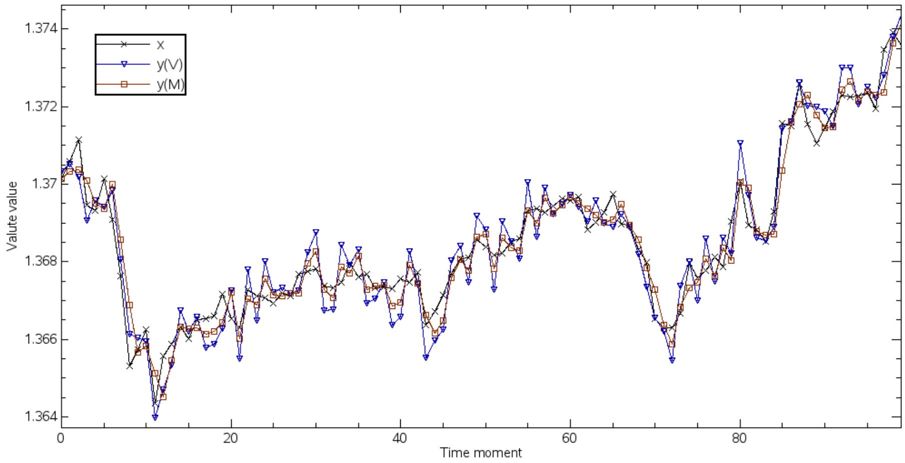
The third experiment is the forecasting of exchange rates between Euro and US dollar. This experiment uses a multilayer perceptron (MLP) and a Volterry network [4]. Values are shown in pic. 9, the MLP-structure is shown in pic. 10, and coefficients of efficiency are shown in table 3.



Pic. 9: ANN structure in forecasting of currency exchange rates.

Tab. 2: Coefficient of efficiency values for experiment of air temperature in Tambov

	Cluster TSU	Cluster TSTU	Cluster MCC
α_{wG} (4 proc)	0.9312	0.9287	0.9401
α_{wG} (6 proc)	0.9304	0.9273	0.9393
α_{wG} (8 proc)	0.9297	0.9259	0.9387
α_{ε} (4 proc)	0.9124	0.8974	0.9199
α_{ε} (6 proc)	0.9112	0.8951	0.9182
α_{ε} (8 proc)	0.9103	0.8936	0.9169



Pic. 10: Empirical exchange rates (x), exchange rates forecast by multilayer perceptron (y(M)) and by Volterry (y(V)).

5.4 Social object modeling

The last experiment is social object simulation. Social object here is the dependence of schoolboys' professional ability on their personal characteristics [2]. Coefficients of efficiency are shown in table 4.

6 Conclusion

So this paper brings together the technology of artificial neural networks and the technology of parallel algorithms. Aning information system with parallel algorithms of ANN training was developed. The theoretical expressions of defining of efficiency of parallel algorithms were deduced, and numerical experiments for check the efficiency of these algorithms were done.

Tab. 3: Coefficient of efficiency values for experiment of forecasting currency exchange rates

	Cluster TSU	Cluster TSTU	Cluster MCC
α_{wG} (4 proc)	0.9319	0.9281	0.9438
α_{wG} (6 proc)	0.9302	0.9271	0.9392
α_{wG} (8 proc)	0.9293	0.9261	0.9389
α_{ε} (4 proc)	0.9127	0.8975	0.9202
α_{ε} (6 proc)	0.9111	0.8949	0.9181
α_{ε} (8 proc)	0.9101	0.8932	0.9165

Tab. 4: Coefficient of efficiency values for the experiment of social object simulation

	Cluster TSU	Cluster TSTU	Cluster MCC
α_{wG} (4 proc)	0.9212	0.9103	0.9396
α_{wG} (6 proc)	0.9212	0.9103	0.9391
α_{wG} (8 proc)	0.9211	0.9098	0.9373
α_{ε} (4 proc)	0.9129	0.8977	0.9205
α_{ε} (6 proc)	0.9114	0.8951	0.9188
α_{ε} (8 proc)	0.9103	0.8935	0.9169

References

[1] Arzamastsev A, Kryuchin O., Azarova P., Zenkova N. Universal program complex for computer modeling on the basis of the artificial neural networks with self-organizing structure. // Tambov University Review. Series: Natural and Technical Sciences. Tambov, 2006. Vol. 11, Issue. 4 P. 564-570.

[2] Arzamastsev A, Kryuchin O., Zenkova N., Sletcov D. The neural network simulation of social object with cluster system using. // Tambov University Review. Series: Natural and Technical Sciences. Tambov, 2010. Vol. 14, Issue. 5 P. 372-375.

[3] Kryuchin O., Parallel algorithms of artificial neural networks training. // Matters of XV international conference on neurocybernetic science. Vol. 2. Symposium "Interface "Brain-Computer""3-rd Symposium on neural computer science and neurocomputers. Rostov-on-Doc. YSU 2009. P. 93-97.

[4] Kryuchin O., Arzamastsev A. The forecasting of valute pairs by artificial neural network // Tambov University Review. Series: Natural and Technical Sciences. Tambov, 2009. Vol. 14, Issue. 5. P. 591-596.

[5] Kryuchin O., Arzamastsev A. The parallel selection of artificial neural network weights coefficients by Monte-Carlo method // Science potential of youth – Russia future [Electronic resource]: II All-Russian scientific Zvorikin readings. Murom, 2010. P.865-866.

[6] Kryuchin O., Kozadaev A., Arzamastsev A. Parallel algorithms of artificial neural networks training and its using for forecasting of shrimp population numerosity in Indian ocean // Tambov University Review. Series: Natural and Technical Sciences. Tambov, 2010. Vol. 15, Issue. 5. P. 185-190.

[7] Kryuchin O., Kozadaev A. The forecasting of air temperature in Tambov by artificial neural networks with clusters systems using // Intellectual systems: Transactions of IX-th international simposium. Moscow. 2010. P. 620-624.

[8] Osowsky S. Sieci neuronowe w ujeciu algorytmicznym // Warszawa. 1996.

Bisher erschienen

Arbeitsberichte aus dem Fachbereich Informatik

(http://www.uni-koblenz-landau.de/koblenz/fb4/publications/Reports/arbeitsberichte)

Klaus G. Troitzsch, Oleg Kryuchin, Alexander Arzamastsev, A universal simulator based on artificial neural networks for computer clusters, Arbeitsberichte aus dem Fachbereich Informatik 2/2011

Klaus G. Troitzsch, Natalia Zenkova, Alexander Arzamastsev, Development of a technology of designing intelligent information systems for the estimation of social objects, Arbeitsberichte aus dem Fachbereich Informatik 1/2011

Kurt Lautenbach, A Petri Net Approach for Propagating Probabilities and Mass Functions, Arbeitsberichte aus dem Fachbereich Informatik 13/2010

Claudia Schon, Linkless Normal Form for ALC Concepts, Arbeitsberichte aus dem Fachbereich Informatik 12/2010

Alexander Hug, Informatik hautnah erleben, Arbeitsberichte aus dem Fachbereich Informatik 11/2010

Marc Santos, Harald F.O. von Kortzfleisch, Shared Annotation Model – Ein Datenmodell für kollaborative Annotationen, Arbeitsberichte aus dem Fachbereich Informatik 10/2010

Gerd Gröner, Steffen Staab, Categorization and Recognition of Ontology Refactoring Pattern, Arbeitsberichte aus dem Fachbereich Informatik 9/2010

Daniel Eißing, Ansgar Scherp, Carsten Saathoff, Integration of Existing Multimedia Metadata Formats and Metadata Standards in the M3O, Arbeitsberichte aus dem Fachbereich Informatik 8/2010

Stefan Scheglmann, Ansgar Scherp, Steffen Staab, Model-driven Generation of APIs for OWL-based Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 7/2010

Daniel Schmeiß, Ansgar Scherp, Steffen Staab, Integrated Mobile Visualization and Interaction of Events and POIs, Arbeitsberichte aus dem Fachbereich Informatik 6/2010

Rüdiger Grimm, Daniel Pähler, E-Mail-Forensik – IP-Adressen und ihre Zuordnung zu Internet-Teilnehmern und ihren Standorten, Arbeitsberichte aus dem Fachbereich Informatik 5/2010

Christoph Ringelstein, Steffen Staab, PAPEL: Syntax and Semantics for Provenance-Aware Policy Definition, Arbeitsberichte aus dem Fachbereich Informatik 4/2010

Nadine Lindermann, Sylvia Valcárcel, Harald F.O. von Kortzfleisch, Ein Stufenmodell für kollaborative offene Innovationsprozesse in Netzwerken kleiner und mittlerer Unternehmen mit Web 2.0, Arbeitsberichte aus dem Fachbereich Informatik 3/2010

Maria Wimmer, Dagmar Lück-Schneider, Uwe Brinkhoff, Erich Schweighofer, Siegfried Kaiser, Andreas Wieber, Fachtagung Verwaltungsinformatik FTVI Fachtagung Rechtsinformatik FTRI 2010, Arbeitsberichte aus dem Fachbereich Informatik 2/2010

Max Braun, Ansgar Scherp, Steffen Staab, Collaborative Creation of Semantic Points of Interest as Linked Data on the Mobile Phone, Arbeitsberichte aus dem Fachbereich Informatik 1/2010

Marc Santos, Einsatz von „Shared In-situ Problem Solving" Annotationen in kollaborativen Lern- und Arbeitsszenarien, Arbeitsberichte aus dem Fachbereich Informatik 20/2009

Carsten Saathoff, Ansgar Scherp, Unlocking the Semantics of Multimedia Presentations in the Web with the Multimedia Metadata Ontology, Arbeitsberichte aus dem Fachbereich Informatik 19/2009

Christoph Kahle, Mario Schaarschmidt, Harald F.O. von Kortzfleisch, Open Innovation: Kundenintegration am Beispiel von IPTV, Arbeitsberichte aus dem Fachbereich Informatik 18/2009

Dietrich Paulus, Lutz Priese, Peter Decker, Frank Schmitt, Pose-Tracking Forschungsbericht, Arbeitsberichte aus dem Fachbereich Informatik 17/2009

Andreas Fuhr, Tassilo Horn, Andreas Winter, Model-Driven Software Migration Extending SOMA, Arbeitsberichte aus dem Fachbereich Informatik 16/2009

Eckhard Großmann, Sascha Strauß, Tassilo Horn, Volker Riediger, Abbildung von grUML nach XSD soamig, Arbeitsberichte aus dem Fachbereich Informatik 15/2009

Kerstin Falkowski, Jürgen Ebert, The STOR Component System Interim Report, Arbeitsberichte aus dem Fachbereicht Informatik 14/2009

Sebastian Magnus, Markus Maron, An Empirical Study to Evaluate the Location of Advertisement Panels by Using a Mobile Marketing Tool, Arbeitsberichte aus dem Fachbereich Informatik 13/2009

Sebastian Magnus, Markus Maron, Konzept einer Public Key Infrastruktur in iCity, Arbeitsberichte aus dem Fachbereich Informatik 12/2009

Sebastian Magnus, Markus Maron, A Public Key Infrastructure in Ambient Information and Transaction Systems, Arbeitsberichte aus dem Fachbereich Informatik 11/2009

Ammar Mohammed, Ulrich Furbach, Multi-agent systems: Modeling and Virification using Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 10/2009

Andreas Sprotte, Performance Measurement auf der Basis von Kennzahlen aus betrieblichen Anwendungssystemen: Entwurf eines kennzahlengestützten Informationssystems für einen Logistikdienstleister, Arbeitsberichte aus dem Fachbereich Informatik 9/2009

Gwendolin Garbe, Tobias Hausen, Process Commodities: Entwicklung eines Reifegradmodells als Basis für Outsourcingentscheidungen, Arbeitsberichte aus dem Fachbereich Informatik 8/2009

Petra Schubert et. al., Open-Source-Software für das Enterprise Resource Planning, Arbeitsberichte aus dem Fachbereich Informatik 7/2009

Ammar Mohammed, Frieder Stolzenburg, Using Constraint Logic Programming for Modeling and Verifying Hierarchical Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 6/2009

Tobias Kippert, Anastasia Meletiadou, Rüdiger Grimm, Entwurf eines Common Criteria-Schutzprofils für Router zur Abwehr von Online-Überwachung, Arbeitsberichte aus dem Fachbereich Informatik 5/2009

Hannes Schwarz, Jürgen Ebert, Andreas Winter, Graph-based Traceability – A Comprehensive Approach. Arbeitsberichte aus dem Fachbereich Informatik 4/2009

Anastasia Meletiadou, Simone Müller, Rüdiger Grimm, Anforderungsanalyse für Risk-Management-Informationssysteme (RMIS), Arbeitsberichte aus dem Fachbereich Informatik 3/2009

Ansgar Scherp, Thomas Franz, Carsten Saathoff, Steffen Staab, A Model of Events based on a Foundational Ontology, Arbeitsberichte aus dem Fachbereich Informatik 2/2009

Frank Bohdanovicz, Harald Dickel, Christoph Steigner, Avoidance of Routing Loops, Arbeitsberichte aus dem Fachbereich Informatik 1/2009

Stefan Ameling, Stephan Wirth, Dietrich Paulus, Methods for Polyp Detection in Colonoscopy Videos: A Review, Arbeitsberichte aus dem Fachbereich Informatik 14/2008

Tassilo Horn, Jürgen Ebert, Ein Referenzschema für die Sprachen der IEC 61131-3, Arbeitsberichte aus dem Fachbereich Informatik 13/2008

Thomas Franz, Ansgar Scherp, Steffen Staab, Does a Semantic Web Facilitate Your Daily Tasks?, Arbeitsberichte aus dem Fachbereich Informatik 12/2008

Norbert Frick, Künftige Anfordeungen an ERP-Systeme: Deutsche Anbieter im Fokus, Arbeitsberichte aus dem Fachbereicht Informatik 11/2008

Jürgen Ebert, Rüdiger Grimm, Alexander Hug, Lehramtsbezogene Bachelor- und Masterstudiengänge im Fach Informatik an der Universität Koblenz-Landau, Campus Koblenz, Arbeitsberichte aus dem Fachbereich Informatik 10/2008

Mario Schaarschmidt, Harald von Kortzfleisch, Social Networking Platforms as Creativity Fostering Systems: Research Model and Exploratory Study, Arbeitsberichte aus dem Fachbereich Informatik 9/2008

Bernhard Schueler, Sergej Sizov, Steffen Staab, Querying for Meta Knowledge, Arbeitsberichte aus dem Fachbereich Informatik 8/2008

Stefan Stein, Entwicklung einer Architektur für komplexe kontextbezogene Dienste im mobilen Umfeld, Arbeitsberichte aus dem Fachbereich Informatik 7/2008

Matthias Bohnen, Lina Brühl, Sebastian Bzdak, RoboCup 2008 Mixed Reality League Team Description, Arbeitsberichte aus dem Fachbereich Informatik 6/2008

Bernhard Beckert, Reiner Hähnle, Tests and Proofs: Papers Presented at the Second International Conference, TAP 2008, Prato, Italy, April 2008, Arbeitsberichte aus dem Fachbereich Informatik 5/2008

Klaas Dellschaft, Steffen Staab, Unterstützung und Dokumentation kollaborativer Entwurfs- und Entscheidungsprozesse, Arbeitsberichte aus dem Fachbereich Informatik 4/2008

Rüdiger Grimm: IT-Sicherheitsmodelle, Arbeitsberichte aus dem Fachbereich Informatik 3/2008

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik 2/2008

Markus Maron, Kevin Read, Michael Schulze: CAMPUS NEWS – Artificial Intelligence Methods Combined for an Intelligent Information Network, Arbeitsberichte aus dem Fachbereich Informatik 1/2008

Lutz Priese,Frank Schmitt, Patrick Sturm, Haojun Wang: BMBF-Verbundprojekt 3D-RETISEG Abschlussbericht des Labors Bilderkennen der Universität Koblenz-Landau, Arbeitsberichte aus dem Fachbereich Informatik 26/2007

Stephan Philippi, Alexander Pinl: Proceedings 14. Workshop 20.-21. September 2007 Algorithmen und Werkzeuge für Petrinetze, Arbeitsberichte aus dem Fachbereich Informatik 25/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS – an Intelligent Bluetooth-based Mobile Information Network, Arbeitsberichte aus dem Fachbereich Informatik 24/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS - an Information Network for Pervasive Universities, Arbeitsberichte aus dem Fachbereich Informatik 23/2007

Lutz Priese: Finite Automata on Unranked and Unordered DAGs Extented Version, Arbeitsberichte aus dem Fachbereich Informatik 22/2007

Mario Schaarschmidt, Harald F.O. von Kortzfleisch: Modularität als alternative Technologie- und Innovationsstrategie, Arbeitsberichte aus dem Fachbereich Informatik 21/2007

Kurt Lautenbach, Alexander Pinl: Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 20/2007

Rüdiger Grimm, Farid Mehr, Anastasia Meletiadou, Daniel Pähler, Ilka Uerz: SOA-Security, Arbeitsberichte aus dem Fachbereich Informatik 19/2007

Christoph Wernhard: Tableaux Between Proving, Projection and Compilation, Arbeitsberichte aus dem Fachbereich Informatik 18/2007

Ulrich Furbach, Claudia Obermaier: Knowledge Compilation for Description Logics, Arbeitsberichte aus dem Fachbereich Informatik 17/2007

Fernando Silva Parreiras, Steffen Staab, Andreas Winter: TwoUse: Integrating UML Models and OWL Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 16/2007

Rüdiger Grimm, Anastasia Meletiadou: Rollenbasierte Zugriffskontrolle (RBAC) im Gesundheitswesen, Arbeitsberichte aud dem Fachbereich Informatik 15/2007

Ulrich Furbach, Jan Murray, Falk Schmidsberger, Frieder Stolzenburg: Hybrid Multiagent Systems with Timed Synchronization-Specification and Model Checking, Arbeitsberichte aus dem Fachbereich Informatik 14/2007

Björn Pelzer, Christoph Wernhard: System Description:“E-KRHyper“, Arbeitsberichte aus dem Fachbereich Informatik, 13/2007

Ulrich Furbach, Peter Baumgartner, Björn Pelzer: Hyper Tableaux with Equality, Arbeitsberichte aus dem Fachbereich Informatik, 12/2007

Ulrich Furbach, Markus Maron, Kevin Read: Location based Informationsystems, Arbeitsberichte aus dem Fachbereich Informatik, 11/2007

Philipp Schaer, Marco Thum: State-of-the-Art: Interaktion in erweiterten Realitäten, Arbeitsberichte aus dem Fachbereich Informatik, 10/2007

Ulrich Furbach, Claudia Obermaier: Applications of Automated Reasoning, Arbeitsberichte aus dem Fachbereich Informatik, 9/2007

Jürgen Ebert, Kerstin Falkowski: A First Proposal for an Overall Structure of an Enhanced Reality Framework, Arbeitsberichte aus dem Fachbereich Informatik, 8/2007

Lutz Priese, Frank Schmitt, Paul Lemke: Automatische See-Through Kalibrierung, Arbeitsberichte aus dem Fachbereich Informatik, 7/2007

Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, Marcel Weinand, Jörg Helbach: Security Requirements for Non-political Internet Voting, Arbeitsberichte aus dem Fachbereich Informatik, 6/2007

Daniel Bildhauer, Volker Riediger, Hannes Schwarz, Sascha Strauß, „grUML – Eine UML-basierte Modellierungssprache für T-Graphen“, Arbeitsberichte aus dem Fachbereich Informatik, 5/2007

Richard Arndt, Steffen Staab, Raphaël Troncy, Lynda Hardman: Adding Formal Semantics to MPEG-7: Designing a Well Founded Multimedia Ontology for the Web, Arbeitsberichte aus dem Fachbereich Informatik, 4/2007

Simon Schenk, Steffen Staab: Networked RDF Graphs, Arbeitsberichte aus dem Fachbereich Informatik, 3/2007

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J. Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe“
(http://www.uni-koblenz.de/fb4/publikationen/gelbereihe)

Lutz Priese: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißen: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005